



Red Hat Training and Certification

Libro de trabajo del estudiante (ROLE)

Red Hat Enterprise Linux 9.0 RH134

Red Hat System Administration II

Edición 2



Red Hat
Learning Community

Join a community dedicated to learning open source

The Red Hat® Learning Community is a collaborative platform for users to accelerate open source skill adoption while working with Red Hat products and experts.



Network with tens of thousands of community members



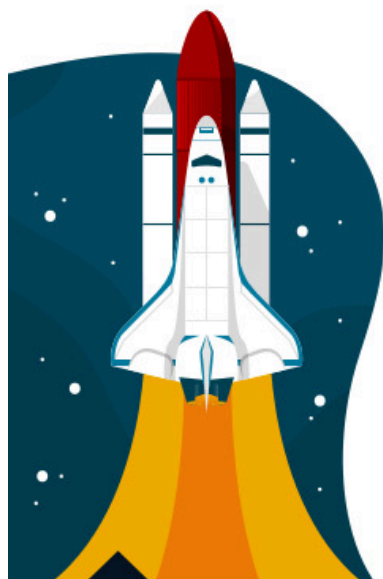
Engage in thousands of active conversations and posts



Join and interact with hundreds of certified training instructors



Unlock badges as you participate and accomplish new goals



This knowledge-sharing platform creates a space where learners can connect, ask questions, and collaborate with other open source practitioners.

Access free Red Hat training videos

Discover the latest Red Hat Training and Certification news

Connect with your instructor – and your classmates – before, after, and during your training course.

Join peers as you explore Red Hat products

Join the conversation learn.redhat.com



Copyright © 2020 Red Hat, Inc. Red Hat, Red Hat Enterprise Linux, the Red Hat logo, and Ansible are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

A long-exposure photograph of a city street at night, showing light trails from cars and buildings in the background. A semi-transparent white box is overlaid on the upper left, containing the title. A faint grid pattern is visible in the bottom right corner.

Red Hat System Administration II

Red Hat Enterprise Linux 9.0 RH134

Red Hat System Administration II

Edición 2 20220609

fecha de publicación 20220609

Autores: Ashish Lingayat, Bernardo Gargallo, Ed Parenti, Jacob Pelchat,
Mike Kelly, Morgan Weetman, Patrick Gomez
Editor: Julian Cable

Copyright © 2022 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are Copyright © 2022 Red Hat, Inc.

No part of this publication may be stored in a retrieval system, transmitted or reproduced in any way, including, but not limited to, photocopy, photograph, magnetic, electronic or other record, without the prior written permission of Red Hat, Inc.

This instructional program, including all material provided herein, is supplied without any guarantees from Red Hat, Inc. Red Hat, Inc. assumes no liability for damages or legal action arising from the use or misuse of contents or details contained herein.

If you believe Red Hat training materials are being used, copied, or otherwise improperly distributed, please send email to training@redhat.com or phone toll-free (USA) +1 (866) 626-2994 or +1 (919) 754-3700.

Red Hat, Red Hat Enterprise Linux, the Red Hat logo, JBoss, OpenShift, Fedora, Hibernate, Ansible, CloudForms, RHCA, RHCE, RHCSA, Ceph, and Gluster are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle American, Inc. and/or its affiliates.

XFS® is a registered trademark of Hewlett Packard Enterprise Development LP or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is a trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack word mark and the Square O Design, together or apart, are trademarks or registered trademarks of OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. Red Hat, Inc. is not affiliated with, endorsed by, or sponsored by the OpenStack Foundation or the OpenStack community.

All other trademarks are the property of their respective owners.

Colaboradores: Adarsh Krishnan, David Sacco, Hemant Chauhan, Roberto Velazquez, Sajith Eyamkuzhy, Samik Sanyal, Yuvaraj Balaraju

Convenciones del documento	xi
.....	xi
Introducción	xiii
Red Hat System Administration II	xiii
Orientación sobre el entorno del aula	xiv
Realización de ejercicios de laboratorio	xix
1. Mejora de la productividad con la línea de comandos	1
Redacción de scripts de Bash simples	2
Ejercicio Guiado: Redacción de scripts de Bash simples	6
Bucles y construcciones condicionales en scripts	9
Ejercicio Guiado: Bucles y construcciones condicionales en scripts	15
Texto coincidente en la salida de comando con expresiones regulares	17
Ejercicio Guiado: Texto coincidente en la salida de comando con expresiones regulares	25
Trabajo de laboratorio: Mejora de la productividad con la línea de comandos	28
Resumen	34
2. Programación de tareas futuras	35
Programación de un trabajo de usuario diferido	36
Ejercicio Guiado: Programación de un trabajo de usuario diferido	39
Programación de trabajos de usuario recurrentes	42
Ejercicio Guiado: Programación de trabajos de usuario recurrentes	45
Programación de trabajos del sistema recurrentes	48
Ejercicio Guiado: Programación de trabajos del sistema recurrentes	52
Administración de archivos temporales	55
Ejercicio Guiado: Administración de archivos temporales	59
Cuestionario: Programación de tareas futuras	62
Resumen	66
3. Ajuste del rendimiento del sistema	67
Ajuste de perfiles de optimización	68
Ejercicio Guiado: Ajuste de perfiles de optimización	75
Influencia en la programación de procesos	80
Ejercicio Guiado: Influencia en la programación de procesos	85
Trabajo de laboratorio: Ajuste del rendimiento del sistema	89
Resumen	95
4. Administración de seguridad de SELinux	97
Cambio del modo de cumplimiento (enforcement) de SELinux	99
Ejercicio Guiado: Cambio del modo de cumplimiento (enforcement) de SELinux	104
Control de contextos de archivo de SELinux	107
Ejercicio Guiado: Control de contextos de archivo de SELinux	112
Ajuste de la política de SELinux con booleanos	115
Ejercicio Guiado: Ajuste de la política de SELinux con booleanos	118
Investigación y resolución de problemas de SELinux	121
Ejercicio Guiado: Investigación y resolución de problemas de SELinux	126
Trabajo de laboratorio: Administración de seguridad de SELinux	130
Resumen	136
5. Administración de almacenamiento básico	137
Adición de particiones, sistemas de archivos y montajes persistentes	138
Ejercicio Guiado: Adición de particiones, sistemas de archivos y montajes persistentes	148
Administración de espacio de intercambio (swap)	152
Ejercicio Guiado: Administración de espacio de intercambio (swap)	157
Trabajo de laboratorio: Administración de almacenamiento básico	161
Resumen	169

6. Administración de la pila (stack) de almacenamiento	171
Creación y ampliación de volúmenes lógicos	172
Ejercicio Guiado: Creación y ampliación de volúmenes lógicos	183
Administración de almacenamiento en capas	189
Ejercicio Guiado: Administración de almacenamiento en capas	196
Trabajo de laboratorio: Administración de la pila (stack) de almacenamiento	202
Resumen	208
7. Acceso al almacenamiento conectado a la red	209
Administración de almacenamiento conectado a la red con NFS	210
Ejercicio Guiado: Administración de almacenamiento conectado a la red con NFS	214
Montaje automático de almacenamiento conectado a la red	218
Ejercicio Guiado: Montaje automático de almacenamiento conectado a la red	223
Trabajo de laboratorio: Acceso al almacenamiento conectado a la red	229
Resumen	236
8. Control del proceso de arranque	237
Selección del objetivo de arranque	238
Ejercicio Guiado: Selección del objetivo de arranque	244
Restablecimiento de la contraseña de root	247
Ejercicio Guiado: Restablecimiento de la contraseña de root	252
Reparación de problemas del sistema de archivos en el arranque	254
Ejercicio Guiado: Reparación de problemas del sistema de archivos en el arranque	257
Trabajo de laboratorio: Control del proceso de arranque	260
Resumen	266
9. Administración de la seguridad de redes	267
Administración de firewalls del servidor	268
Ejercicio Guiado: Administración de firewalls del servidor	276
Control del etiquetado de puertos de SELinux	279
Ejercicio Guiado: Control del etiquetado de puertos de SELinux	282
Trabajo de laboratorio: Administración de la seguridad de redes	286
Resumen	294
10. Instalación de Red Hat Enterprise Linux	295
Instalación de Red Hat Enterprise Linux	296
Ejercicio Guiado: Instalación de Red Hat Enterprise Linux	301
Automatización de la instalación con Kickstart	304
Ejercicio Guiado: Automatización de la instalación con Kickstart	313
Instalación y configuración de máquinas virtuales	316
Cuestionario: Instalación y configuración de máquinas virtuales	322
Trabajo de laboratorio: Instalación de Red Hat Enterprise Linux	324
Resumen	330
11. Ejecución de contenedores	331
Conceptos de contenedores	332
Cuestionario: Conceptos de contenedores	339
Implementación de contenedores	343
Ejercicio Guiado: Implementación de contenedores	353
Administración del almacenamiento del contenedor y los recursos de red	359
Ejercicio Guiado: Administración del almacenamiento del contenedor y los recursos de red	370
Administración de contenedores como servicios del sistema	376
Ejercicio Guiado: Administración de contenedores como servicios del sistema	383
Trabajo de laboratorio: Ejecución de contenedores	389
Resumen	396
12. Revisión exhaustiva	397

Revisión exhaustiva	398
Trabajo de laboratorio: Corrección de problemas de arranque y mantenimiento de servidores	401
Trabajo de laboratorio: Configuración y administración de sistemas de archivos y almacenamiento	408
Trabajo de laboratorio: Configuración y administración de seguridad del servidor	414
Trabajo de laboratorio: Ejecución de contenedores	424



Convenciones del documento

En esta sección, se describen diversas convenciones y prácticas usadas en todos los cursos de capacitación de Red Hat.

Admoniciones

En los cursos de capacitación de Red Hat, se usan las siguientes admoniciones:



Referencias

Describen el lugar donde se puede encontrar documentación externa relevante para un tema.



nota

Son consejos, atajos o enfoques alternativos para una tarea determinada. Ignorar una nota no debería tener consecuencias negativas, pero podría pasarse por alto algún truco que puede simplificar una tarea.



Importante

Estas detallan cosas que se olvidan con facilidad: cambios de configuración que solo se aplican a la sesión actual o servicios que se deben reiniciar para poder aplicar una actualización. Ignorar estas admoniciones no provocará pérdida de datos, pero puede causar irritación y frustración.



Advertencia

No deben ignorarse. Es muy probable que ignorar estas admoniciones provoque pérdida de datos.

Lenguaje inclusivo

La capacitación de Red Hat actualmente está revisando su uso del lenguaje en diversas áreas para ayudar a eliminar cualquier término que pueda ser ofensivo. Se trata de un proceso continuo y requiere la alineación con los productos y servicios que se abordan en los cursos de capacitación de Red Hat. Red Hat agradece su paciencia durante este proceso.

Introducción

Red Hat System Administration II

Este curso está diseñado específicamente para estudiantes que hayan completado el curso *Red Hat System Administration I (RH124)*. El curso *Red Hat System Administration II (RH134)* se centra en las tareas principales necesarias para convertirse en un administrador de Linux de tiempo completo y validar esas habilidades mediante el examen *Red Hat Certified System Administrator*. Este curso aborda en detalle la administración de Enterprise Linux, que incluye sistemas de archivos, particiones, volúmenes lógicos, SELinux, firewalls y solución de problemas.

Objetivos del curso

- Ampliar las habilidades obtenidas durante el curso de *Red Hat System Administration I (RH124)*.
- Desarrollar las habilidades que necesita un administrador de sistemas Red Hat Enterprise Linux con certificación RHCSA.

Destinatarios

- Este curso está diseñado particularmente para estudiantes que hayan completado el curso *Red Hat System Administration I (RH124)*. Por la organización de los temas, no es adecuado que los estudiantes usen RH134 como punto de entrada del plan de estudios. Se alienta a los estudiantes que no hayan tomado un curso de Red Hat anterior a tomar *Red Hat System Administration I (RH124)* si no conocen Linux o el curso *RHCSA Rapid Track (RH199)* si tienen experiencia con la administración de Enterprise Linux.

Requisitos

- Haber realizado el curso *Red Hat System Administration I (RH124)* o tener conocimientos equivalentes.

Orientación sobre el entorno del aula

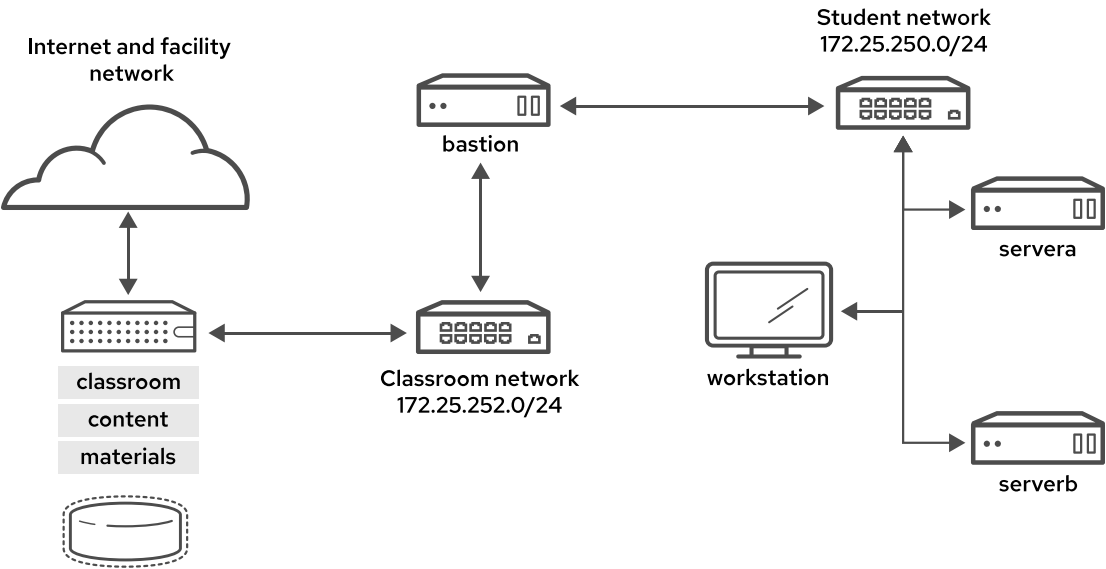


Figura 0.1: Entorno del aula

En este curso, el sistema de cómputo principal usado para las actividades prácticas de aprendizaje es `workstation`. Los estudiantes también usan otras dos máquinas para estas actividades: `servera` y `serverb`. Estos tres sistemas se encuentran en el dominio `DNS lab.example.com`.

Todos los sistemas de cómputo de los estudiantes tienen una cuenta de usuario estándar (`student`) con la contraseña `student`. La contraseña `root` de todos los sistemas de los estudiantes es `redhat`.

Máquinas del aula

Nombre de la máquina	dirección IP	Rol
<code>bastion.lab.example.com</code>	172.25.250.254	Sistema de puerta de enlace para conectar la red privada de los estudiantes al servidor del aula (debe estar siempre en ejecución)
<code>classroom.example.com</code>	172.25.254.254	Servidor que aloja los materiales requeridos para el aula
<code>workstation.lab.example.com</code>	172.25.250.9	Estación de trabajo gráfica para uso del estudiante
<code>servera.lab.example.com</code>	172.25.250.10	Servidor administrado "A"
<code>serverb.lab.example.com</code>	172.25.250.11	Servidor administrado "B"

La función principal de **bastion** es actuar como enrutador entre la red que conecta las máquinas de los estudiantes y la red del aula. Si **bastion** está apagada, las máquinas de los estudiantes solo podrán acceder a sistemas en la red de estudiantes individuales.

Varios sistemas en el aula brindan servicios de soporte. Dos servidores, **content.example.com** y **materials.example.com**, son fuentes de software y materiales del trabajo de laboratorio usados en actividades prácticas. Se provee información sobre cómo usar estos servidores en las instrucciones para estas actividades. La máquina virtual **workstation** proporciona estas actividades. Tanto **classroom** como **bastion** se deben estar ejecutando siempre para el uso adecuado del entorno de laboratorio.

**nota**

Al iniciar sesión en **servera** o **serverb** es posible que vea un mensaje sobre la activación de **cockpit**. Puede ignorar los mensajes.

```
[student@workstation ~]$ ssh student@serverb
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list of
known hosts.
Activate the web console with: systemctl enable --now cockpit.socket

[student@serverb ~]$
```

Control de sus sistemas

Se le asignan computadoras remotas en un aula de aprendizaje en línea de Red Hat (Red Hat Online Learning Environment, ROLE). Se accede a los cursos de autoaprendizaje a través de una aplicación web alojada en rol.redhat.com [<http://rol.redhat.com>]. Inicie sesión en este sitio con sus credenciales de usuario del Portal de clientes de Red Hat.

Control de las máquinas virtuales

Las máquinas virtuales del entorno de su aula se controlan a través de controles de interfaz en la página web. El estado de cada máquina virtual en el aula se muestra en la pestaña **Lab Environment**.

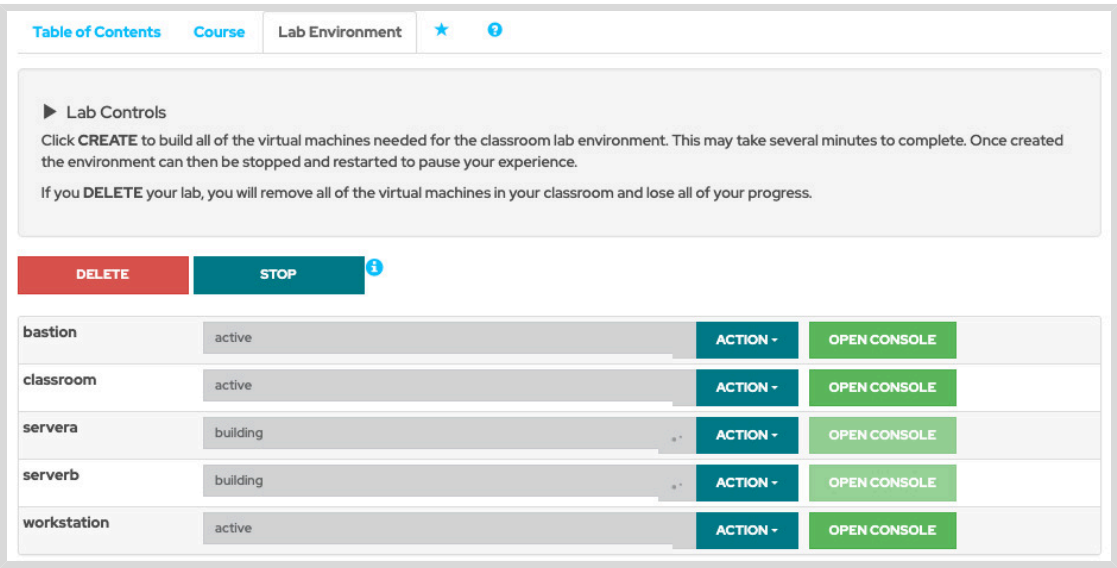


Figura 0.2: Ejemplo de la página de gestión del Entorno de laboratorio de un curso

Estados de la máquina

Estado de la máquina virtual	Descripción
building (en construcción)	La máquina virtual se está creando.
active (activa)	La máquina virtual se está ejecutando y está disponible. Si acaba de arrancar, podría estar iniciando servicios todavía.
stopped (detenida)	La máquina virtual se ha apagado completamente. Al iniciarse, la máquina virtual arranca en el mismo estado en que se hallaba antes de apagarse. El estado del disco se conserva.

Acciones del aula

Botón o acción	Descripción
CREATE	Crea el aula de ROLE. Crea e inicia todas las máquinas virtuales necesarias para esta aula. La creación puede tardar varios minutos en completarse.
CREATING	Se están creando las máquinas virtuales del aula de ROLE. Crea e inicia todas las máquinas virtuales necesarias para esta aula. La creación puede tardar varios minutos en completarse.
DELETE	Elimina el aula de ROLE. Destruye todas las máquinas virtuales del aula. Se perderá todo el trabajo guardado en los discos del sistema.
START	Inicia todas las máquinas virtuales en el aula.
STARTING	Todas las máquinas virtuales en el aula se están iniciando.

Botón o acción	Descripción
STOP	Detiene todas las máquinas virtuales en el aula.

Acciones de la máquina

Botón o acción	Descripción
OPEN CONSOLE	Se conecta con la consola del sistema de la máquina virtual en una nueva pestaña del navegador. Puede iniciar sesión directamente en la máquina virtual y ejecutar los comandos cuando sea necesario. Habitualmente, debe iniciar sesión solo en la máquina virtual workstation y usar desde allí ssh para conectarse con las otras máquinas virtuales.
ACTION > Start	Inicia (power on [encender]) la máquina virtual.
ACTION > Shutdown	Apaga la máquina virtual correctamente y preserva el contenido del disco.
ACTION > Power Off	Fuerza el apagado de la máquina virtual al tiempo que preserva el contenido del disco. Esta acción equivale a desenchufar una máquina física.
ACTION > Reset	Fuerza el apagado de la máquina virtual y restablece el almacenamiento asociado para que vuelva a su estado original. Se perderá todo el trabajo guardado en los discos del sistema.

Al inicio de un ejercicio, si se le indica que restablezca el nodo de una máquina virtual, haga clic en **ACTION > Reset** solo para esa máquina virtual específica.

Al inicio de un ejercicio, si se le indica que restablezca todas las máquinas virtuales, haga clic en **ACTION > Reset** en cada máquina virtual de la lista.

Si desea que el entorno del aula vuelva a su estado original al inicio del curso, haga clic en **DELETE** para suprimir el entorno del aula completo. Después de eliminar el trabajo de laboratorio, haga clic en **CREATE** para aprovisionar un nuevo conjunto de sistemas del aula.



Advertencia

La operación **DELETE** (ELIMINAR TRABAJO DE LABORATORIO) no puede deshacerse. Se perderá todo el trabajo que haya completado en el entorno del aula.

Temporizadores de detención automática y eliminación automática

La inscripción al aprendizaje en línea de Red Hat le otorga derecho a un cierto tiempo de uso del equipo. Para ayudarlo a conservar el tiempo asignado, el aula de ROLE usa temporizadores, que apagan o eliminan el entorno del aula cuando el temporizador correspondiente caduca.

Para ajustar los temporizadores, ubique los dos botones + en la parte inferior de la página de administración del curso. Haga clic en el botón + de detención automática para agregar otra hora al temporizador de detención automática. Haga clic en el botón + de eliminación automática para agregar otro día al temporizador de eliminación automática. La detención automática tiene

un tiempo máximo de 11 horas, y la eliminación automática un tiempo máximo de 14 días. Preste atención y mantenga los temporizadores configurados mientras trabaja, de forma tal que el entorno no se apague de manera imprevista. Tenga cuidado de no establecer los temporizadores en tiempos innecesariamente prolongados, ya que podría perder el tiempo de suscripción asignado.

Realización de ejercicios de laboratorio

Es posible que vea los siguientes tipos de actividades de laboratorio en este curso:

- Un *ejercicio guiado* es un ejercicio práctico que sigue a una sección de presentación. Lo guía paso a paso para realizar un procedimiento.
- Un *cuestionario* se suele usar para corroborar el aprendizaje de los conocimientos o cuando una actividad práctica resulta poco práctica por algún otro motivo.
- Un *trabajo de laboratorio al final del capítulo* es una actividad práctica que se puede calificar para ayudarlo a verificar su aprendizaje. Usted trabajará en una serie de pasos de alto nivel, los cuales están basados en los ejercicios guiados de ese capítulo, pero los pasos no lo guiarán en cada comando. Se proporciona una solución con un tutorial paso a paso.
- Se usa un *trabajo de laboratorio con revisión integral* al final del curso. También es una actividad práctica que se puede calificar y puede abarcar el contenido de todo el curso. Usted trabaja en una especificación de lo que debe lograr en la actividad, sin recibir los pasos específicos para hacerlo. Reiteramos: se proporciona una solución con un tutorial paso a paso que cumple la especificación.

Para preparar su entorno de laboratorio al inicio de cada actividad práctica, ejecute el comando **lab start** con el nombre de actividad especificado de las instrucciones de la actividad. Del mismo modo, al final de cada actividad práctica, ejecute el comando **lab finish** con el mismo nombre de actividad para eliminar todo después de la actividad. Cada actividad práctica tiene un nombre único dentro de un curso.

La sintaxis para ejecutar un script de ejercicio es la siguiente:

```
[student@workstation ~]$ lab action exercise
```

Las *acciones* son **start** (iniciar), **grade** (calificar) o **finish** (finalizar). Todos los ejercicios soportan las acciones **start** (iniciar) y **finish** (finalizar). Solo los ejercicios de laboratorio al final del capítulo y los ejercicios de laboratorio con revisión integral soportan la acción **grade** (calificar).

start (iniciar)

La acción **start** verifica los recursos necesarios para comenzar un ejercicio. Esto podría incluir los ajustes de configuración, la creación de recursos, la verificación de los servicios con requisitos previos y la verificación de los resultados necesarios de los ejercicios anteriores. Puede realizar un ejercicio en cualquier momento, incluso sin realizar los ejercicios anteriores.

grade (calificar)

Para las actividades calificables, la acción **grade** dirige el comando **lab** para evaluar su trabajo y muestra una lista de criterios de calificación con un estado **PASS** o **FAIL** para cada uno. Para que el estado sea **PASS** en todos los criterios, corrija los errores y vuelva a ejecutar la acción **grade**.

finish (finalizar)

La acción **finish** elimina los recursos que se configuraron durante el ejercicio. Puede realizar un ejercicio tantas veces como desee.

El comando **lab** admite la finalización con tabulación. Por ejemplo, para enumerar todos los ejercicios que puede iniciar, ingrese **lab start** y, luego, presione la tecla **Tab** dos veces.

capítulo 1

Mejora de la productividad con la línea de comandos

Meta

Ejecutar los comandos de manera más eficiente mediante el uso de las funciones avanzadas de la shell de Bash, los scripts de shell y diversas utilidades proporcionadas por Red Hat Enterprise Linux.

Objetivos

- Ejecutar los comandos de manera más eficiente mediante el uso de las funciones avanzadas de la shell de Bash, los scripts de shell y diversas utilidades proporcionadas por Red Hat Enterprise Linux.
- Ejecutar tareas repetitivas con bucles `for`, evaluar códigos de salida de comandos y scripts, ejecutar pruebas con operadores y crear estructuras condicionales con sentencias `if`.
- Crear expresiones regulares para hacer coincidir datos, aplicar expresiones regulares a archivos de texto con el comando `grep` y usar `grep` para buscar archivos y datos de comandos canalizados.

Secciones

- Redacción de scripts de Bash simples (y ejercicio guiado)
- Bucles y construcciones condicionales en scripts (y ejercicio guiado)
- Texto coincidente en la salida de comando con expresiones regulares (y ejercicio guiado)

Trabajo de laboratorio

Mejora de la productividad con la línea de comandos

Redacción de scripts de Bash simples

Objetivos

Ejecutar los comandos de manera más eficiente mediante el uso de las funciones avanzadas de la shell de Bash, los scripts de shell y diversas utilidades proporcionadas por Red Hat Enterprise Linux.

Creación y ejecución de scripts de shell de Bash

Puede realizar muchas tareas simples y comunes de administración de sistemas usando herramientas de la línea de comandos. Las tareas que revisten mayor complejidad suelen requerir la concatenación de varios comandos que se pasen resultados entre ellos. Usando el entorno de shell Bash y las funciones de scripting, puede combinar los comandos de Linux en *scripts de shell* para resolver los problemas repetitivos del mundo real.

Un script de shell Bash es un archivo ejecutable que contiene una lista de comandos, y posiblemente con lógica de programación para controlar la toma de decisiones en la tarea general. Si está bien escrito, un script de shell puede convertirse en una herramienta potente de la línea de comandos cuando se ejecuta solo y puede usarlo con otros scripts.

Para los administradores de sistemas en todos los entornos operativos, es esencial que tengan experiencia con scripting de shell. Puede usar scripts de shell para mejorar la eficiencia y precisión de la finalización de tareas rutinarias.

Si bien puede usar cualquier editor de texto, los editores avanzados, como `vim` o `emacs`, entienden la sintaxis de shell Bash y pueden proporcionar resaltado codificado por colores. Este resaltado ayuda a identificar errores comunes de script, como sintaxis incorrecta, comillas que no coinciden, paréntesis, llaves y corchetes, y otros errores estructurales.

Especificación del intérprete de comandos

La primera línea de un script comienza con la notación `#!`, que comúnmente se denominan `she-bang` o `hash-bang`, a partir de los nombres de esos dos caracteres, `sharp` o `hash` y `bang`. Esta notación es una directiva del intérprete que indica el intérprete de comandos y las opciones de comando opcionales que se necesitan para procesar las líneas restantes del archivo. Para archivos de script de sintaxis Bash normales, la primera línea es esta directiva:

```
#!/usr/bin/bash
```

Ejecución de un script de shell Bash

Un archivo de script de shell debe tener permisos de ejecución para ejecutarlo como un comando común. Use el comando `chmod` para modificar los permisos del archivo. Use el comando `chown`, si es necesario, para otorgar permiso de ejecución solo para usuarios o grupos específicos.

Si el script está almacenado en un directorio que se enumera en la variable de entorno `PATH` de la shell, puede ejecutar el script de shell usando solo su nombre de archivo, de manera similar a ejecutar comandos compilados. Debido a que el análisis de `PATH` ejecuta el primer nombre de archivo coincidente que se encuentra, evite siempre usar nombres de comandos existentes para

nombrar sus archivos de script. Si un script no está en un directorio PATH, ejecute el script con su nombre de ruta absoluto, que se puede determinar al consultar el archivo con el comando **which**. De manera alternativa, ejecute un script en su directorio de trabajo actual con el prefijo de directorio **.**, como **./scriptname**.

```
[user@host ~]$ which hello
~/bin/hello
[user@host ~]$ echo $PATH
/home/user/.local/bin:/home/user/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/
sbin:/usr/local/bin
```

Entrecomillado de caracteres especiales

Una serie de caracteres y palabras tienen significados especiales para la shell Bash. Cuando debe usar estos caracteres para sus valores literales, en lugar de para sus significados especiales, los *evitará* en el script. Use el carácter de barra invertida (****), comillas simples (**' '**) o comillas dobles (**" "**) para eliminar (o *evitar*) el significado especial de estos caracteres.

El carácter de barra invertida quita el significado especial del carácter individual que le sigue inmediatamente. Por ejemplo, para usar el comando **echo** para mostrar la cadena **# not a comment** literal, el carácter de hash **#** no debe interpretarse como comentario.

El siguiente ejemplo muestra el carácter de barra invertida (****) que modifica el carácter hash para que no se interprete como un comentario.

```
[user@host ~]$ echo # not a comment

[user@host ~]$ echo \# not a comment
# not a comment
```

Para evitar más de un carácter en una cadena de texto, use la barra invertida varias veces o encierre toda la cadena entre comillas simples (**' '**) para interpretarla literalmente. Las comillas simples conservan el significado literal de todos los caracteres que encierran. Observe el carácter de barra invertida y las comillas simples en estos ejemplos:

```
[user@host ~]$ echo # not a comment #

[user@host ~]$ echo \# not a comment #
# not a comment
[user@host ~]$ echo \# not a comment \#
# not a comment #
[user@host ~]$ echo '# not a comment #'
# not a comment #
```

Use comillas dobles para suprimir **globbing** (coincidencia de patrones en el nombre del archivo) y **shell expansion**, pero permita la sustitución de comandos y variables. A nivel conceptual, la sustitución de variables es idéntica a la sustitución de comandos, pero puede usar sintaxis de llaves opcional. Observe los siguientes ejemplos de las diversas formas de comillas a continuación.

Use comillas simples para interpretar literalmente todo el texto encerrado. Además de suprimir el **globbing** y la expansión de la shell, las comillas simples dirigen a la shell para que suprima la sustitución de comandos y variables. El signo de interrogación (**?**) se incluye dentro de las comillas, ya que es un *metacarácter* que también evita la expansión.


```
[user@host ~]$ var=$(hostname -s); echo $var
host
[user@host ~]$ echo "***** hostname is ${var} *****"
***** hostname is host *****
[user@host ~]$ echo Your username variable is \${USER}.
Your username variable is $USER.
[user@host ~]$ echo "Will variable $var evaluate to $(hostname -s)?"
Will variable host evaluate to host?
[user@host ~]$ echo 'Will variable $var evaluate to $(hostname -s)?'
Will variable $var evaluate to $(hostname -s)?
[user@host ~]$ echo "\"Hello, world\""
"Hello, world"
[user@host ~]$ echo '"Hello, world"'
"Hello, world"
```

Provisión de salida de un script de shell

El comando `echo` visualiza texto arbitrario al asignar el texto como argumento del comando. De manera predeterminada, el texto se envía a la *salida estándar* (*STDOUT*), pero puede enviar texto a otra parte mediante la redirección de salida. En el siguiente script de Bash simple, el comando `echo` muestra el mensaje "Hello, wor ld" a *STDOUT*, que se muestra de forma predeterminada en el dispositivo de pantalla

```
[user@host ~]$ cat ~/bin/hello
#!/usr/bin/bash

echo "Hello, world"

[user@host ~]$ hello
Hello, world
```



nota

Este usuario solo puede ejecutar `hello` en el prompt porque el directorio `~/bin` (`/home/user/bin`) está en la variable `PATH` del usuario y el script `hello` tiene permiso de ejecutable. El analizador `PATH` encontrará el script primero, si no se encuentra ningún otro archivo ejecutable llamado `hello` en ningún directorio `PATH` anterior. El subdirectorio `bin` de su directorio de inicio está diseñado para almacenar sus scripts personales para que se puedan invocar fácilmente.

El comando `echo` se usa ampliamente en scripts de shell para mostrar mensajes informativos o de error. Los mensajes pueden ser indicadores útiles del progreso de un script y pueden dirigirse a salida estándar, error estándar, o bien ser redirigidos a un archivo de registro para su archivado. Al mostrar mensajes de error, una buena práctica de programación es redirigir los mensajes de error a *STDERR* para separarlos de la salida normal del programa.

```
[user@host ~]$ cat ~/bin/hello
#!/usr/bin/bash

echo "Hello, world"
echo "ERROR: Houston, we have a problem." >&2
```

```
[user@host ~]$ hello 2> hello.log
Hello, world
[user@host ~]$ cat hello.log
ERROR: Houston, we have a problem.
```

El comando `echo` también puede resultar de utilidad para depurar un script de shell problemático. Agregar enunciados `echo` en un script, para mostrar valores de variables y otra información de tiempo de ejecución, puede ayudar a aclarar cómo se comporta un script.



Referencias

Páginas del manual: `bash(1)`, `echo(1)` y `echo(1p)`.

► Ejercicio Guiado

Redacción de scripts de Bash simples

En este ejercicio, escribirá un script Bash simple con una secuencia de comandos y la ejecutará desde la línea de comandos.

Resultados

- Escribir y ejecutar un simple script Bash.
- Redirigir la salida de un script de Bash simple a un archivo.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start console-write
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Cree y ejecute un script simple de Bash.

- 2.1. Use el comando `vim` para crear el archivo `firstscript.sh` en su directorio de inicio.

```
[student@servera ~]$ vim firstscript.sh
```

- 2.2. Inserte el siguiente texto y guarde el archivo. Tenga en cuenta que la cantidad de señales hash (`#`) es arbitraria.

```
#!/usr/bin/bash
echo "This is my first bash script" > ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
```

- 2.3. Use el comando `sh` para ejecutar el script.

```
[student@servera ~]$ sh firstscript.sh
```

2.4. Revise el archivo de salida generado por el script.

```
[student@servera ~]$ cat output.txt
This is my first bash script

#####
```

► 3. Agregue más comandos al script `firstscript.sh`, ejecútelo y revise la salida.

3.1. Use el editor de texto Vim para editar el script `firstscript.sh`.

```
[student@servera ~]$ vim firstscript.sh
```

En la siguiente salida, se muestra el contenido esperado del archivo de texto `firstscript.sh`:

```
#!/usr/bin/bash
#
echo "This is my first bash script" > ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "LIST BLOCK DEVICES" >> ~/output.txt
echo "" >> ~/output.txt
lsblk >> ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "FILESYSTEM FREE SPACE STATUS" >> ~/output.txt
echo "" >> ~/output.txt
df -h >> ~/output.txt
echo "#####" >> ~/output.txt
```

3.2. Haga que el archivo `firstscript.sh` sea ejecutable con el comando `chmod`.

```
[student@servera ~]$ chmod a+x firstscript.sh
```

3.3. Ejecute el script `firstscript.sh`.

```
[student@servera ~]$ ./firstscript.sh
```

3.4. Revise el archivo de salida generado por el script.

```
[student@servera ~]$ cat output.txt
This is my first bash script

#####
LIST BLOCK DEVICES

NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sr0      11:0    1   558K  0 rom
vda      252:0    0    10G  0 disk
└─vda1   252:1    0     1M  0 part
└─vda2   252:2    0   200M  0 part /boot/efi
```

```
└─vda3 252:3    0  500M  0 part /boot
└─vda4 252:4    0  9.3G  0 part /
vdb    252:16   0    5G  0 disk
vdc    252:32   0    5G  0 disk
vdd    252:48   0    5G  0 disk

#####
FILESYSTEM FREE SPACE STATUS

Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        844M   0  844M   0% /dev
tmpfs           888M   0  888M   0% /dev/shm
tmpfs           355M  9.4M  346M   3% /run
/dev/vda4       9.4G  1.7G  7.7G  18% /
/dev/vda3       495M  161M  335M  33% /boot
/dev/vda2       200M   7.6M  193M   4% /boot/efi
tmpfs           178M   0  178M   0% /run/user/1000
#####
```

- 4. Elimine los archivos de ejercicios y regrese a la máquina `workstation`.

4.1. Elimine los archivos `firstscript.sh` y `output.txt`.

```
[student@servera ~]$ rm firstscript.sh output.txt
```

4.2. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish console-write
```

Esto concluye la sección.

Bucles y construcciones condicionales en scripts

Objetivos

Ejecutar tareas repetitivas con bucles `for`, evaluar códigos de salida de comandos y scripts, ejecutar pruebas con operadores y crear estructuras condicionales con sentencias `if`.

Uso de bucles para iterar comandos

Los administradores de sistemas a menudo se encuentran con tareas repetitivas en sus actividades cotidianas. Una tarea repetitiva es ejecutar un comando varias veces en un objetivo, como controlar un proceso una vez por minuto durante 10 minutos para saber si ha finalizado. Otro ejemplo es ejecutar un comando una vez para varios objetivos, como realizar una copia de seguridad de varias bases de datos en un sistema. El *bucle for* es una construcción de bucle de Bash que se usa para las iteraciones de tareas.

Procesamiento de ítems desde la línea de comandos

En Bash, la construcción del bucle `for` emplea la siguiente sintaxis:

```
for VARIABLE in LIST; do
  COMMAND VARIABLE
done
```

El bucle procesa las cadenas proporcionadas en *LIST* y termina después de procesar la última cadena de la lista. El bucle `for` almacena temporalmente cada cadena de lista como el valor de *VARIABLE*, luego ejecuta el bloque de comandos que usan la variable. El nombre de la variable es arbitrario. Normalmente, se hace referencia al valor de la variable con comandos en el bloque de comandos.

Proporcione la lista de cadenas para el bucle `for` a partir de una lista ingresada directamente por el usuario o generada a partir de la expansión de shell, como expansión de variable, expansión de llave, expansión de nombre de archivo o sustitución de comandos.

Estos ejemplos demuestran diferentes maneras de proporcionar cadenas a los bucles `for`:

```
[user@host ~]$ for HOST in host1 host2 host3; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for HOST in host{1,2,3}; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for HOST in host{1..3}; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for FILE in file{a..c}; do ls $FILE; done
filea
```



```

fileb
filec
[user@host ~]$ for PACKAGE in $(rpm -qa | grep kernel); \
do echo "$PACKAGE was installed on \
$(date -d @$(rpm -q --qf "%{INSTALLTIME}\n" $PACKAGE)); done
kernel-tools-libs-5.14.0-70.2.1.el9_0.x86_64 was installed on Thu Mar 24 10:52:40
PM EDT 2022
kernel-tools-5.14.0-70.2.1.el9_0.x86_64 was installed on Thu Mar 24 10:52:40 PM
EDT 2022
kernel-core-5.14.0-70.2.1.el9_0.x86_64 was installed on Thu Mar 24 10:52:46 PM EDT
2022
kernel-modules-5.14.0-70.2.1.el9_0.x86_64 was installed on Thu Mar 24 10:52:47 PM
EDT 2022
kernel-5.14.0-70.2.1.el9_0.x86_64 was installed on Thu Mar 24 10:53:04 PM EDT 2022
[user@host ~]$ for EVEN in $(seq 2 2 10); do echo "$EVEN"; done
2
4
6
8
10

```

Códigos de salida del script Bash

Después de que un script interpreta y procesa todo su contenido, el proceso de script sale y devuelve el control al proceso principal que lo llamó. Sin embargo, se puede salir de un script antes de que finalice, como cuando el script encuentra una condición de error. Use el comando `exit` para salir inmediatamente del script y omitir el procesamiento del resto del script.

Use el comando `exit` como un argumento entero opcional entre 0 y 255, que representa un *código de salida*. Se devuelve un código de salida a un proceso principal para indicar el estado en la salida. Un valor de código de salida de 0 representa una finalización correcta del script sin errores. Todos los demás valores diferentes a cero indican un código de salida con error. El programador de scripts define estos códigos. Use valores únicos para representar las diferentes condiciones de error que se encuentran. Recupere el código de salida del último comando completado de la variable incorporada `?`, como en los siguientes ejemplos:

```

[user@host bin]$ cat hello
#!/usr/bin/bash
echo "Hello, world"
exit 0
[user@host bin]$ ./hello
Hello, world
[user@host bin]$ echo $?
0

```

Cuando se usa el comando `exit` de un script sin un argumento de código de salida, el script devuelve el código de salida del último comando que se ejecutó dentro del script.

Prueba de la lógica para cadenas y directorios, y comparación de valores

Para asegurarse de que condiciones imprevistas no ocasionen fácilmente la falla de scripts, se recomienda que verifique la entrada de comandos, como argumentos de la línea de comandos,

entradas del usuario, sustituciones de comandos, expansiones de variables y expansiones de nombres de archivo. Puede verificar la integridad de sus scripts con el comando Bash `test`.

Todos los comandos producen un código de salida al finalizar.

Para ver el estado de salida, vea la variable `$?` inmediatamente después de la ejecución del comando `test`. Un estado de salida de 0 indica una salida exitosa sin nada que informar y los valores diferentes de cero indican alguna condición o falla. Realice pruebas con diversos operadores para determinar si un número es mayor que (`gt`), mayor o igual que (`ge`), menor que (`lt`), menor o igual que (`le`), o igual (`eq`) a otro número.

Use operadores para probar si una cadena de texto es igual (`=` o `==`) o no es lo mismo (`!=`) que otra cadena de texto, o si la cadena tiene una longitud cero (`z`) o un valor distinto de cero longitud (`n`). También puede probar si existe un archivo regular (`-f`) o directorio (`-d`) y algunos atributos especiales, como si el archivo es un enlace simbólico (`-L`) o si el usuario tiene permisos de lectura (`-r`).

**nota**

Los scripts de shell usan muchos tipos de operadores, además de los operadores que se analizan aquí. La página del manual de `test(1)` enumera los operadores de expresiones condicionales con descripciones. La página del manual de `bash(1)` también explica el uso y la evaluación del operador, pero puede ser complejo de leer. Red Hat recomienda que los estudiantes aprendan sobre scripting de shell con libros y cursos dedicados a la programación de shell.

A continuación se ejemplifica el uso del comando `test` con los operadores comparativos numéricos de Bash:

```
[user@host ~]$ test 1 -gt 0 ; echo $?  
0  
[user@host ~]$ test 0 -gt 1 ; echo $?  
1
```

Realice pruebas con la sintaxis del comando de prueba de Bash, `[ <TESTEXPRESSION> ]` o la sintaxis de comando de prueba extendida más reciente, `[[ <TESTEXPRESSION> ]]`, que proporciona características como el globbing de nombres de archivos y la coincidencia de patrones regex. En la mayoría de los casos, debe usar la sintaxis `[[ <TESTEXPRESSION> ]]`.

A continuación se ejemplifica la sintaxis de comandos de prueba y los operadores comparativos numéricos de Bash:

```
[user@host ~]$ [[ 1 -eq 1 ]]; echo $?  
0  
[user@host ~]$ [[ 1 -ne 1 ]]; echo $?  
1  
[user@host ~]$ [[ 8 -gt 2 ]]; echo $?  
0  
[user@host ~]$ [[ 2 -ge 2 ]]; echo $?  
0  
[user@host ~]$ [[ 2 -lt 2 ]]; echo $?
```

```
1
[user@host ~]$ [[ 1 -lt 2 ]]; echo $?
0
```

A continuación se ejemplifican los operadores comparativos de cadenas de Bash:

```
[user@host ~]$ [[ abc = abc ]]; echo $?
0
[user@host ~]$ [[ abc == def ]]; echo $?
1
[user@host ~]$ [[ abc != def ]]; echo $?
0
```

A continuación se ejemplifica el uso de operadores individuales (un argumento) de cadenas de Bash:

```
[user@host ~]$ STRING=''; [[ -z "$STRING" ]]; echo $?
0
[user@host ~]$ STRING='abc'; [[ -n "$STRING" ]]; echo $?
0
```



nota

Los caracteres de espacio dentro de los corchetes son obligatorios, porque separan las palabras y los elementos dentro de la expresión de prueba. La rutina de análisis de comandos de la shell divide los elementos de comandos en palabras y operadores al reconocer espacios y otros metacaracteres de acuerdo con reglas de análisis incorporadas. Para obtener un análisis completo de este concepto avanzado, consulte la página del manual `getopt(3)`. El carácter de corchete izquierdo (`[`) es en sí un alias incorporado para el comando `test`. Las palabras de shell, ya sean comandos, subcomandos, opciones, argumentos u otros elementos de token, están siempre delimitadas por espacios.

Estructuras condicionales

Los scripts de shell simples representan una recopilación de comandos que se ejecutan de principio a fin. Los programadores incorporan la toma de decisiones en los scripts de shell mediante el uso de estructuras condicionales. Un script puede ejecutar rutinas específicas cuando se cumplen las condiciones establecidas.

Uso de la construcción `if/then`

La estructura condicional más simple es la construcción *if/then*, que tiene la siguiente sintaxis:

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
fi
```

Con esta construcción, si el script cumple con la condición dada, ejecuta el código en el bloque de enunciado. No se realiza ninguna acción si la condición no se cumple. Las condiciones de

prueba comunes en los enunciados `if/then` incluyen las pruebas numéricas, de cadenas y de archivos analizadas anteriormente. En enunciado `fi` del final cierra la construcción `if/then`. En la siguiente sección de código, se demuestra el uso de una construcción `if/then` para iniciar el servicio `psacct` si no se encuentra activo:

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [[ $? -ne 0 ]]; then sudo systemctl start psacct; fi
```

Uso de la construcción If/Then/Else

Puede ampliar aún más la construcción `if/then` para que requiera diferentes conjuntos de acciones en función de si una condición se cumple o no. Use la construcción `if/then/else` para lograr este comportamiento, como en este ejemplo:

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
else
    <STATEMENT>
    ...
    <STATEMENT>
fi
```

En la siguiente sección de código, se demuestra una declaración `if/then/else` para iniciar el servicio `psacct` si no se encuentra activo y para detenerlo si está activo:

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [[ $? -ne 0 ]]; then \
sudo systemctl start psacct; \
else \
sudo systemctl stop psacct; \
fi
```

Uso de la construcción If/Then/Elif/Then/Else

Amplíe una construcción `if/then/else` para probar más de una condición y ejecutar un conjunto diferente de acciones cuando cumpla con una condición específica. El siguiente ejemplo muestra la construcción para una condición agregada:

```
if <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
elif <CONDITION>; then
    <STATEMENT>
    ...
    <STATEMENT>
else
    <STATEMENT>
```

```
...
<STATEMENT>
fi
```

En esta estructura condicional, Bash prueba la condición en el orden del script. Cuando una condición es verdadera, Bash ejecuta las acciones asociadas con la condición y, luego, omite el resto de la estructura condicional. Si ninguna condición es verdadera, Bash ejecuta las acciones en la cláusula `else`.

En el siguiente ejemplo, se demuestra el uso de una declaración `if/then/elif/then/else` para ejecutar el cliente `mysql` si el servicio `mariadb` está activo, para ejecutar el cliente `psql` si el servicio `postgresql` está activo o para ejecutar el cliente `sqlite3` si los servicios `mariadb` y `postgresql` no están activos.

```
[user@host ~]$ systemctl is-active mariadb > /dev/null 2>&1
[user@host ~]$ MARIADB_ACTIVE=$?
[user@host ~]$ sudo systemctl is-active postgresql > /dev/null 2>&1
[user@host ~]$ POSTGRESQL_ACTIVE=$?
[user@host ~]$ if [[ "$MARIADB_ACTIVE" -eq 0 ]]; then \
mysql; \
elif [[ "$POSTGRESQL_ACTIVE" -eq 0 ]]; then \
psql; \
else \
sqlite3; \
fi
```



Referencias

Página del manual: `bash(1)`

► Ejercicio Guiado

Bucles y construcciones condicionales en scripts

En este ejercicio, usa bucles para imprimir de manera eficiente el nombre de host desde varios servidores.

Resultados

- Cree un bucle `for` para iterar a través de una lista de ítems de la línea de comandos y en un script de shell.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start console-commands
```

Instrucciones

- Use los comandos `ssh` y `hostname` para imprimir el nombre de host de las máquinas `servera` y `serverb` en la salida estándar.

```
[student@workstation ~]$ ssh student@servera hostname
servera.lab.example.com
[student@workstation ~]$ ssh student@serverb hostname
serverb.lab.example.com
```

- Cree un bucle `for` para ejecutar el comando `hostname` en las máquinas `servera` y `serverb`.

```
[student@workstation ~]$ for HOST in servera serverb
do
ssh student@${HOST} hostname
done
servera.lab.example.com
serverb.lab.example.com
```

- Cree un script de shell en el directorio `/home/student/bin` para ejecutar el mismo bucle `for`. Asegúrese de que el script esté incluido en la variable de entorno `PATH`.
 - 3.1. Cree el directorio `/home/student/bin` para contener el script de shell, si aún no existe.

```
[student@workstation ~]$ mkdir ~/bin
```

- 3.2. Verifique que el subdirectorio **bin** de su directorio de inicio esté en su variable de entorno **PATH**.

```
[student@workstation ~]$ echo $PATH
/home/student/.local/bin:/home/student/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:/home/student/.venv/labs/bin
```

- 3.3. Cree un script de shell denominado **printhostname.sh** en el directorio **/home/student/bin** para realizar el bucle **for**. Use el comando **cat** para verificar el contenido del archivo **printhostname.sh**.

```
[student@workstation ~]$ vim ~/bin/printhostname.sh
[student@workstation ~]$ cat ~/bin/printhostname.sh
#!/usr/bin/bash
#Execute for loop to print server hostname.
for HOST in servera serverb
do
    ssh student@${HOST} hostname
done
exit 0
```

- 3.4. Otorgue permiso de ejecución al script creado.

```
[student@workstation ~]$ chmod +x ~/bin/printhostname.sh
```

- 3.5. Ejecute el script desde su directorio de inicio.

```
[student@workstation ~]$ printhostname.sh
servera.lab.example.com
serverb.lab.example.com
```

- 3.6. Verifique que el código de salida de su script sea 0.

```
[student@workstation ~]$ echo $?
0
```

Finalizar

En la máquina **workstation**, cambie al directorio de inicio del usuario **student** y use el comando **lab** para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish console-commands
```

Esto concluye la sección.

Texto coincidente en la salida de comando con expresiones regulares

Objetivos

Crear expresiones regulares para hacer coincidir datos, aplicar expresiones regulares a archivos de texto con el comando `grep` y usar `grep` para buscar archivos y datos de comandos canalizados.

Escritura de expresiones regulares

Las expresiones regulares proporcionan un mecanismo de coincidencia de patrones que facilita la búsqueda de contenido específico. Los comandos `vim`, `grep` y `less` pueden usar expresiones regulares. Los lenguajes de programación como Perl, Python y C también soportan expresiones regulares, pero pueden tener pequeñas diferencias en la sintaxis.

Las expresiones regulares son un lenguaje único, con su propia sintaxis y reglas. Esta sección presenta la sintaxis de expresiones regulares implementada en `bash`, con ejemplos.

Descripción de una expresión regular simple

La expresión regular más simple es una coincidencia exacta de la cadena para buscar. Una coincidencia exacta es cuando los caracteres de la expresión regular coinciden con el tipo y el orden de la cadena.

Imagine que un usuario está explorando el siguiente archivo en busca de todas las veces que aparece el patrón `cat`:

```
cat
dog
concatenate
dogma
category
educated
boondoggle
vindication
chilidog
```

La cadena `cat` es una coincidencia exacta del carácter `c`, seguida de los caracteres `a` y `t` sin otros caracteres en el medio. Buscar el archivo con la cadena `cat` como expresión regular arroja las siguientes coincidencias:

```
cat
concatenate
category
educated
vindication
```

Coincidencias con el inicio y el final de una línea

La expresión regular coincidiría con la cadena de búsqueda independientemente de en qué parte de la línea ocurrió: al principio, en el medio o al final de la palabra o línea. Use un metacarácter de *delimitador de línea* para controlar en qué parte de una línea buscar una coincidencia.

Para coincidir solo al principio de una línea, use el carácter de intercalación (^). Para coincidir al final de una línea, use el signo de dólar (\$).

Usando el mismo archivo del ejemplo anterior, la expresión regular `^cat` coincidiría con dos líneas.

```
cat
category
```

La expresión regular `cat$` no encontraría una sola coincidencia, donde los caracteres `cat` están al final de una línea.

```
cat
```

Ubique las líneas en el archivo que terminan con `dog` usando un delimitador de final de línea para crear la expresión regular `dog$`, que coincidirá con dos líneas:

```
dog
chilidog
```

Para localizar una línea que contenga solo la expresión de búsqueda exacta, use el tanto el delimitador del principio y el del final de línea. Por ejemplo, para localizar la palabra `cat` cuando está al principio y al final de una línea simultáneamente, use `^cat$`.

```
cat
```

Uso de comodines y multiplicadores en expresiones regulares

Las expresiones regulares usan un carácter de punto (.) como comodín para hacer coincidir cualquier carácter individual en una sola línea. La expresión regular de `c.t` busca una cadena que contenga una `c` seguida de un carácter individual seguido por una `t`. Ejemplos de coincidencias pueden incluir `cat`, `concatenate`, `vindication`, `cut` y `c$t`.

Al usar un comodín no restringido, no puede predecir el carácter que coincide con el comodín. Para hacer coincidir caracteres específicos, reemplace el comodín no restringido con caracteres apropiados.

Usar caracteres de corchetes, como la expresión regular `c[ao]t`, coincidirá con patrones que comiencen con una `c`, seguida de una `a`, una `o` o una `u`, seguida de una `t`. Las posibles expresiones coincidentes pueden tener las cadenas `cat`, `cot` y `cut`.

Los *multiplicadores* son un mecanismo usado a menudo con comodines. Los multiplicadores se aplican al carácter o comodín anterior en la expresión regular. Uno de los multiplicadores más comunes que se usan es el carácter de asterisco (*). Cuando se usa en una expresión regular, el multiplicador asterisco coincide con cero o más ocurrencias de la expresión multiplicada. Puede usar el asterisco con expresiones, además de caracteres.

Por ejemplo, la expresión regular `c[ao]*t` puede coincidir con `coat` o `coot`. Una expresión regular de `c.*t` coincide con `cat`, `coat`, `culvert` e incluso `ct` (coincide con cero caracteres

entre la c y la t). Cualquier cadena que comience con un c está seguida de cero o más caracteres y termina con un t será una coincidencia.

Otro tipo de multiplicador indicaría una cantidad más precisa de caracteres deseados en el patrón. Un ejemplo de un multiplicador explícito es la expresión regular 'c.\{2\}t', que coincide con cualquier palabra que comience con una c, seguida de exactamente dos caracteres y que termine con una t. La expresión 'c.\{2\}t' coincidiría con dos palabras en el siguiente ejemplo:

```
cat
coat
convert
cart
covert
cypher
```



nota

Este curso presentó dos mecanismos distintos de análisis de texto de metacaracteres: *coincidencia de patrones* de shell (también conocido como comodín de archivo o expansión de nombre de archivo) y *expresiones regulares*. Debido a que ambos sistemas usan metacaracteres similares, como el carácter de asterisco (*), pero tienen diferencias en la interpretación y las reglas de los metacaracteres, los dos mecanismos pueden resultar confusos hasta que se los domine a cada uno lo suficiente.

La coincidencia de patrones es una técnica de análisis de líneas de comandos diseñada para especificar fácilmente varios nombres de archivos y se soporta principalmente solo para representar patrones de nombres de archivos en la línea de comandos. Las expresiones regulares están diseñadas para representar cualquier forma o patrón en las cadenas de texto, sin importar cuán complejas sean. Las expresiones regulares son respaldadas internamente por numerosos comandos de procesamiento de texto, como `grep`, `sed`, `awk`, `python`, `perl` y muchas aplicaciones, con algunas variaciones dependientes del comando en las reglas de interpretación.

Expresiones regulares en Bash

Opción	Descripción
.	El punto (.) coincide con cualquier carácter individual.
?	El ítem anterior es opcional y coincide como máximo una vez.
*	El ítem anterior coincide cero o más veces.
+	El ítem anterior coincide una o más veces.
{n}	El ítem anterior coincide exactamente n veces.
{n,}	El ítem anterior coincide n o más veces.
{,m}	El ítem anterior coincide como máximo m veces.
{n,m}	El ítem anterior coincide como mínimo n veces y como máximo m veces.

Opción	Descripción
<code>[:alnum:]</code>	Caracteres alfanuméricos: <code>[:alpha:]</code> y <code>[:digit:]</code> ; en la configuración regional "C" y en la codificación de caracteres ASCII, esta expresión es lo mismo que <code>[0-9A-Za-z]</code> .
<code>[:alpha:]</code>	Caracteres alfabéticos: <code>[:lower:]</code> y <code>[:upper:]</code> ; en la configuración regional "C" y en la codificación de caracteres ASCII, esta expresión es lo mismo que <code>[A-Za-z]</code> .
<code>[:blank:]</code>	Caracteres en blanco: espacio y tabulación.
<code>[:cntrl:]</code>	Caracteres de control. En ASCII, estos caracteres tienen códigos octales del 000 al 037, y 177 (DEL).
<code>[:digit:]</code>	Dígitos: 0 1 2 3 4 5 6 7 8 9.
<code>[:graph:]</code>	Caracteres gráficos: <code>[:alnum:]</code> y <code>[:punct:]</code> .
<code>[:lower:]</code>	Letras minúsculas; en la configuración regional 'C' y en la codificación de caracteres ASCII: a b c d e f g h i j k l m n o p q r s t u v w x y z.
<code>[:print:]</code>	Caracteres imprimibles: <code>[:alnum:]</code> , <code>[:punct:]</code> y el espacio.
<code>[:punct:]</code>	Caracteres de puntuación; en la configuración regional 'C' y la codificación de caracteres ASCII: ! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ { } ~.
<code>[:space:]</code>	Caracteres de espacio: en la configuración regional 'C', esto es tabulación, nueva línea, tabulación vertical, avance de página, retorno de carro y espacio.
<code>[:upper:]</code>	Letras mayúsculas; en la configuración regional 'C' y en la codificación de caracteres ASCII: A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.
<code>[:xdigit:]</code>	Dígitos hexadecimales: 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.
<code>\b</code>	Coincidir con la cadena vacía en el extremo de una palabra.
<code>\B</code>	Coincidir con la cadena vacía, siempre que no sea en el extremo de una palabra.
<code>\<</code>	Coincidir con la cadena vacía al principio de una palabra.
<code>\></code>	Coincidir con la cadena vacía en el extremo de una palabra.
<code>\w</code>	Coincidir con el componente de la palabra. Sinónimo de <code>[_[:alnum:]]</code> .
<code>\W</code>	Coincidir con la ausencia de componente de la palabra. Sinónimo de <code>^[^[:alnum:]]</code> .
<code>\s</code>	Coincidir con el espacio en blanco. Sinónimo de <code>'[:space:]'</code> .
<code>\S</code>	Coincidir con la ausencia de espacio en blanco. Sinónimo de <code>^[^[:space:]]</code> .

Coincidir con expresiones regulares en la línea de comandos

El comando `grep` usa expresiones regulares para aislar los datos correspondientes. Puede usar el comando `grep` para hacer coincidir datos en un solo archivo o en varios archivos. Cuando usa `grep` para hacer coincidir datos en varios archivos, imprime el nombre del archivo seguido de dos puntos y luego las líneas que coinciden con la expresión regular.

Aislamiento de datos usando el comando `grep`

El comando `grep` especifica una expresión regular y un archivo para analizar las coincidencias.

```
[user@host ~]$ grep '^computer' /usr/share/dict/words
computer
computerese
computerise
computerite
computerizable
computerization
computerize
computerized
computerizes
computerizing
computerlike
computernik
computers
```



nota

Se recomienda usar comillas simples para encapsular la expresión regular para proteger cualquier metacarácter de shell (como `$`, `*` y `{}`). Encapsular la expresión regular asegura que los caracteres sean interpretados por el comando previsto y no por la shell.

El comando `grep` puede procesar la salida de otros comandos mediante el uso de un carácter operador de barra vertical (`|`). El siguiente ejemplo muestra el comando `grep` analizando líneas de la salida de otro comando.

```
[root@host ~]# ps aux | grep chrony
chrony      662  0.0  0.1 29440 2468 ?        S    10:56   0:00 /usr/sbin/chronyd
```

Opciones del comando `grep`

El comando `grep` tiene muchas opciones útiles para controlar cómo analiza las líneas.

Tabla de opciones de `grep` comunes

Opción	Función
<code>-i</code>	Usa la expresión regular proporcionada, pero no detecta diferencias entre mayúsculas y minúsculas.

Opción	Función
-v	Solo muestra las líneas que <i>no</i> contienen coincidencias con la expresión regular.
-r	Busca datos que coincidan con la expresión regular de forma recursiva en un grupo de archivos o directorios.
-A <i>NUMBER</i>	Muestra <i>NUMBER</i> (número) de líneas después de la coincidencia de la expresión regular.
-B <i>NUMBER</i>	Muestra <i>NUMBER</i> (número) de líneas antes de la coincidencia de la expresión regular.
-e	Si se usan varias opciones -e, se pueden suministrar varias expresiones regulares y se usarán con un OR lógico.

Vea las páginas `man` para buscar otras opciones para el comando `grep`.

Ejemplos del comando `grep`

Los siguientes ejemplos usan diversos archivos de configuración y de registro.

Las expresiones regulares distinguen entre minúsculas y mayúsculas de manera predeterminada. Use la opción `-i` del comando `grep` para ejecutar una búsqueda que distinga entre minúsculas y mayúsculas. El siguiente ejemplo muestra un extracto del archivo de configuración `/etc/httpd/conf/httpd.conf`:

```
[user@host ~]$ cat /etc/httpd/conf/httpd.conf
...output omitted...
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 80
...output omitted...
```

El siguiente ejemplo busca la expresión regular `serverroot` en el archivo de configuración `/etc/httpd/conf/httpd.conf`.

```
[user@host ~]$ grep -i serverroot /etc/httpd/conf/httpd.conf
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'
# with ServerRoot set to '/www' will be interpreted by the
# ServerRoot: The top of the directory tree under which the server's
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# same ServerRoot for multiple httpd daemons, you will need to change at
ServerRoot "/etc/httpd"
```

Use la opción `-v` del comando `grep` para *invertir* la búsqueda de la expresión regular. Esta opción solo muestra las líneas que *no* coinciden con la expresión regular.

En el siguiente ejemplo, se muestran todas las líneas, independientemente de las mayúsculas y las minúsculas, que *no* contienen la expresión regular `server`.

```
[user@host ~]$ grep -v -i server /etc/hosts
127.0.0.1 localhost.localdomain localhost
172.25.254.254 classroom.example.com classroom
172.25.254.254 content.example.com content
172.25.254.254 materials.example.com materials
### rht-vm-hosts file listing the entries to be appended to /etc/hosts

172.25.250.9 workstation.lab.example.com workstation
172.25.250.254 bastion.lab.example.com bastion
172.25.250.220 utility.lab.example.com utility
172.25.250.220 registry.lab.example.com registry
```

Para ver un archivo sin distraerse por las líneas de comentario, use la opción `-v` del comando `grep`. En el siguiente ejemplo, la expresión regular coincide con todas las líneas que comienzan con un carácter hash (`#`) o el carácter de punto y coma (`;`). Cualquiera de estos dos caracteres al comienzo de una línea indica un comentario que se omite de la salida.

```
[user@host ~]$ grep -v '^[#;]' /etc/ethertypes
IPv4  0800  ip ip4  # Internet IP (IPv4)
X25   0805
ARP   0806  ether-arp #
FR_ARP 0808      # Frame Relay ARP      [RFC1701]
```

El comando `grep` con la opción `-e` le permite buscar más de una expresión regular a la vez. En el siguiente ejemplo, que usa una combinación de los comandos `less` y `grep`, se localizan todas las apariciones de `pam_unix`, `user root` y `Accepted publickey` en el archivo de registro `/var/log/secure`.

```
[root@host ~]# cat /var/log/secure | grep -e 'pam_unix' \
-e 'user root' -e 'Accepted publickey' | less
Mar  4 03:31:41 localhost passwd[6639]: pam_unix(passwd:chauthtok): password
changed for root
Mar  4 03:32:34 localhost sshd[15556]: Accepted publickey for devops from
10.30.0.167 port 56472 ssh2: RSA SHA256:M8ikhCEDm2tQ95Z0o7ZvufqEixCFct
+wowZLNzNlBT0
Mar  4 03:32:34 localhost systemd[15560]: pam_unix(systemd-user:session): session
opened for user devops(uid=1001) by (uid=0)
```

Para buscar texto en un archivo que tiene abierto con los comandos `vim` o `less`, primero ingrese el carácter de barra (/) y luego escriba el patrón que desea buscar. Presione `Enter` para iniciar la búsqueda. Presione `N` para encontrar la siguiente coincidencia.

```
[root@host ~]# vim /var/log/boot.log
...output omitted...
[[0;32m OK ^[[0m] Finished ^[[0;1;39mdracut pre-pivot and cleanup hook^[[0m.^M
    Starting ^[[0;1;39mCleaning Up and Shutting Down Daemons^[[0m...^M
[[0;32m OK ^[[0m] Stopped target ^[[0;1;39mRemote Encrypted Volumes^[[0m.^M
[[0;32m OK ^[[0m] Stopped target ^[[0;1;39mTimer Units^[[0m.^M
[[0;32m OK ^[[0m] Closed ^[[0;1;39mD-Bus System Message Bus Socket^[[0m.^M
/Daemons
```

```
[root@host ~]# less /var/log/messages
...output omitted...
Mar  4 03:31:19 localhost kernel: pci 0000:00:02.0: vgaarb: setting as boot VGA
device
Mar  4 03:31:19 localhost kernel: pci 0000:00:02.0: vgaarb: VGA device added:
decodes=io+mem,owns=io+mem,locks=none
Mar  4 03:31:19 localhost kernel: pci 0000:00:02.0: vgaarb: bridge control
possible
Mar  4 03:31:19 localhost kernel: vgaarb: loaded
Mar  4 03:31:19 localhost kernel: SCSI subsystem initialized
Mar  4 03:31:19 localhost kernel: ACPI: bus type USB registered
Mar  4 03:31:19 localhost kernel: usbcore: registered new interface driver usbfs
Mar  4 03:31:19 localhost kernel: usbcore: registered new interface driver hub
Mar  4 03:31:19 localhost kernel: usbcore: registered new device driver usb
/device
```



Referencias

Páginas del manual `regex(7)` y `grep(1)`

► Ejercicio Guiado

Texto coincidente en la salida de comando con expresiones regulares

En este trabajo de laboratorio, busca texto en los registros del sistema y la salida de comandos para encontrar información de manera más eficiente.

Resultados

- Buscar de manera eficiente el texto en los archivos de registro y los archivos de configuración.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start console-regex
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Use el comando `grep` para encontrar el GID y el UID para los grupos `postfix` y `postdrop` y usuarios. Para ello, use el comando `rpm -q --scripts` que consulta la información de un paquete específico y muestra los scripts que se usan como parte del proceso de instalación.

```
[student@servera ~]$ rpm -q --scripts postfix | grep -e 'user' -e 'group'
# Add user and groups if necessary
/usr/sbin/groupadd -g 90 -r postdrop 2>/dev/null
/usr/sbin/groupadd -g 89 -r postfix 2>/dev/null
/usr/sbin/groupadd -g 12 -r mail 2>/dev/null
/usr/sbin/useradd -d /var/spool/postfix -s /sbin/nologin -g postfix -G mail -M -r
-u 89 postfix 2>/dev/null
setgid_group=postdrop \
```

- 3. Modifique la expresión regular anterior para mostrar los primeros dos mensajes en el archivo `/var/log/maillog`. En esta búsqueda, no está usando el carácter de intercalación (^) porque no está buscando el primer carácter en una línea.

```
[root@servera ~]# grep 'postfix' /var/log/maillog | head -n 2
Apr  1 15:27:16 servera postfix/postfix-script[3121]: starting the Postfix mail
system
Apr  1 15:27:16 servera postfix/master[3123]: daemon started -- version 3.5.9,
configuration /etc/postfix
```

- 4. Encuentre el nombre del directorio `queue` para el servidor `Postfix`. Busque el archivo de configuración `/etc/postfix/main.cf` para obtener toda la información sobre colas. Use la opción `-i` del comando `grep` para ignorar distinciones entre mayúsculas y minúsculas.

```
[root@servera ~]# grep -i 'queue' /etc/postfix/main.cf
# testing. When soft_bounce is enabled, mail will remain queued that
# The queue_directory specifies the location of the Postfix queue.
queue_directory = /var/spool/postfix
# QUEUE AND PROCESS OWNERSHIP
# The mail_owner parameter specifies the owner of the Postfix queue
# is the Sendmail-compatible mail queue listing command.
# setgid_group: The group for mail submission and queue management
```

- 5. Confirme que el servicio `postfix` escriba mensajes en el archivo `/var/log/messages`. Use el comando `less` y, luego, el carácter de barra (/) para buscar el archivo. Presione `n` para pasar a la siguiente entrada que coincida con la búsqueda. Presione `q` para salir del comando `less`.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Apr  1 15:27:15 servera systemd[1]: Starting Postfix Mail Transport Agent...
...output omitted...
Apr  1 15:27:16 servera systemd[1]: Started Postfix Mail Transport Agent.
...output omitted...
/Postfix
```

- 6. Use el comando `ps aux` para confirmar que el servidor `postfix` se está ejecutando actualmente. Use el comando `grep` para limitar la salida a las líneas necesarias.

```
[root@servera ~]# ps aux | grep postfix
root      3123  0.0  0.2 38172 4384 ?        Ss   15:27   0:00 /usr/
libexec/postfix/master -w
postfix   3124  0.0  0.4 45208 8236 ?        S    15:27   0:00 pickup -l -t
unix -u
postfix   3125  0.0  0.4 45252 8400 ?        S    15:27   0:00 qmgr -l -t unix
-u
root      3228  0.0  0.1 221668 2288 pts/0    S+   15:55   0:00 grep --
color=auto postfix
```

- 7. Confirme que las colas `qmgr`, `cleanup` y `pickup` estén configuradas correctamente. Use la opción `-e` del comando `grep` para hacer coincidir múltiples entradas en el mismo archivo. El `/etc/postfix/master.cf` archivo es el archivo de configuración

```
[root@servera ~]# grep -e qmgr -e pickup -e cleanup /etc/postfix/master.cf
pickup      unix  n      -      n      60      1      pickup
cleanup     unix  n      -      n      -       0      cleanup
qmgr        unix  n      -      n      300     1      qmgr
#qmgr       unix  n      -      n      300     1      oqmgr
```

- 8. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish console-regex
```

Esto concluye la sección.

► Trabajo de laboratorio

Mejora de la productividad con la línea de comandos

En este trabajo de laboratorio, crea un script de Bash que puede filtrar y obtener información relevante de diferentes hosts.

Resultados

- Crear un script Bash y redirigir su salida a un archivo.
- Usar bucles para simplificar su código.
- Filtrar el contenido relevante usando `grep` y expresiones regulares.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start console-review
```

Instrucciones

1. Cree el archivo de script `/home/student/bin/bash-lab` en la máquina `workstation`. El contenido inicial en el script será la directiva de intérprete shebang.
2. Edite su archivo de script recién creado para cumplir con la siguiente información solicitada de los hosts `servera` y `serverb`. Los sistemas usan claves SSH para la autenticación, por lo que no se necesita una contraseña.

Comando o archivo	Contenido solicitado
hostname -f	Obtenga toda la salida.
echo "#####"	Obtenga toda la salida.
lscpu	Obtenga solo las líneas que comienzan con la cadena CPU.
echo "#####"	Obtenga toda la salida.
/etc/selinux/config	Ignore las líneas vacías. Ignore las líneas que comienzan con #.
echo "#####"	Obtenga toda la salida.
/var/log/secure	Obtenga todas las entradas de "Failed password".
echo "#####"	Obtenga toda la salida.

Guarde la información requerida en los archivos `output-servera` y `output-serverb` en el directorio `/home/student` en `workstation`.



nota

Puede usar el comando `sudo` sin requerir una contraseña en los hosts `servera` y `serverb`. Recuerde usar un bucle para simplificar su script. También puede usar múltiples comandos `grep` concatenados con el uso del carácter de barra vertical (`|`).

- Ejecute el script `/home/student/bin/bash-lab` y revise el contenido de la salida en `workstation`.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade console-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish console-review
```

Esto concluye la sección.

► Solución

Mejora de la productividad con la línea de comandos

En este trabajo de laboratorio, crea un script de Bash que puede filtrar y obtener información relevante de diferentes hosts.

Resultados

- Crear un script Bash y redirigir su salida a un archivo.
- Usar bucles para simplificar su código.
- Filtrar el contenido relevante usando `grep` y expresiones regulares.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start console-review
```

Instrucciones

1. Cree el archivo de script `/home/student/bin/bash-lab` en la máquina `workstation`. El contenido inicial en el script será la directiva de intérprete shebang.

- 1.1. En la máquina `workstation`, cree el directorio `/home/student/bin/` si es necesario.

```
[student@workstation ~]$ mkdir -p /home/student/bin
```

- 1.2. Use el comando `vim` para crear y editar el archivo de script `/home/student/bin/bash-lab`.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

- 1.3. Inserte el siguiente texto y guarde el archivo.

```
#!/usr/bin/bash
```

- 1.4. Haga que su archivo de script sea ejecutable.

```
[student@workstation ~]$ chmod a+x ~/bin/bash-lab
```

2. Edite su archivo de script recién creado para cumplir con la siguiente información solicitada de los hosts **servera** y **serverb**. Los sistemas usan claves SSH para la autenticación, por lo que no se necesita una contraseña.

Comando o archivo	Contenido solicitado
<code>hostname -f</code>	Obtenga toda la salida.
<code>echo "#####"</code>	Obtenga toda la salida.
<code>lscpu</code>	Obtenga solo las líneas que comienzan con la cadena CPU.
<code>echo "#####"</code>	Obtenga toda la salida.
<code>/etc/selinux/config</code>	Ignore las líneas vacías. Ignore las líneas que comienzan con <code>#</code> .
<code>echo "#####"</code>	Obtenga toda la salida.
<code>/var/log/secure</code>	Obtenga todas las entradas de "Failed password".
<code>echo "#####"</code>	Obtenga toda la salida.

Guarde la información requerida en los archivos `output-servera` y `output-serverb` en el directorio `/home/student` en `workstation`.



nota

Puede usar el comando `sudo` sin requerir una contraseña en los hosts **servera** y **serverb**. Recuerde usar un bucle para simplificar su script. También puede usar múltiples comandos `grep` concatenados con el uso del carácter de barra vertical (`|`).

- 2.1. Use el comando `vim` para abrir y editar el archivo de script `/home/student/bin/bash-lab`.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

- 2.2. Agregue las siguientes líneas en negrita al archivo del script `/home/student/bin/bash-lab`.



nota

La siguiente salida es un ejemplo de cómo puede lograr el script solicitado. En scripting de Bash, puede adoptar diferentes enfoques y obtener el mismo resultado.

```
#!/usr/bin/bash
#
USR='student'
OUT='/home/student/output'
#
for SRV in servera serverb
do
ssh ${USR}@${SRV} "hostname -f" > ${OUT}-${SRV}
echo "#####" >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "lscpu | grep '^CPU'" >> ${OUT}-${SRV}
echo "#####" >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "grep -v '^$' /etc/selinux/config|grep -v '^#'" >> ${OUT}-${SRV}
echo "#####" >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "sudo grep 'Failed password' /var/log/secure" >> ${OUT}-${SRV}
echo "#####" >> ${OUT}-${SRV}
done
```

3. Ejecute el script `/home/student/bin/bash-lab` y revise el contenido de la salida en workstation.

3.1. En workstation, ejecute el script `/home/student/bin/bash-lab`.

```
[student@workstation ~]$ bash-lab
```

- 3.2. Revise el contenido de los archivos `/home/student/output-servera` y `/home/student/output-serverb`.

```
[student@workstation ~]$ cat /home/student/output-servera
servera.lab.example.com
#####
CPU op-mode(s):          32-bit, 64-bit
CPU(s):                  2
CPU family:              6
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Apr  1 05:42:07 servera sshd[1275]: Failed password for invalid user operator1
from 172.25.250.9 port 42460 ssh2
Apr  1 05:42:09 servera sshd[1277]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 42462 ssh2
Apr  1 05:42:11 servera sshd[1279]: Failed password for invalid user manager1 from
172.25.250.9 port 42464 ssh2
#####
[student@workstation ~]$ cat /home/student/output-serverb
serverb.lab.example.com
#####
CPU op-mode(s):          32-bit, 64-bit
CPU(s):                  2
CPU family:              6
#####
SELINUX=enforcing
```



```
SELINUXTYPE=targeted
#####
Apr  1 05:42:14 serverb sshd[1252]: Failed password for invalid user operator1
from 172.25.250.9 port 53494 ssh2
Apr  1 05:42:17 serverb sshd[1257]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 53496 ssh2
Apr  1 05:42:19 serverb sshd[1259]: Failed password for invalid user manager1 from
172.25.250.9 port 53498 ssh2
#####
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade console-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish console-review
```

Esto concluye la sección.

Resumen

- Crear y ejecutar scripts de Bash simples para realizar tareas de administración sencillas.
- Usar bucles para iterar a través de una lista de ítems de la línea de comandos y en un script de shell.
- Usar estructuras condicionales para incorporar la toma de decisiones en los scripts de shell.
- Buscar texto en los archivos de registro y configuración usando expresiones regulares y el comando `grep`.

Programación de tareas futuras

Meta

Programar tareas para que se ejecuten en una fecha y hora específicas.

Objetivos

- Configurar un comando para que se ejecute una vez en el futuro.
- Programar comandos para que se ejecuten en un horario de repetición usando el archivo `crontab` de un usuario.
- Programar comandos para que se ejecuten en un horario de repetición con el archivo `crontab` del sistema y directorios.
- Habilitar y deshabilitar los temporizadores de `systemd` y configurar un temporizador que administre archivos temporales.

Secciones

- Programación de un trabajo de usuario diferido (y ejercicio guiado)
- Programación de trabajos de usuario recurrentes (y ejercicio guiado)
- Programación de trabajos del sistema recurrentes (y ejercicio guiado)
- Administración de archivos temporales (y ejercicio guiado)
- Programación de tareas futuras (cuestionario)

Programación de un trabajo de usuario diferido

Objetivos

Configurar un comando para que se ejecute una vez en el futuro.

Descripción de tareas de usuario diferidas

A veces, es posible que deba ejecutar uno o más comandos en un momento futuro específico. Un ejemplo es un usuario que programa una tarea de mantenimiento de larga duración para que se realice en medio de la noche. Otro ejemplo es un administrador de sistemas que está trabajando en una configuración de firewall y pone en cola un trabajo de seguridad para restablecer la configuración del firewall a un estado de trabajo anterior en diez minutos, a menos que desactive el trabajo antes de que se ejecute, porque la nueva configuración de firewall funcionó.

Estos comandos programados a menudo se denominan *tareas* o *trabajos* y el termino *diferido* indica que estas tareas o trabajos se ejecutarán en el futuro.

Una solución disponible para que los usuarios de Red Hat Enterprise Linux programen tareas diferidas es el comando `at`, que está instalado y habilitado de manera predeterminada. El paquete `at` proporciona el daemon del sistema `atd` y los comandos `at` y `atq` para interactuar con el daemon.

Cualquier usuario puede poner en cola trabajos para el daemon `atd` con el comando `at`. El daemon `atd` proporciona 26 colas, identificados de la `a` a la `z`, con trabajos en colas ordenadas alfabéticamente que obtienen menos prioridad en el sistema (con valores *nice* más altos, se analizan en un capítulo posterior).

Programación de tareas de usuario diferidas

Use el comando `at` *TIMESPEC* para comenzar a ingresar un nuevo trabajo al programa. El comando `at` luego lee de STDIN (su teclado) para obtener los comandos para ejecutar. Al ingresar comandos manualmente, complete la entrada presionando `Ctrl+D` mientras está en una línea vacía. Puede usar la redirección de entrada desde un archivo de script para ingresar comandos más complejos. Por ejemplo, use el comando `at now +5min < myscript` para programar los comandos en `myscript` para que se inicien en 5 minutos, sin necesidad de escribir los comandos manualmente en una ventana de terminal.

El argumento *TIMESPEC* del comando `at` acepta especificaciones de tiempo natural para describir cuándo debe ejecutarse un trabajo. Por ejemplo, especifique una hora como `02:00pm`, `15:59`, `midnight` o incluso `teatime`, seguida de una fecha opcional o una cantidad de días en el futuro.

El argumento *TIMESPEC* espera especificaciones de fecha y hora en ese orden. Si proporciona la fecha y no la hora, la hora predeterminada es la hora actual. Si proporciona la hora y no la fecha, la fecha se considera coincidente y los trabajos se ejecutan cuando la hora coincide.

El siguiente ejemplo muestra una programación de trabajo sin proporcionar la fecha. El comando `at` programa el trabajo para hoy o mañana, según haya transcurrido el tiempo o no.

```
[user@host ~]$ date
Wed May 18 21:01:18 CDT 2022
[user@host ~]$ at 21:03 < myscript
job 3 at Wed May 18 21:03:00 2022
[user@host ~]$ at 21:00 < myscript
job 4 at Thu May 19 21:00:00 2022
```

Las páginas del manual para el comando `at` y otras fuentes de documentación usan minúsculas para escribir las especificaciones de tiempo natural, pero puede usar minúscula, mayúscula inicial o mayúscula. Estos son algunos ejemplos de especificaciones de tiempo que puede usar:

- `now +5min`
- `teatime tomorrow` (la hora del té es 16:00)
- `noon +4 days`
- `5pm august 3 2021`

Para obtener otras especificaciones de tiempo válidas, consulte el documento `timespec` local como se indica en las referencias.

Inspección y administración de trabajos de usuario diferidos

Para obtener una descripción general de los trabajos pendientes de su usuario actual, use el comando `atq` o `at -l`.

```
[user@host ~]$ atq
28 Mon May 16 05:13:00 2022 a user
29 Tue May 17 16:00:00 2022 h user
30 Wed May 18 12:00:00 2022 a user
```

En la salida anterior, cada línea representa un trabajo futuro programado diferente. La siguiente descripción se aplica a la primera línea de la salida:

- **28** es el número de trabajo único.
- **Mon May 16 05:13:00 2022** es la fecha y hora de ejecución del trabajo programado.
- **a** indica que el trabajo está programado con la cola predeterminada `a`.
- **user** es el propietario del trabajo (y el usuario con el cual se ejecutará el trabajo).



Importante

Los usuarios no privilegiados pueden ver y controlar sus propios trabajos. El usuario `root` puede ver y administrar todos los trabajos.

Use el comando `at -c JOBNUMBER` para inspeccionar los comandos que se ejecutan cuando el daemon `atd` ejecuta un trabajo. Este comando muestra el entorno del trabajo, que se establece desde el entorno del usuario cuando creó el trabajo, y la sintaxis del comando que se ejecutará.

Eliminación de trabajos del programa

El comando `atrm JOBNUMBER` elimina un trabajo programado. Elimine el trabajo programado cuando ya no lo necesita; por ejemplo, cuando una configuración de firewall remota dio un resultado satisfactorio, y no es necesario que se restablezca.



Referencias

Páginas del manual: `at(1)` y `atd(8)`

`/usr/share/doc/at/timespec`

► Ejercicio Guiado

Programación de un trabajo de usuario diferido

En este ejercicio, usa el comando `at` para programar varios comandos para ejecutar en momentos específicos en el futuro.

Resultados

- Programar un trabajo para que se ejecute en un momento específico en el futuro.
- Inspeccionar los comandos que ejecuta un trabajo programado.
- Eliminar los trabajos programados.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start scheduling-at
```

Instrucciones

- 1. En `workstation`, abra una sesión de SSH `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Programe un trabajo para ejecutar en tres minutos a partir de ahora. Guarde la salida del comando `date` en el archivo `/home/student/myjob.txt`.

- 2.1. Pase la cadena `date >> /home/student/myjob.txt` como entrada al comando `at` para que el trabajo se ejecute en tres minutos a partir de ahora.

```
[student@servera ~]$ echo "date >> /home/student/myjob.txt" | at now +3min
warning: commands will be executed using /bin/sh
job 1 at Mon Apr  4 05:00:00 2022
```

- 2.2. Detalle los trabajos programados.

```
[student@servera ~]$ atq
1 Mon Apr  4 05:00:00 2022 a student
```

- 2.3. Monitoree la cola de trabajos diferidos en tiempo real. Después de que se ejecuta, el daemon `atd` elimina el trabajo de la cola.

De manera predeterminada, el comando actualiza la salida del comando `atq` cada dos segundos. Después de que el daemon `atd` elimine el trabajo aplazado de la cola, presione `Ctrl+c` para salir del comando `watch` y volver al prompt de shell.

```
[student@servera ~]$ watch atq
Every 2.0s: atq                  servera.lab.example.com: Mon Apr  4 04:58:43 2022

1 Mon Apr  4 05:00:00 2022 a student
```

- 2.4. Verifique que el contenido de `/home/student/myjob.txt` coincida con la salida del comando `date`.

La salida coincide con la salida del comando `date`, lo cual confirma que el trabajo programado se ejecutó satisfactoriamente.

```
[student@servera ~]$ cat myjob.txt
Mon Apr  4 05:00:00 AM EDT 2022
```

- ▶ 3. Programe interactivamente un trabajo en la cola `g` que se ejecuta en `teatime` (16:00). El trabajo debe imprimir el mensaje `It's teatime` en el archivo `/home/student/tea.txt`. Anexe los mensajes nuevos al archivo `/home/student/tea.txt`.

```
[student@servera ~]$ at -q g teatime
warning: commands will be executed using /bin/sh
at> echo "It's teatime" >> /home/student/tea.txt
at> Ctrl+d
job 2 at Mon Apr  4 16:00:00 2022
```

- ▶ 4. Programe interactivamente otro trabajo con la cola `b` que se ejecuta en `16:05`. El trabajo debe imprimir el mensaje `The cookies are good` en el archivo `/home/student/cookies.txt`. Anexe los mensajes nuevos al archivo `/home/student/cookies.txt`.

```
[student@servera ~]$ at -q b 16:05
warning: commands will be executed using /bin/sh
at> echo "The cookies are good" >> /home/student/cookies.txt
at> Ctrl+d
job 3 at Mon Apr  4 16:05:00 2022
```

- ▶ 5. Inspeccione los comandos en los trabajos pendientes.

- 5.1. Vea los números de trabajo de los trabajos pendientes.

Observe los números de trabajo en la salida, que pueden variar en su sistema. Use los números de trabajo de su sistema.

```
[student@servera ~]$ atq
2 Mon Apr  4 16:00:00 2022 g student
3 Mon Apr  4 16:05:00 2022 b student
```

- 5.2. Vea los comandos del trabajo pendiente número 2.

El trabajo ejecuta un comando `echo` que anexa el mensaje `It's teatime` al archivo `/home/student/tea.txt`.


```
[student@servera ~]$ at -c 2
...output omitted...
echo "It's teatime" >> /home/student/tea.txt
marcinDELIMITER274c4275
```

5.3. Vea los comandos del trabajo pendiente número 3.

El trabajo ejecuta un comando `echo` que anexa el mensaje `The cookies are good` al archivo `/home/student/cookies.txt`.

```
[student@servera ~]$ at -c 3
...output omitted...
echo "The cookies are good" >> /home/student/cookies.txt
marcinDELIMITER73e30b75
```

- ▶ 6. Vea el número de trabajo de un trabajo que se ejecuta en `teatime` (16:00) y quítelo usando el comando `atrm`.

```
[student@servera ~]$ atq
2 Mon Apr 4 16:00:00 2022 g student
3 Mon Apr 4 16:05:00 2022 b student
[student@servera ~]$ atrm 2
```

- ▶ 7. Verifique que el trabajo programado para ejecutarse en `teatime` (16:00) ya no exista.

7.1. Vea la lista de trabajos pendientes y confirme que el trabajo programado para ejecutarse en `teatime` (16:00) ya no existe.

```
[student@servera ~]$ atq
3 Mon Apr 4 16:05:00 2022 b student
```

7.2. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish scheduling-at
```

Esto concluye la sección.

Programación de trabajos de usuario recurrentes

Objetivos

Programar comandos para que se ejecuten en un horario de repetición usando el archivo `crontab` de un usuario.

Descripción de trabajos de usuario recurrentes

Los trabajos *recurrentes* están programados para ejecutarse repetidamente. Los sistemas Red Hat Enterprise Linux proporcionan el daemon `crond`, que está habilitado e iniciado de forma predeterminada. El daemon `crond` lee múltiples archivos de configuración: uno por usuario y un conjunto de archivos en todo el sistema. Cada usuario tiene un archivo personal que edita con el comando `crontab -e`. Al ejecutar trabajos recurrentes, estos archivos de configuración proporcionan un control detallado a los usuarios y administradores. Si el trabajo programado no está escrito para usar el redireccionamiento, el daemon `crond` envía por correo electrónico cualquier salida o error generado al propietario del trabajo.

Programación de trabajos de usuario recurrentes

Use el comando `crontab` para administrar los trabajos programados. La siguiente lista muestra los comandos que puede usar un usuario local para administrar sus trabajos:

Ejemplos del comando `crontab`

Comando	Uso previsto
<code>crontab -l</code>	Detallar los trabajos para el usuario actual.
<code>crontab -r</code>	Eliminar todos los trabajos del usuario actual.
<code>crontab -e</code>	Editar trabajos para el usuario actual.
<code>crontab filename</code>	Eliminar todos los trabajos y reemplazar con los leídos de <i>nombre de archivo</i> . Este comando usa la entrada <code>stdin</code> cuando no se especifica ningún archivo.

Un usuario con privilegios puede usar la opción `-u` del comando `crontab` para administrar trabajos para otro usuario. El comando `crontab` nunca se usa para administrar trabajos del sistema y no se recomienda usar los comandos `crontab` como el usuario `root` debido a la capacidad de explotar trabajos personales configurados para ejecutarse como `root`. Dichos trabajos privilegiados deben configurarse como se describe en la sección posterior que describe los trabajos recurrentes del sistema.

Descripción del formato de trabajo del usuario

El comando `crontab -e` invoca el editor `vim` de manera predeterminada, a menos que la variable de entorno `EDITOR` esté configurada para otro editor. Cada trabajo debe usar una línea única en el archivo `crontab`. Siga estas recomendaciones para entradas válidas al escribir trabajos recurrentes:

- Líneas vacías para facilitar la lectura.
- Comentarios sobre las líneas que comienzan con el signo de número (#).
- Variables de entorno con formato `NAME=value`, que afectan a todas las líneas después de la línea donde se declaran.

La configuración de las variables estándares incluye la variable `SHELL` para declarar la shell que se usa para interpretar las líneas restantes del archivo `crontab`. La variable `MAILTO` determina quién debe recibir la salida enviada por correo electrónico.

**nota**

La capacidad de enviar un correo electrónico requiere una configuración adicional del sistema para un servidor de correo local o un retransmisor SMTP.

Los campos en el archivo `crontab` aparecen en el siguiente orden:

- Minutes (Minutos)
- Hours (Horas)
- Day of month (Día del mes)
- Month (Mes)
- Day of week (Día de la semana)
- Comando

El comando se ejecuta cuando los campos *Day of the month* (Día del mes) o *Day of the week* (Día de la semana) usan el mismo valor que no sea el carácter *. Por ejemplo, para ejecutar un comando el día 11 de cada mes y todos los viernes a las 12:15 (formato de 24 horas), use el siguiente formato de trabajo:

```
15 12 11 * Fri command
```

Los primeros cinco campos usan las mismas reglas de sintaxis:

- Use el carácter * para ejecutar en todas las instancias posibles del campo.
- Un número para especificar la cantidad de minutos u horas, una fecha o un día de la semana. Para los días de semana, 0 equivale a domingo, 1 equivale a lunes, 2 equivale a martes, etc. 7 también equivale a domingo.
- Use x-y para un rango, que incluye los valores x y y.
- Use x, y para listas. Las listas también pueden incluir rangos, por ejemplo, 5, 10-13, 17 en la columna **Minutes**, para indicar que un trabajo debe ejecutarse a los 5 minutos, a los 10 minutos, a los 11 minutos, a los 12 minutos, a los 13 minutos y a los 17 minutos de la hora.
- */x indica un intervalo de x, por ejemplo, */7 en la columna **Minutes** ejecuta un trabajo exactamente cada siete minutos.

Asimismo, se usan abreviaturas en inglés de tres letras para los meses o los días de la semana, por ejemplo, Jan (enero), Feb (febrero) y Mon (lunes), Tue (martes).

El último campo contiene el comando completo con opciones y argumentos que se ejecutará usando la shell predeterminada. Si el comando contiene un símbolo de porcentaje no codificado (%), ese símbolo de porcentaje se tratará como el carácter de una línea nueva, y todo lo que esté después del símbolo de porcentaje se enviará al comando como enterada `stdin`.

Ejemplos de trabajos de usuario recurrentes

El siguiente trabajo ejecuta el comando `/usr/local/bin/yearly_backup` exactamente a las 9:00 a. m. el 3 de febrero, todos los años. Febrero se representa como el número 2 en el ejemplo, ya que es el segundo mes del año.

```
0 9 3 2 * /usr/local/bin/yearly_backup
```

El siguiente trabajo envía un correo electrónico que contiene la palabra **Chime** al propietario de este trabajo, cada cinco minutos entre las 9:00 a. m. y las 4:00 p. m., pero solo en cada viernes de julio.

```
* /5 9-16 * Jul 5 echo "Chime"
```

El rango de horas 9-16 precedente significa que el temporizador de trabajo comienza a la novena hora (09:00) y continúa hasta el final de la decimosexta hora (16:59). El trabajo comienza a ejecutarse a las 09:00 con la última ejecución a las 16:55 porque a cinco minutos de las 16:55 es 17:00 que está más allá del alcance de las horas dadas.

Si un rango es específico para las horas en lugar de un valor único, todas las horas dentro del rango coincidirán. Por lo tanto, con las horas de 9-16, este ejemplo coincide cada cinco minutos desde las 09:00 hasta las 16:55.



nota

Este trabajo de ejemplo envía la salida como un correo electrónico porque **crond** reconoce que el trabajo permitió que la salida fuera al canal **STDIO** sin redirección. Dado que los trabajos cron se ejecutan en un entorno en segundo plano sin un dispositivo de salida (conocido como *terminal de control*), **crond** almacena en búfer la salida y crea un correo electrónico para enviarlo al usuario especificado en la configuración. Para los trabajos del sistema, el correo electrónico se enviará a la cuenta **root**.

El siguiente trabajo ejecuta el comando `/usr/local/bin/daily_report` todos los días hábiles (de lunes a viernes) dos minutos antes de la medianoche.

```
58 23 * * 1-5 /usr/local/bin/daily_report
```

El siguiente trabajo ejecuta el comando **mutt** para enviar el mensaje de correo **Checking in** al destinatario **developer@example.com** en cada día de trabajo (lunes a viernes), a las 9:00 a. m.

```
0 9 * * 1-5 mutt -s "Checking in" developer@example.com % Hi there, just checking in.
```



Referencias

Páginas del manual: **crond(8)**, **crontab(1)**, **crontab(5)**

► Ejercicio Guiado

Programación de trabajos de usuario recurrentes

En este ejercicio, programará comandos para ejecutar en un programa de repetición como un usuario no privilegiado con el comando `crontab`.

Resultados

- Programar trabajos recurrentes para ejecutar como usuario no privilegiado.
- Inspeccionar los comandos que ejecuta un trabajo recurrente programado.
- Eliminar trabajos recurrentes programados.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start scheduling-cron
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Programe un trabajo recurrente como el usuario `student` que anexa la fecha y hora actuales al archivo `/home/student/my_first_cron_job.txt` cada dos minutos. El trabajo debe ejecutarse solo de lunes a viernes, no los sábados ni los domingos.



Importante

Si está trabajando en este ejercicio fuera de los días especificados en la instrucción anterior, ajuste la hora o la fecha del sistema en consecuencia para que el trabajo se ejecute mientras está trabajando.

- 2.1. Abra el archivo `crontab` con el editor de textos predeterminado.

```
[student@servera ~]$ crontab -e
```

- 2.2. Inserte la siguiente línea:

```
* /2 * * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

- 2.3. Presione **Esc** y escriba **:wq** para guardar los cambios y salir del editor. Cuando el editor sale, debe ver la siguiente salida:

```
...output omitted...
crontab: installing new crontab
[student@servera ~]$
```

- ▶ 3. Use el comando **crontab -l** para enumerar los trabajos recurrentes programados. Inspeccione el comando que programó para que se ejecute como un trabajo recurrente en el paso anterior.

Verifique que el trabajo ejecute el comando **/usr/bin/date** y anexe su salida al archivo **/home/student/my_first_cron_job.txt**.

```
[student@servera ~]$ crontab -l
*/2 * * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

- ▶ 4. Coloque su prompt de shell en espera hasta que se cree el archivo **/home/student/my_first_cron_job.txt** como resultado de la ejecución exitosa del trabajo recurrente que programó. Espere a que regrese su prompt de shell.

El comando **while** usa **! test -f** para continuar ejecutando un bucle y está en reposo por un segundo hasta que se crea el archivo **my_first_cron_job.txt** en el directorio **/home/student**.

```
[student@servera ~]$ while ! test -f my_first_cron_job.txt; do sleep 1s; done
```

- ▶ 5. Verifique que el contenido de **/home/student/my_first_cron_job.txt** coincida con la salida del comando **date**.

```
[student@servera ~]$ cat my_first_cron_job.txt
Mon Apr  4 03:04:01 AM EDT 2022
```

- ▶ 6. Elimine todos los trabajos recurrentes programados para el usuario **student**.

6.1. Elimine todos los trabajos recurrentes programados para el usuario **student**.

```
[student@servera ~]$ crontab -r
```

6.2. Verifique que no existen trabajos recurrentes para el usuario **student**.

```
[student@servera ~]$ crontab -l
no crontab for student
```

6.3. Regrese a la máquina **workstation** como el usuario **student**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish scheduling-cron
```

Esto concluye la sección.

Programación de trabajos del sistema recurrentes

Objetivos

Programar comandos para que se ejecuten en un horario de repetición con el archivo `crontab` del sistema y directorios.

Trabajos del sistema recurrentes

Los administradores de sistemas a menudo necesitan ejecutar trabajos recurrentes. Lo mejor es ejecutar estos trabajos desde cuentas del sistema en lugar de desde cuentas de usuario. Programe estos trabajos con archivos `crontab` de todo el sistema en lugar de con el comando `crontab`. Las entradas de trabajo en los archivos `crontab` de todo el sistema son similares a las entradas `crontab` de los usuarios. Los archivos `crontab` de todo el sistema tienen un campo adicional antes del campo `command` para especificar el usuario que ejecuta el comando.

El archivo `/etc/crontab` tiene un diagrama de sintaxis útil en los comentarios.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
# | | | | | sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed
```

El archivo `/etc/crontab` y otros archivos en el directorio `/etc/cron.d/` definen los trabajos recurrentes del sistema. Siempre debe crear sus archivos `crontab` personalizados en el directorio `/etc/cron.d/` para programar trabajos recurrentes del sistema. Coloque el archivo `crontab` personalizado en el directorio `/etc/cron.d` para evitar que una actualización de paquete sobrescriba el archivo `/etc/crontab`. Los paquetes que requieren trabajos recurrentes del sistema colocan sus archivos `crontab` en el directorio `/etc/cron.d/` que contiene las entradas de trabajo. Los administradores también usan esta ubicación para agrupar trabajos relacionados en un solo archivo.

El sistema `crontab` también incluye repositorios para scripts para ejecutarse cada hora, día, semana y mes. Estos repositorios están ubicados en los directorios `/etc/cron.hourly/`, `/etc/cron.daily/`, `/etc/cron.weekly/` y `/etc/cron.monthly/`. Estos directorios contienen scripts de shell ejecutables, no archivos `crontab`.

**nota**

Use el comando `chmod +x script_name` para hacer un script ejecutable. Si un script no es ejecutable, no se ejecuta.

Ejecución de comandos periódicos con Anacron

El comando `run-parts` también ejecuta los trabajos diarios, semanales y mensuales desde el archivo de configuración `/etc/anacrontab`.

El archivo `/etc/anacrontab` garantiza que los trabajos programados se ejecuten siempre y que no se omitan accidentalmente porque el sistema se apagó o se hibernó. Por ejemplo, si un trabajo del sistema que se ejecuta diariamente no se ejecutó en un momento específico debido a que el sistema se estaba reiniciando, el trabajo se completa cuando el sistema está listo. Puede ocurrir un retraso antes de que se inicie el trabajo, si se especifica en el parámetro `Delay in minutes` en el archivo `/etc/anacrontab`.

Los archivos en el directorio `/var/spool/anacron/` determinan los trabajos diarios, semanales y mensuales. Cuando el daemon `crond` comienza un trabajo desde `/etc/anacrontab`, actualiza los sellos de hora de esos archivos. Con este sello de hora, puede determinar la última vez que se ejecutó el trabajo. La sintaxis del archivo `/etc/anacrontab` es diferente a la de los archivos de configuración de `crontab` regulares. El archivo `/etc/anacrontab` contiene cuatro campos por línea, como se detalla a continuación.

Period in days

Define el intervalo en días para el trabajo que se ejecuta en una programación recurrente. Este campo acepta un valor entero o macro. Por ejemplo, la macro `@daily` es equivalente al entero 1, lo que ejecuta el trabajo diariamente. De manera similar, la macro `@weekly` es equivalente al entero 7, lo que ejecuta el trabajo semanalmente.

Delay in minutes

Define el tiempo que el daemon `crond` debe esperar antes de iniciar el trabajo.

Job identifier

Este campo identifica el nombre único del trabajo en los mensajes de registro.

Command

El comando que se ejecutará.

El archivo `/etc/anacrontab` también contiene declaraciones variables del entorno con la sintaxis `NAME=value`. La variable `START_HOURS_RANGE` especifica el intervalo de tiempo para que se ejecuten los trabajos. Los trabajos no se inician fuera de este rango. Cuando un trabajo no se ejecuta dentro de este intervalo de tiempo en un día particular, el trabajo tiene que esperar hasta el día siguiente para su ejecución.

Temporizador systemd

La unidad de temporizador `systemd` activa otra unidad de un tipo diferente (como un servicio) cuyo nombre de unidad coincide con el nombre de la unidad del temporizador. La unidad del temporizador permite la activación basada en el temporizador de otras unidades. El temporizador `systemd` registra los eventos del temporizador en los diarios (journals) del sistema para una depuración más fácil.

Unidad de temporizador de muestra

El paquete `sysstat` proporciona la unidad de temporizador `systemd`, llamada el servicio `sysstat-collect.timer`, para recopilar estadísticas del sistema cada 10 minutos. En la siguiente salida, se muestra el contenido del archivo de configuración `/usr/lib/systemd/system/sysstat-collect.timer`.

```
...output omitted...
[Unit]
Description=Run system activity accounting tool every 10 minutes

[Timer]
OnCalendar=*:00/10

[Install]
WantedBy=sysstat.service
```

La opción `OnCalendar=*:00/10` significa que esta unidad de temporizador activa la unidad `sysstat-collect.service` correspondiente cada 10 minutos. Puede especificar intervalos de tiempo más complejos.

Por ejemplo, un valor de `2022-04-* 12:35,37,39:16` frente a la opción `OnCalendar` hace que la unidad del temporizador active la unidad de servicio correspondiente a la hora `12:35:16`, `12:37:16` y `12:39:16`, todos los días durante abril de 2022. También puede especificar temporizadores relativos con la opción `OnUnitActiveSec`. Por ejemplo, con la opción `OnUnitActiveSec=15min`, la unidad del temporizador desencadena la unidad correspondiente 15 minutos después de la última vez que la unidad del temporizador activó su unidad correspondiente.



Importante

No modifique ninguna unidad en los archivos de configuración bajo el directorio `/usr/lib/systemd/system` porque la unidad `systemd` anula los cambios de configuración que realizó en ese archivo. Cree una copia del archivo de configuración en el directorio `/etc/systemd/system`, luego, modifique el archivo copiado para evitar que cualquier actualización del paquete proveedor anule los cambios. Si existen dos archivos con el mismo nombre bajo los directorios `/usr/lib/systemd/system` y `/etc/systemd/system`, la unidad del temporizador `systemd` analiza el archivo bajo el directorio `/etc/systemd/system`.

Después de cambiar el archivo de configuración de la unidad del temporizador, use el comando `systemctl daemon-reload` para asegurar que la unidad del temporizador `systemd` cargue los cambios.

```
[root@host ~]# systemctl daemon-reload
```

Después de recargar la configuración del daemon `systemd`, use el comando `systemctl` para activar la unidad del temporizador.

```
[root@host ~]# systemctl enable --now <unitname>.timer
```



Referencias

Páginas del manual `crontab(5)`, `anacron(8)`, `anacrontab(5)`, `systemd.time(7)`, `systemd.timer(5)` y `crond(8)`

► Ejercicio Guiado

Programación de trabajos del sistema recurrentes

En este ejercicio, programa comandos para que se ejecuten en diversos horarios agregando archivos de configuración a los directorios crontab del sistema.

Resultados

- Programar un trabajo del sistema recurrente para contar el número de usuarios activos
- Actualizar la unidad de temporizador `systemd` que recopila los datos de actividad del sistema.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start scheduling-system
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Programe un trabajo del sistema recurrente que genere un mensaje de registro que indique la cantidad de usuarios activos en el sistema. Este trabajo debe ejecutarse diariamente y usar el comando `w -h | wc -l` para recuperar el número de usuarios activos en el sistema. Use el comando `logger` para generar el mensaje de registro de los usuarios activos actualmente.

- 2.1. Cree el archivo de script `/etc/cron.daily/usercount` con el siguiente contenido:

```
#!/bin/bash
USERCOUNT=$(w -h | wc -l)
logger "There are currently ${USERCOUNT} active users"
```

- 2.2. Haga el archivo de script ejecutable.

```
[root@servera ~]# chmod +x /etc/cron.daily/usercount
```

- 3. Instale el paquete `sysstat`. La unidad del temporizador debe desencadenar la unidad de servicio cada 10 minutos para recopilar datos de actividad del sistema con el script de shell `/usr/lib64/sa/sa1`. Modifique el archivo de configuración de la unidad del temporizador para recopilar los datos de la actividad del sistema cada dos minutos.

3.1. Instale el paquete `sysstat`.

```
[root@servera ~]# dnf install sysstat
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

3.2. Copie el archivo `/usr/lib/systemd/system/sysstat-collect.timer` en el archivo `/etc/systemd/system/sysstat-collect.timer`.

```
[root@servera ~]# cp /usr/lib/systemd/system/sysstat-collect.timer \
/etc/systemd/system/sysstat-collect.timer
```

3.3. Edite el archivo `/etc/systemd/system/sysstat-collect.timer` para que la unidad del temporizador funcione cada dos minutos. Reemplace todas las apariciones de la cadena `10 minutes` con `2 minutes` en todo el archivo de configuración de la unidad, incluidas las apariciones de las líneas comentadas. Use el comando `vim /etc/systemd/system/sysstat-collect.timer` para editar el archivo de configuración.

A partir de estos cambios, la unidad `sysstat-collect.timer` desencadena la unidad `sysstat-collect.service` cada dos minutos y recopila los datos de actividad del sistema en un archivo binario en el directorio `/var/log/sa`.

```
...output omitted...
# Activates activity collector every 2 minutes

[Unit]
Description=Run system activity accounting tool every 2 minutes

[Timer]
OnCalendar=*:00/2

[Install]
WantedBy=sysstat.service
```

3.4. Haga que el daemon `systemd` conozca los cambios.

```
[root@servera ~]# systemctl daemon-reload
```

3.5. Active la unidad `sysstat-collect.timer`.

```
[root@servera ~]# systemctl enable --now sysstat-collect.timer
...output omitted...
```

- 3.6. Espere hasta que se cree el archivo binario en el directorio `/var/log/sa`.

El comando `while`, `ls /var/log/sa | wc -l` devuelve 0 cuando el archivo no existe, o devuelve 1 cuando el archivo existe. El comando `while` se detiene durante un segundo cuando el archivo no está presente. El bucle `while` sale cuando el archivo está presente.

```
[root@servera ~]# while [ $(ls /var/log/sa | wc -l) -eq 0 ]; \
do sleep 1s; done
```

- 3.7. Verifique que el archivo binario en el directorio `/var/log/sa` se haya modificado en los últimos dos minutos.

```
[root@servera ~]# ls -l /var/log/sa
total 4
-rw-r--r--. 1 root root 2540 Apr  5 04:08 sa05
[root@servera ~]# date
Tue Apr  5 04:08:29 AM EDT 2022
```

- 3.8. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish scheduling-system
```

Esto concluye la sección.

Administración de archivos temporales

Objetivos

Habilitar y deshabilitar los temporizadores de `systemd` y configurar un temporizador que administre archivos temporales.

Administración de archivos temporales

La mayoría de las aplicaciones y servicios críticos usan archivos y directorios temporales. Algunas aplicaciones (y usuarios) usan el directorio `/tmp` para almacenar datos de trabajo transitorios, mientras que otras aplicaciones usan ubicaciones específicas de la tarea como directorios volátiles `daemon` y específicos del usuario en `/run`, que existe solo en la memoria. Cuando el sistema se reinicia o pierde energía, los sistemas de archivos basados en memoria se limpian automáticamente.

Por lo general, los daemons y los scripts funcionan correctamente solo cuando existen los archivos y directorios temporales esperados. Además, es necesario purgar los archivos temporales ubicados en el almacenamiento persistente para evitar problemas de espacio en disco o datos de trabajo obsoletos.

Red Hat Enterprise Linux incluye la herramienta `systemd-tmpfiles`, que proporciona un método estructurado y configurable para administrar directorios y archivos temporales.

En el arranque del sistema, una de las primeras unidades de servicio `systemd` lanzadas es el servicio `systemd-tmpfiles-setup`. Este servicio ejecuta las opciones `--create --remove` del comando `systemd-tmpfiles`, que lee las instrucciones de los archivos de configuración `/usr/lib/tmpfiles.d/*.conf`, `/run/tmpfiles.d/*.conf` y `/etc/tmpfiles.d/*.conf`. Estos archivos de configuración enumeran los archivos y directorios que el servicio `systemd-tmpfiles-setup` debe crear, eliminar o proteger con permisos.

Limpieza de archivos temporales con un temporizador de systemd

Para evitar que los sistemas de larga duración llenen sus discos con datos obsoletos, una unidad de temporizador `systemd` denominada `systemd-tmpfiles-clean.timer` desencadena `systemd-tmpfiles-clean.service` en un intervalo regular, que ejecuta el comando `systemd-tmpfiles --clean`.

La configuración de una unidad de temporizador `systemd` tiene una sección `[Timer]` para indicar cómo iniciar el servicio con el mismo nombre que el temporizador.

Use el siguiente comando `systemctl` para ver el contenido del archivo de configuración de la unidad `systemd-tmpfiles-clean.timer`.

```
[user@host ~]$ systemctl cat systemd-tmpfiles-clean.timer
# /usr/lib/systemd/system/systemd-tmpfiles-clean.timer
# SPDX-License-Identifier: LGPL-2.1-or-later
#
# This file is part of systemd.
#
```

```
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.

[Unit]
Description=Daily Cleanup of Temporary Directories
Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)
ConditionPathExists=!/etc/initrd-release

[Timer]
OnBootSec=15min
OnUnitActiveSec=1d
```

En la configuración anterior, el parámetro `OnBootSec=15min` indica que la unidad `systemd-tmpfiles-clean.service` se desencadena 15 minutos después de que el sistema se haya iniciado. El parámetro `OnUnitActiveSec=1d` indica que cualquier otro desencadenador para la unidad `systemd-tmpfiles-clean.service` ocurre 24 horas después de que la unidad de servicio se activó por última vez.

Cambie los parámetros en el archivo de configuración de la unidad del temporizador `systemd-tmpfiles-clean.timer`. Por ejemplo, un valor `30min` para el parámetro `OnUnitActiveSec` desencadena la unidad de servicio `systemd-tmpfiles-clean.service` 30 minutos después de la última activación de la unidad de servicio. Como resultado, `systemd-tmpfiles-clean.service` se activa cada 30 minutos después de que se reconozcan los cambios.

Después de cambiar el archivo de configuración de la unidad del temporizador, use el comando `systemctl daemon-reload` para asegurar que `systemd` cargue la nueva configuración.

```
[root@host ~]# systemctl daemon-reload
```

Después de recargar la configuración del administrador `systemd`, use el siguiente comando `systemctl` para activar la unidad `systemd-tmpfiles-clean.timer`.

```
[root@host ~]# systemctl enable --now systemd-tmpfiles-clean.timer
```

Limpieza manual de archivos temporales

El comando `systemd-tmpfiles --clean` analiza los mismos archivos de configuración que el comando `systemd-tmpfiles --create`, pero en lugar de crear archivos y directorios, purgará todos los archivos a los que no se haya accedido o que no hayan sido modificados ni cambiados en una fecha anterior a la antigüedad máxima definida en el archivo de configuración.

Encuentre información detallada acerca del formato de los archivos de configuración para el servicio `systemd-tmpfiles` en la página del manual `tmpfiles.d(5)`. La sintaxis básica consta de las siguientes columnas: Tipo, Ruta, Modo, UID, GID, Edad y Argumento. Tipo se refiere a la acción que el servicio `systemd-tmpfiles` debe realizar; por ejemplo, `d` para crear un directorio si aún no existe o `Z` para restaurar recursivamente contextos de SELinux, permisos de archivos y propiedad.

A continuación, se dan ejemplos de configuración de purgación con explicaciones:

```
d /run/systemd/seats 0755 root root -
```


Al crear archivos y directorios, cree el directorio `/run/systemd/seats` si no existe, con el usuario `root` y el grupo de propietarios `root`, y con permisos de `rw-r-x`. Si este directorio existe, no realice ninguna acción. El servicio `systemd-tmpfiles` no purga este directorio automáticamente.

```
D /home/student 0700 student student 1d
```

Cree el directorio `/home/student` si aún no existe. Si existe, vacíele todo su contenido. Cuando el sistema ejecuta el comando `systemd-tmpfiles --clean`, elimina todos los archivos en el directorio a los que no accedió, que no cambió ni modificó durante más de un día.

```
L /run/fstablink - root root - /etc/fstab
```

Cree el enlace simbólico `/run/fstablink` para apuntar a la carpeta `/etc/fstab`. Nunca purgue automáticamente esta línea.

Precedencia de archivos de configuración

Los archivos de configuración `systemd-tmpfiles-clean` pueden encontrarse en tres lugares:

- `/etc/tmpfiles.d/*.conf`
- `/run/tmpfiles.d/*.conf`
- `/usr/lib/tmpfiles.d/*.conf`

Use los archivos en el directorio `/etc/tmpfiles.d/` para configurar ubicaciones temporales personalizadas y anular los valores predeterminados provistos por el proveedor. Los archivos en el directorio `/run/tmpfiles.d/` son archivos volátiles, normalmente usados por daemons para administrar sus propios archivos temporales de tiempo de ejecución. Los paquetes RPM relevantes proporcionan los archivos en el directorio `/usr/lib/tmpfiles.d/`; por lo tanto, no edite estos archivos.

Si un archivo en el directorio `/run/tmpfiles.d/` tiene el mismo nombre de archivo que un archivo en el directorio `/usr/lib/tmpfiles.d/`, el servicio usa el archivo en el directorio `/run/tmpfiles.d/`. Si un archivo en el directorio `/etc/tmpfiles.d/` tiene el mismo nombre de archivo que un archivo en el directorio `/run/tmpfiles.d/` o `/usr/lib/tmpfiles.d/`, el servicio usa el archivo en el directorio `/etc/tmpfiles.d/`.

Dadas estas reglas de precedencia, puede reemplazar fácilmente la configuración proporcionada por el proveedor si copia el archivo relevante en el directorio `/etc/tmpfiles.d/` y, luego, lo edita. Al usar estas ubicaciones de configuración correctamente, puede administrar los ajustes configurados por el administrador desde un sistema de administración de configuración central y las actualizaciones de paquetes no sobrescribirán los ajustes configurados.



nota

Cuando evalúe configuraciones nuevas o modificadas, es útil solo aplicar los comandos de un solo archivo de configuración a la vez. Especifique el nombre del archivo de configuración en la línea de comandos `systemd-tmpfiles`.



Referencias

Páginas del manual: `systemd-tmpfiles(8)`, `tmpfiles.d(5)`, `stat(1)`, `stat(2)` y `systemd.timer(5)`.

► Ejercicio Guiado

Administración de archivos temporales

En este ejercicio, usted configura `systemd-tmpfiles` para cambiar la rapidez con la que elimina los archivos temporales de `/tmp`, y también para purgar periódicamente los archivos de otro directorio.

Resultados

- Configurar `systemd-tmpfiles` para eliminar archivos temporales no usados de `/tmp`.
- Configurar `systemd-tmpfiles` para purgar periódicamente archivos de otro directorio.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start scheduling-tmpfiles
```

Instrucciones

- 1. Inicie sesión en el sistema `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Configure el servicio `systemd-tmpfiles` para limpiar el directorio `/tmp` de los archivos no usados de los últimos cinco días. Asegúrese de que una actualización del paquete no sobrescriba los archivos de configuración.

- 2.1. Copie el archivo `/usr/lib/tmpfiles.d/tmp.conf` en el directorio `/etc/tmpfiles.d`.

```
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf \
/etc/tmpfiles.d/tmp.conf
```

- 2.2. Busque la línea de configuración en el archivo `/etc/tmpfiles.d/tmp.conf` que se aplica al directorio `/tmp`. Reemplace la edad existente de los archivos temporales en esa línea de configuración con la nueva edad de 5 días. Elimine todas las otras líneas del archivo, incluidas las líneas comentadas. Puede usar el comando `vim /etc/tmpfiles.d/tmp.conf` para editar el archivo de configuración.

En la configuración, el tipo `q` es idéntico al tipo `d` e indica al servicio `systemd-tmpfiles` que cree el directorio `/tmp` si no existe. Los permisos octales del directorio deben estar establecidos en `1777`. Tanto el usuario propietario como el grupo del directorio `/tmp` deben ser `root`. El directorio `/tmp` no debe contener los archivos temporales no usados de los últimos cinco días.

El archivo `/etc/tmpfiles.d/tmp.conf` debe aparecer de la siguiente manera:

```
q /tmp 1777 root root 5d
```

2.3. Compruebe la configuración del archivo `/etc/tmpfiles.d/tmp.conf`.

Debido a que el comando no devuelve ningún error, confirma que los ajustes de configuración son correctos.

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf
```

- ▶ 3. Agregue una nueva configuración que asegure que el directorio `/run/momentary` exista con la propiedad del usuario y del grupo establecida en `root`. Los permisos octales para el directorio deben ser `0700`. La configuración debe purgar los archivos de este directorio que no se hayan usado en los últimos 30 segundos.

3.1. Cree el archivo `/etc/tmpfiles.d/momentary.conf` con el siguiente contenido.

Con la configuración, el servicio `systemd-tmpfiles` asegura que el directorio `/run/momentary` exista con sus permisos octales establecidos en `0700`. El propietario del directorio `/run/momentary` debe ser el usuario `root` y el grupo. El servicio purga cualquier archivo en este directorio si permanece sin usar durante 30 segundos.

```
[root@servera ~]# vim /etc/tmpfiles.d/momentary.conf
d /run/momentary 0700 root root 30s
```

3.2. Compruebe la configuración del archivo `/etc/tmpfiles.d/momentary.conf`. El comando crea el directorio `/run/momentary` si aún no existe.

Debido a que el comando no devuelve ningún error, confirma que los ajustes de configuración son correctos.

```
[root@servera ~]# systemd-tmpfiles --create \
/etc/tmpfiles.d/momentary.conf
```

3.3. Verifique que el comando `systemd-tmpfiles` cree el directorio `/run/momentary` con los permisos, el propietario y el propietario del grupo adecuados.

El permiso octal del directorio `/run/momentary` se establece en `0700` y la propiedad del usuario y del grupo se establece en `root`.

```
[root@servera ~]# ls -ld /run/momentary
drwx-----. 2 root root 40 Apr  4 06:35 /run/momentary
```

- ▶ 4. Verifique que el comando `systemd-tmpfiles --clean` elimine cualquier archivo en el directorio `/run/momentary`, no usado en los últimos 30 segundos en función de la configuración de `systemd-tmpfiles` para el directorio.

- 4.1. Cree el archivo `/run/momentary/test`.

```
[root@servera ~]# touch /run/momentary/test
```

- 4.2. Configure su prompt de shell para que no regrese durante 30 segundos.

```
[root@servera ~]# sleep 30
```

- 4.3. Después de que regrese el prompt de shell, limpie los archivos obsoletos del directorio `/run/momentary`, según la regla a la que se hace referencia en el archivo de configuración `/etc/tmpfiles.d/momentary.conf`.

El comando elimina el archivo `/run/momentary/test`, ya que permanece sin usar durante 30 segundos. Este comportamiento se basa en la regla a la que se hace referencia en el archivo de configuración `/etc/tmpfiles.d/momentary.conf`.

```
[root@servera ~]# systemd-tmpfiles --clean \
/etc/tmpfiles.d/momentary.conf
```

- 4.4. Verifique que el archivo `/run/momentary/test` no existe.

```
[root@servera ~]# ls -l /run/momentary/test
ls: cannot access '/run/momentary/test': No such file or directory
```

- 4.5. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish scheduling-tempfiles
```

Esto concluye la sección.

► Cuestionario

Programación de tareas futuras

Elija las respuestas correctas para las siguientes preguntas.

- 1. ¿Qué comando muestra todos los trabajos de usuario que programó para ejecutarse como trabajos diferidos?
 - a. atq
 - b. atrm
 - c. at -c
 - d. at --display

- 2. ¿Qué comando elimina el trabajo de usuario diferido con el número de trabajo 5?
 - a. at -c 5
 - b. atrm 5
 - c. at 5
 - d. at --delete 5

- 3. ¿Qué comando muestra todos los trabajos de usuario recurrentes programados para el usuario que inició sesión actualmente?
 - a. crontab -r
 - b. crontab -l
 - c. crontab -u
 - d. crontab -V

- 4. ¿Qué formato de trabajo ejecuta /usr/local/bin/daily_backup cada hora desde las 9:00 a. m. hasta las 6:00 p. m. todos los días de lunes a viernes?
 - a. 00 * * * Mon-Fri /usr/local/bin/daily_backup
 - b. * */9 * * Mon-Fri /usr/local/bin/daily_backup
 - c. 00 */18 * * * /usr/local/bin/daily_backup
 - d. 00 09-18 * * Mon-Fri /usr/local/bin/daily_backup

- 5. ¿Qué directorio contiene los scripts de shell destinados a ejecutarse diariamente?
 - a. /etc/cron.d
 - b. /etc/cron.hourly
 - c. /etc/cron.daily
 - d. /etc/cron.weekly

- ▶ 6. ¿Qué archivo de configuración define la configuración de los trabajos del sistema que se ejecutan de forma diaria, semanal y mensual?
 - a. `/etc/crontab`
 - b. `/etc/anacrontab`
 - c. `/etc/inittab`
 - d. `/etc/sysconfig/crond`

- ▶ 7. ¿Qué unidad `systemd` desencadena regularmente la limpieza de los archivos temporales?
 - a. `systemd-tmpfiles-clean.timer`
 - b. `systemd-tmpfiles-clean.service`
 - c. `dnf-makecache.timer`
 - d. `unbound-anchor.timer`

► Solución

Programación de tareas futuras

Elija las respuestas correctas para las siguientes preguntas.

- 1. ¿Qué comando muestra todos los trabajos de usuario que programó para ejecutarse como trabajos diferidos?
 - a. atq
 - b. atrm
 - c. at -c
 - d. at --display

- 2. ¿Qué comando elimina el trabajo de usuario diferido con el número de trabajo 5?
 - a. at -c 5
 - b. atrm 5
 - c. at 5
 - d. at --delete 5

- 3. ¿Qué comando muestra todos los trabajos de usuario recurrentes programados para el usuario que inició sesión actualmente?
 - a. crontab -r
 - b. crontab -l
 - c. crontab -u
 - d. crontab -V

- 4. ¿Qué formato de trabajo ejecuta /usr/local/bin/daily_backup cada hora desde las 9:00 a. m. hasta las 6:00 p. m. todos los días de lunes a viernes?
 - a. 00 * * * Mon-Fri /usr/local/bin/daily_backup
 - b. * */9 * * Mon-Fri /usr/local/bin/daily_backup
 - c. 00 */18 * * * /usr/local/bin/daily_backup
 - d. 00 09-18 * * Mon-Fri /usr/local/bin/daily_backup

- 5. ¿Qué directorio contiene los scripts de shell destinados a ejecutarse diariamente?
 - a. /etc/cron.d
 - b. /etc/cron.hourly
 - c. /etc/cron.daily
 - d. /etc/cron.weekly

- ▶ 6. ¿Qué archivo de configuración define la configuración de los trabajos del sistema que se ejecutan de forma diaria, semanal y mensual?
 - a. `/etc/crontab`
 - b. `/etc/anacrontab`
 - c. `/etc/inittab`
 - d. `/etc/sysconfig/crond`

- ▶ 7. ¿Qué unidad `systemd` desencadena regularmente la limpieza de los archivos temporales?
 - a. `systemd-tmpfiles-clean.timer`
 - b. `systemd-tmpfiles-clean.service`
 - c. `dnf-makecache.timer`
 - d. `unbound-anchor.timer`

Resumen

- Los trabajos o las tareas están programados para ejecutarse una vez en el futuro.
- Los trabajos de usuario recurrentes ejecutan las tareas del usuario en una programación de repetición.
- Los trabajos recurrentes del sistema realizan tareas administrativas en una programación de repetición que tiene impacto en todo el sistema.
- Las unidades del temporizador `systemd` pueden ejecutar tanto los trabajos diferidos como los recurrentes.

capítulo 3

Ajuste del rendimiento del sistema

Meta

Mejorar el rendimiento del sistema estableciendo parámetros de ajuste y ajustando la prioridad de programación de los procesos.

Objetivos

- Optimizar el rendimiento del sistema seleccionando un perfil de ajuste administrado por el daemon `tuned`.
- Dar o quitar la prioridad a procesos específicos con los comandos `nice` y `renice`.

Secciones

- Ajuste de perfiles de optimización (y ejercicio guiado)
- Influencia en la programación de procesos (y ejercicio guiado)

Trabajo de laboratorio

Ajuste del rendimiento del sistema

Ajuste de perfiles de optimización

Objetivos

Optimizar el rendimiento del sistema seleccionando un perfil de ajuste administrado por el daemon `tuned`.

Sistemas de ajuste

Los administradores del sistema optimizan el rendimiento de un sistema ajustando diversas configuraciones de dispositivos basadas en una variedad de cargas de trabajo de casos de uso. El daemon `tuned` aplica la configuración de ajuste de forma estática y dinámica, con perfiles de optimización que reflejan los requisitos particulares de la carga de trabajo.

Configuración de ajuste estático

El daemon `tuned` aplica la configuración del sistema cuando se inicia el servicio o al seleccionar un nuevo perfil de ajuste. El ajuste estático configura los parámetros predefinidos del `kernel` en los perfiles a los que se aplica el daemon `tuned` en el tiempo de ejecución. Con el ajuste estático, el daemon `tuned` define los parámetros del `kernel` para las expectativas de rendimiento general sin ajustar estos parámetros a medida que cambian los niveles de actividad.

Configuración de ajuste dinámico

Con el ajuste dinámico, el daemon `tuned` monitorea la actividad del sistema y ajusta la configuración según los cambios de comportamiento del tiempo de ejecución. El ajuste dinámico se modifica continuamente para adaptarse a la carga de trabajo actual, comenzando con los ajustes declarados iniciales en el perfil ajustado seleccionado.

Por ejemplo, los dispositivos de almacenamiento experimentan un alto uso durante el arranque y el inicio de sesión, pero tienen una actividad mínima cuando las cargas de trabajo de los usuarios consisten en el uso de navegadores web y clientes de correo electrónico. De manera similar, la actividad de la CPU y los dispositivos de red aumentan durante el uso máximo en un día laboral. El daemon `tuned` monitorea la actividad de estos componentes y ajusta la configuración de parámetros para maximizar el rendimiento durante momentos de mucha actividad y reducir los ajustes durante momentos de poca actividad. El daemon `tuned` usa los parámetros de rendimiento proporcionados en los perfiles de ajuste predefinidos.

Para monitorear y ajustar la configuración de los parámetros, el daemon `tuned` usa módulos denominados *complementos (plug-ins)* de monitoreo y ajuste, respectivamente.

Los complementos (plug-ins) de monitoreo analizan el sistema y obtienen información de él, de modo que los complementos (plug-ins) de ajuste usan esta información para el ajuste dinámico. En este momento, el daemon `tuned` se envía con tres complementos (plug-ins) de monitoreo diferentes:

- `disk`: monitorea la carga del disco en función del número de operaciones de E/S para cada dispositivo de disco.
- `net`: monitorea la carga de la red en función del número de paquetes transferidos por tarjeta de red.
- `load`: monitorea la carga de la CPU para cada CPU.

Los complementos (plug-ins) de ajuste ajustan los subsistemas individuales. Usan los datos obtenidos por los complementos (plug-ins) de monitoreo y los parámetros de rendimiento proporcionados por los perfiles de ajuste predefinidos. Entre otros, el daemon `tuned` incluye los siguientes complementos (plug-ins) de ajuste:

- `disk`: define diferentes parámetros de disco, por ejemplo, el programador de disco, el tiempo de espera de inactividad o la administración avanzada de energía.
- `net`: configura la velocidad de la interfaz y la funcionalidad Wake-on-LAN (WoL).
- `cpu`: define diferentes parámetros de CPU, por ejemplo, el regulador de CPU o la latencia.

De forma predeterminada, el ajuste dinámico está deshabilitado. Puede habilitarlo estableciendo la variable `dynamic_tuning` en 1 en el archivo de configuración `/etc/tuned/tuned-main.conf`. Si habilita el ajuste dinámico, el daemon `tuned` monitorea periódicamente el sistema y ajusta la configuración de ajuste a los cambios de comportamiento del tiempo de ejecución. Puede establecer el tiempo en segundos entre actualizaciones mediante el uso de la variable `update_interval` en el archivo de configuración `/etc/tuned/tuned-main.conf`.

```
[root@host ~]$ cat /etc/tuned/tuned-main.conf
...output omitted...
# Dynamically tune devices, if disabled only static tuning will be used.
dynamic_tuning = 1
...output omitted...
# Update interval for dynamic tunings (in seconds).
# It must be multiply of the sleep_interval.
update_interval = 10
...output omitted...
```

La utilidad de ajuste

La instalación mínima de Red Hat Enterprise Linux incluye el paquete `tuned` de manera predeterminada. Puede instalar y habilitar el paquete manualmente con los siguientes comandos:

```
[root@host ~]$ dnf install tuned
...output omitted...
[root@host ~]$ systemctl enable --now tuned
Created symlink /etc/systemd/system/multi-user.target.wants/tuned.service → /usr/lib/systemd/system/tuned.service.
```

La aplicación `tuned` proporciona perfiles divididos en las siguientes categorías:

- Perfiles de ahorro de energía
- Perfiles de aumento de rendimiento

Los perfiles de aumento de rendimiento incluyen perfiles que se centran en los siguientes aspectos:

- Baja latencia de almacenamiento y red
- Alto rendimiento de almacenamiento y red
- Rendimiento de máquinas virtuales
- Rendimiento del host de virtualización

La siguiente tabla muestra una lista de los perfiles de ajuste distribuidos con Red Hat Enterprise Linux 9.

Perfiles de ajuste distribuidos con Red Hat Enterprise Linux 9

Perfil de ajuste	Propósito
balanced	Ideal para los sistemas que requieren un equilibrio entre ahorro de energía y rendimiento.
powersave	Ajusta el sistema para maximizar el ahorro de energía.
throughput-performance	Ajusta el sistema para maximizar el rendimiento.
accelerator-performance	Ajusta lo mismo que throughput-performance y también reduce la latencia a menos de 100 μ s.
latency-performance	Ideal para los sistemas de servidores que requieren baja latencia a expensas del consumo de energía.
network-throughput	Este perfil se deriva del perfil throughput-performance . Se aplican parámetros de ajuste de red adicionales para maximizar el rendimiento de la red.
network-latency	Este perfil se deriva del perfil latency-performance . Habilita parámetros de ajuste de red adicionales para proporcionar una baja latencia de red.
desktop	Este perfil se deriva del perfil balanced . Proporciona una respuesta más rápida de las aplicaciones interactivas.
hpc-compute	Este perfil se deriva del perfil latency-performance . Ideal para computación de alto rendimiento.
virtual-guest	Ajusta el sistema para maximizar el rendimiento si se ejecuta en una máquina virtual.
virtual-host	Ajusta el sistema para maximizar el rendimiento si actúa como host de máquinas virtuales.
intel-sst	Optimizado para sistemas con configuraciones de tecnología Intel Speed Select. Úselo como superposición en otros perfiles.
optimize-serial-console	Aumenta la capacidad de respuesta de la consola serial. Úselo como superposición en otros perfiles.

La aplicación **tuned** almacena los perfiles de ajuste en los directorios `/usr/lib/tuned` y `/etc/tuned`. Cada perfil tiene un directorio separado y, dentro del directorio, el archivo de configuración principal `tuned.conf` y, opcionalmente, otros archivos.

```
[root@host ~]# cd /usr/lib/tuned
[root@host tuned]# ls
accelerator-performance  hpc-compute          network-throughput    throughput-
performance
balanced                  intel-sst             optimize-serial-console virtual-
guest
desktop                   latency-performance  powersave             virtual-
host
functions                  network-latency       recommend.d
[root@host tuned]$ ls virtual-guest
tuned.conf
```

Un archivo de configuración `tuned.conf` típico se ve de la siguiente manera:

```
[root@host tuned]# cat virtual-guest/tuned.conf
#
# tuned configuration
#

[main]
summary=Optimize for running inside a virtual guest
include=throughput-performance

[sysctl]
# If a workload mostly uses anonymous memory and it hits this limit, the entire
# working set is buffered for I/O, and any more write buffering would require
# swapping, so it's time to throttle writes until I/O can catch up. Workloads
# that mostly use file mappings may be able to use even higher values.
#
# The generator of dirty data starts writeback at this percentage (system default
# is 20%)
vm.dirty_ratio = 30

# Filesystem I/O is usually much more efficient than swapping, so try to keep
# swapping low. It's usually safe to go even lower than this on systems with
# server-grade storage.
vm.swappiness = 30
```

La sección `[main]` del archivo puede incluir un resumen del perfil de ajuste. Esta sección también acepta el parámetro `include`, que puede usar para hacer que el perfil herede todas las configuraciones del perfil al que se hace referencia.

Esto es muy útil al crear nuevos perfiles de ajuste, ya que puede usar uno de los perfiles provistos como base y luego agregar o modificar los parámetros que desea configurar. Para crear o modificar perfiles de ajuste, copie los archivos del perfil de ajuste del directorio `/usr/lib/tuned` al directorio `/etc/tuned` y, luego, modifíquelos. En caso de que haya directorios de perfil con el mismo nombre en los directorios `/usr/lib/tuned` y `/etc/tuned`, estos últimos siempre tienen prioridad. Nunca modifique directamente los archivos en el directorio del sistema `/usr/lib/tuned`.

El resto de las secciones del archivo `tuned.conf` usan los complementos (plug-ins) de ajuste para modificar los parámetros en el sistema. En el ejemplo anterior, la sección `[sysctl]` modifica los parámetros del kernel `vm.dirty_ratio` y `vm.swappiness` a través del complemento (plug-in) `sysctl`.

Administración de perfiles desde la línea de comandos

Use el comando `tuned-adm` para modificar la configuración del daemon `tuned`. El comando `tuned-adm` consulta la configuración actual, enumera los perfiles disponibles, recomienda un perfil de ajuste para el sistema, modifica los perfiles directamente o desactiva los ajustes.

Puede identificar el perfil de ajuste activo actualmente con el comando `tuned-adm active`.

```
[root@host ~]# tuned-adm active
Current active profile: virtual-guest
```

El comando `tuned-adm list` enumera todos los perfiles de ajuste disponibles, incluidos ambos perfiles incorporados y los perfiles de ajuste personalizados.

```
[root@host ~]# tuned-adm list
Available profiles:
- accelerator-performance      - Throughput performance based tuning with ...
- balanced                     - General non-specialized tuned profile
- desktop                      - Optimize for the desktop use-case
- hpc-compute                  - Optimize for HPC compute workloads
- intel-sst                    - Configure for Intel Speed Select Base Frequency
- latency-performance          - Optimize for deterministic performance at ...
- network-latency              - Optimize for deterministic performance at ...
...output omitted...
Current active profile: virtual-guest
```

Use el comando `tuned-adm profile profilename` para cambiar a un perfil diferente que se adapte mejor a los requisitos de ajuste actuales del sistema.

```
[root@host ~]$ tuned-adm profile throughput-performance
[root@host ~]$ tuned-adm active
Current active profile: throughput-performance
```

El comando `tuned-adm recommend` puede recomendar un perfil de ajuste para el sistema. El sistema usa este mecanismo para determinar el perfil predeterminado después de la instalación.

```
[root@host ~]$ tuned-adm recommend
virtual-guest
```



nota

El comando `tuned-adm recommend` basa su recomendación en diversas características del sistema, entre ellas, si el sistema es una máquina virtual y otras categorías predefinidas seleccionadas durante la instalación del sistema.

Para revertir los cambios de configuración que aplica el perfil actual, cambie a otro perfil o desactive el daemon ajustado. Desactive la actividad de ajuste de la aplicación `tuned` con el comando `tuned-adm off`.


```
[root@host ~]$ tuned-adm off
[root@host ~]$ tuned-adm active
No current active profile.
```

Administración de perfiles con la consola web

Para administrar los perfiles de rendimiento del sistema con la consola web, debe iniciar sesión y aumentar los privilegios. El modo de aumento de privilegios permite que el usuario ejecute comandos, con privilegios administrativos, que modifican los perfiles de rendimiento del sistema. Debido a que cambiar los perfiles de ajuste modifica algunos parámetros del sistema, debe hacerlo con privilegios administrativos.

Puede cambiar al modo de acceso administrativo en la consola web haciendo clic en los botones **Limited access** o **Turn on administrative access**. A continuación, ingrese su contraseña cuando se le solicite. Después de aumentar los privilegios, el botón **Limited access** cambia a **Administrative access**. Como recordatorio de seguridad, recuerde que siempre debe volver al modo de acceso limitado una vez que realice en su sistema la tarea que requiere privilegios administrativos.

Como usuario privilegiado, haga clic en la opción del menú **Overview** en la barra de navegación izquierda. El perfil **Performance profile** muestra el perfil activo actual.

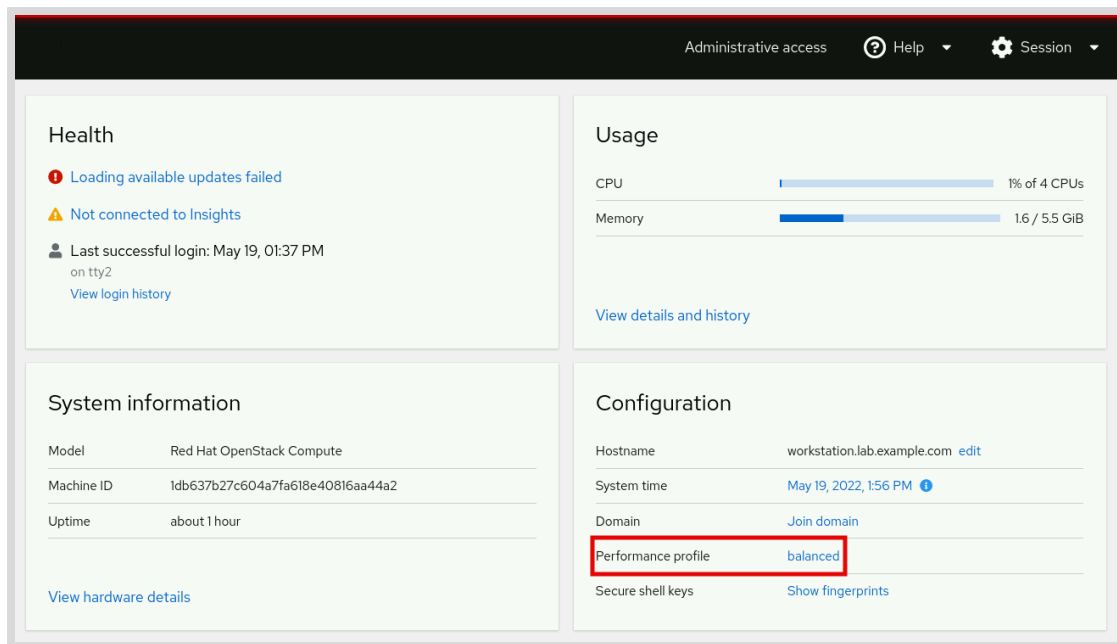


Figura 3.1: Perfil de rendimiento activo

Para seleccionar un perfil diferente, haga clic en el enlace del perfil activo. En la interfaz de usuario **Change performance profile** (Cambiar perfil de rendimiento), desplácese por la lista de perfiles para seleccionar el que mejor se adapte al propósito del sistema y haga clic en el botón **Change profile** (Cambiar perfil).

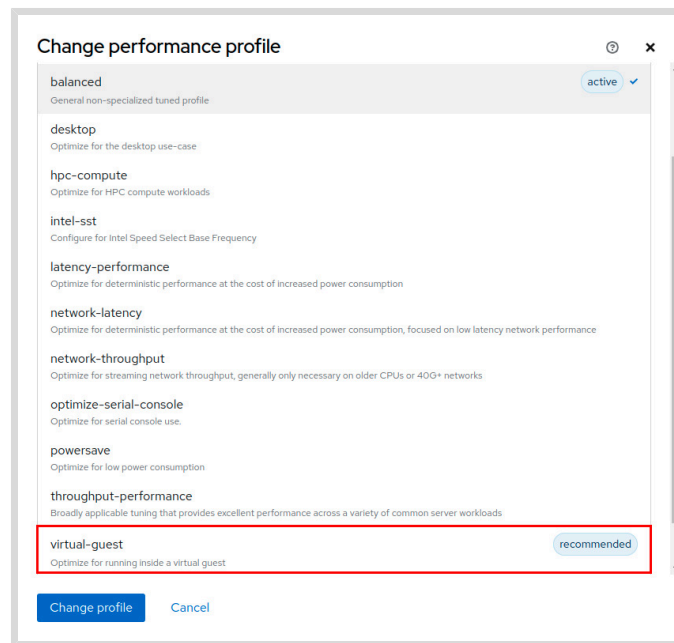


Figura 3.2: Seleccione un perfil de rendimiento preferido

Para verificar los cambios, vuelva a la página principal de **Overview** y confirme que se muestra el perfil activo en el campo **Performance profile**.



Referencias

Páginas del manual: `tuned(8)`, `tuned.conf(5)`, `tuned-main.conf(5)` y `tuned-adm(1)`

Para obtener más información, consulte la guía *Monitoring and Managing System Status and Performance* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/monitoring_and_managing_system_status_and_performance/index

► Ejercicio Guiado

Ajuste de perfiles de optimización

En este ejercicio, ajusta el rendimiento de un servidor activando el servicio `tuned` y aplicando un perfil de ajuste.

Resultados

- Configurar un sistema para usar un perfil de ajuste.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start tuning-profiles
```

Instrucciones

- 1. Inicie sesión en `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Verifique que el paquete `tuned` esté instalado, habilitado e iniciado.

- 2.1. Verifique que el paquete `tuned` esté instalado.

```
[student@servera ~]$ dnf list tuned
...output omitted...
Installed Packages
tuned.noarch                2.18.0-1.el9                @System
```

- 2.2. Verifique que el servicio esté habilitado.

```
[student@servera ~]$ systemctl is-enabled tuned
enabled
```

- 2.3. Verifique que el servicio se esté ejecutando actualmente.

```
[student@servera ~]$ systemctl is-active tuned
active
```

- 3. Enumere los perfiles de ajuste disponibles e identifique el perfil activo.

```
[student@servera ~]$ sudo tuned-adm list
[sudo] password for student: student
Available profiles:
- accelerator-performance      - Throughput performance based tuning with disabled
  higher latency STOP states
- balanced                     - General non-specialized tuned profile
- desktop                      - Optimize for the desktop use-case
- hpc-compute                  - Optimize for HPC compute workloads
- intel-sst                    - Configure for Intel Speed Select Base Frequency
- latency-performance          - Optimize for deterministic performance at the cost
  of increased power consumption
- network-latency              - Optimize for deterministic performance at the cost
  of increased power consumption, focused on low latency network performance
- network-throughput           - Optimize for streaming network throughput,
  generally only necessary on older CPUs or 40G+ networks
- optimize-serial-console      - Optimize for serial console use.
- powersave                   - Optimize for low power consumption
- throughput-performance       - Broadly applicable tuning that provides excellent
  performance across a variety of common server workloads
- virtual-guest                - Optimize for running inside a virtual guest
- virtual-host                 - Optimize for running KVM guests
Current active profile: virtual-guest
```

- 4. Revise el archivo de configuración `tuned.conf` para el perfil activo actual, `virtual-guest`. Puede encontrarlo en el directorio `/usr/lib/tuned/virtual-guest`. El perfil de ajuste `virtual-guest` se basa en el perfil `throughput-performance`, pero define diferentes valores para los parámetros `vm.dirty_ratio` y `vm.swappiness`. Verifique que el perfil de ajuste `virtual-guest` aplique estos valores en su sistema.

- 4.1. Revise el archivo de configuración `virtual-guest` del directorio `/usr/lib/tuned/virtual-guest`. Verifique los valores para los parámetros `vm.dirty_ratio` y `vm.swappiness`.

```
[student@servera ~]$ cat /usr/lib/tuned/virtual-guest/tuned.conf
#
# tuned configuration
#

[main]
summary=Optimize for running inside a virtual guest
include=throughput-performance

[sysctl]
# If a workload mostly uses anonymous memory and it hits this limit, the entire
# working set is buffered for I/O, and any more write buffering would require
# swapping, so it's time to throttle writes until I/O can catch up. Workloads
# that mostly use file mappings may be able to use even higher values.
#
# The generator of dirty data starts writeback at this percentage (system default
# is 20%)
vm.dirty_ratio = 30
```

```
# Filesystem I/O is usually much more efficient than swapping, so try to keep
# swapping low. It's usually safe to go even lower than this on systems with
# server-grade storage.
vm.swappiness = 30
```

4.2. Verifique que el perfil de ajuste aplique estos valores en su sistema.

```
[student@servera ~]$ sysctl vm.dirty_ratio
vm.dirty_ratio = 30
[student@servera ~]$ sysctl vm.swappiness
vm.swappiness = 30
```

- ▶ 5. Revise el archivo de configuración `tuned.conf` para el perfil de ajuste principal `virtual-guest`, `throughput-performance`. Puede encontrarlo en el directorio `/usr/lib/tuned/throughput-performance`. Observe que el perfil de ajuste `throughput-performance` establece un valor diferente para los parámetros `vm.dirty_ratio` y `vm.swappiness`, aunque el perfil `virtual-guest` los sobrescribe. Verifique que el perfil de ajuste `virtual-guest` aplique el valor para el parámetro `vm.dirty_background_ratio`, que hereda del perfil `throughput-performance`.

- 5.1. Revise el archivo de configuración `throughput-performance` del directorio `/usr/lib/tuned/throughput-performance`. Verifique los valores para los parámetros `vm.dirty_ratio`, `vm.swappiness` y `vm.dirty_background_ratio`.

```
[student@servera ~]$ cat /usr/lib/tuned/throughput-performance/tuned.conf
#
# tuned configuration
#

[main]
summary=Broadly applicable tuning that provides excellent performance across a
variety of common server workloads

...output omitted...

[sysctl]
# If a workload mostly uses anonymous memory and it hits this limit, the entire
# working set is buffered for I/O, and any more write buffering would require
# swapping, so it's time to throttle writes until I/O can catch up. Workloads
# that mostly use file mappings may be able to use even higher values.
#
# The generator of dirty data starts writeback at this percentage (system default
# is 20%)
vm.dirty_ratio = 40

# Start background writeback (via writeback threads) at this percentage (system
# default is 10%)
vm.dirty_background_ratio = 10

# PID allocation wrap value. When the kernel's next PID value
# reaches this value, it wraps back to a minimum PID value.
# PIDs of value pid_max or larger are not allocated.
#
# A suggested value for pid_max is 1024 * <# of cpu cores/threads in system>
```

```
# e.g., a box with 32 cpus, the default of 32768 is reasonable, for 64 cpus,
# 65536, for 4096 cpus, 4194304 (which is the upper limit possible).
#kernel.pid_max = 65536

# The swappiness parameter controls the tendency of the kernel to move
# processes out of physical memory and onto the swap disk.
# 0 tells the kernel to avoid swapping processes out of physical memory
# for as long as possible
# 100 tells the kernel to aggressively swap processes out of physical memory
# and move them to swap cache
vm.swappiness=10

...output omitted...
```

- 5.2. Verifique que el perfil de ajuste `virtual-guest` aplique el parámetro heredado `vm.dirty_background_ratio`.

```
[student@servera ~]$ sysctl vm.dirty_background_ratio
vm.dirty_background_ratio = 10
```

- ▶ 6. Cambie el perfil de ajuste activo actual a `throughput-performance` y, a continuación, confirme los resultados. Verifique que los parámetros `vm.dirty_ratio` y `vm.swappiness` cambien a los valores en el archivo de configuración `throughput-performance`.

- 6.1. Cambie el perfil de ajuste activo actual.

```
[student@servera ~]$ sudo tuned-adm profile throughput-performance
```

- 6.2. Confirme que `throughput-performance` es el perfil de ajuste activo.

```
[student@servera ~]$ sudo tuned-adm active
Current active profile: throughput-performance
```

- 6.3. Verifique los valores para los parámetros `vm.dirty_ratio` y `vm.swappiness`.

```
[student@servera ~]$ sysctl vm.dirty_ratio
vm.dirty_ratio = 40
[student@servera ~]$ sysctl vm.swappiness
vm.swappiness = 10
```

- ▶ 7. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish tuning-profiles
```

Esto concluye la sección.

Influencia en la programación de procesos

Objetivos

Dar o quitar la prioridad a procesos específicos con los comandos `nice` y `renice`.

Programación de procesos de Linux

Los sistemas de cómputo modernos usan CPU de varios núcleos y subprocesos que pueden ejecutar muchos subprocesos de instrucciones simultáneamente. Las supercomputadoras de alto rendimiento más grandes pueden tener cientos o miles de CPU con cientos de núcleos de procesamiento y estructuras de subprocesos por CPU y pueden procesar millones de subprocesos de instrucciones en paralelo. Si bien un solo usuario que ejecuta muchas aplicaciones puede saturar el sistema de escritorio típico o la estación de trabajo personal con la actividad de la CPU, una estación de trabajo con el tamaño y la configuración adecuados está diseñada para adaptarse a la carga de trabajo deseada del usuario. Sin embargo, el típico servidor empresarial o de Internet gestiona cientos o miles de usuarios y solicitudes de aplicaciones por segundo, lo que puede provocar fácilmente la saturación de la CPU. Todos los sistemas bajo la carga de la CPU experimentarán escenarios que requieren el manejo de más subprocesos que las unidades de procesamiento de CPU que el sistema tiene para programar inmediatamente los subprocesos.

Linux y otros sistemas operativos usan una técnica llamada *particionamiento de tiempo* o *multitarea* para la administración de procesos. El *programador de procesos* del sistema operativo cambia rápidamente entre subprocesos en cada núcleo de CPU disponible. Este comportamiento da la impresión de que se está ejecutando una cantidad significativa de procesos al mismo tiempo.

Prioridad de los procesos

Cada proceso tiene una medida de importancia variable, conocida históricamente como una *prioridad* de proceso. Linux implementa *políticas de programación* que definen las reglas mediante las cuales se organizan y priorizan los procesos para obtener el tiempo de procesamiento de la CPU. Linux tiene diferentes *políticas de programación* diseñadas para manejar solicitudes de aplicaciones interactivas, procesamiento de aplicaciones por lotes no interactivo y requisitos de aplicaciones en tiempo real. Las políticas de programación en tiempo real aún usan prioridades de procesos y colas, pero las políticas de programación actuales que no son de tiempo real (*normales*) usan el Programador Completamente equitativo (CFS), que en su lugar organiza los procesos que esperan tiempo de CPU en un árbol de búsqueda binario. Esta introducción a la prioridad del proceso describe la política de programación predeterminada denominada `SCHED_NORMAL` o `SCHED_OTHER`.

A los procesos que se ejecutan bajo la política `SCHED_NORMAL` se les asigna una prioridad *estática* en tiempo real de 0, para garantizar que todos los procesos en tiempo real del sistema tengan una prioridad más alta que los procesos normales. Sin embargo, este valor de prioridad estática no se incluye al organizar subprocesos de procesos normales para la programación de CPU. En cambio, el algoritmo de programación CFS organiza los subprocesos de proceso normales en un árbol binario ponderado en el tiempo, con el primer ítem que tiene la menor cantidad de tiempo de CPU usado anteriormente hasta el último ítem que tiene el tiempo de CPU más acumulado.

Valor nice

El orden del árbol binario también está influenciado por un valor *nice* por proceso modificable por el usuario, que varía de -20 (mayor prioridad) a 19 (menor prioridad), con un valor predeterminado de 0. Los procesos heredan su valor *nice* inicial de su proceso principal.

Un valor *nice* más alto indica una disminución en la prioridad del proceso con respecto al valor predeterminado, lo que puede recordarse como *hacer que el proceso sea más nice* que otros procesos. Un valor *nice* más bajo indica una disminución en la prioridad del proceso con respecto al valor predeterminado, lo que puede recordarse como *hacer que el proceso sea menos nice* que otros procesos.

La modificación del valor *nice* en un proceso aumentará o disminuirá la posición del subproceso en el árbol binario. Aumentar el valor *nice* disminuirá la posición del subproceso y disminuir el valor aumentará la posición del subproceso.



Importante

En general, las prioridades solo determinan indirectamente la cantidad de tiempo de CPU que recibe un subproceso de proceso. En un sistema no saturado con capacidad de CPU disponible, cada proceso se programa para el tiempo de CPU inmediato, tanto como lo desee cada proceso. La importancia relativa del proceso, como se administra en el árbol binario, determina solo qué subprocesos se seleccionan y se colocan en las CPU primero.

En un sistema saturado de CPU, donde hay más subprocesos en espera que unidades de procesamiento de CPU, los subprocesos de mayor prioridad (menor *nice*) se colocan primero, hasta que todas las unidades de CPU estén ocupadas, mientras que los subprocesos de menor prioridad (mayor *nice*) deben esperar inicialmente en el árbol binario. Sin embargo, el Programador completamente equitativo está diseñado para balancear la importancia del proceso, los valores *nice* y el tiempo de CPU acumulado anterior, y ajusta dinámicamente el árbol binario de modo que todos los procesos obtengan un tiempo de CPU equitativo.

Permiso para modificar valores nice

Los usuarios con privilegios pueden disminuir el valor *nice* de un proceso para hacer que un proceso sea menos *nice*, lo que hará que un proceso se coloque repetidamente más arriba en el árbol binario y, por lo tanto, se programe con mayor frecuencia. En un sistema saturado, se reduce el tiempo total de CPU disponible para otros procesos.

Los usuarios sin privilegios solo pueden aumentar el valor *nice* en sus propios procesos, lo que hará que estos sean más *nice* y, por lo tanto, se coloquen más abajo en el árbol binario. Los usuarios sin privilegios no pueden disminuir los valores *nice* de sus procesos para aumentar su importancia, ni pueden ajustar los valores *nice* para el proceso de otro usuario.

Visualización de valores nice

Los valores *nice* se asignan a un valor de prioridad y ambos valores están disponibles para su visualización en los comandos de listado de procesos. Un valor *nice* de -20 se asigna a una prioridad 0 en el comando `top`. Un valor *nice* de 19 se asigna a una prioridad 39 en el comando `top`.

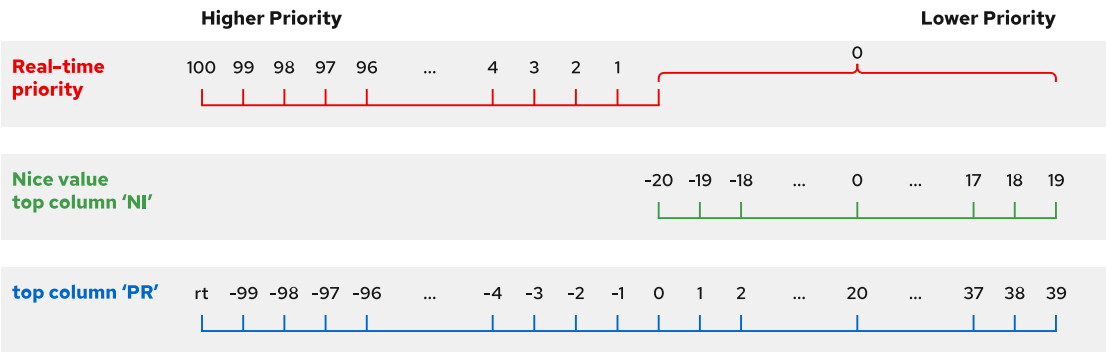


Figura 3.3: Prioridades y valores nice según lo informado por el comando top

En *Figura 3.3*, los valores nice se alinean con los valores de prioridad que usa el comando `top`. El comando `top` muestra la prioridad del proceso en la columna `PR` y el valor nice en la columna `NI`. El esquema de numeración de prioridad `top`, que muestra las prioridades del proceso en tiempo real como números negativos, se hereda de los algoritmos de prioridad internos.

La siguiente salida es el resumen y una lista parcial de procesos en el comando `top`:

Tasks: 192 total, 1 running, 191 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 1.6 sy, 0.0 ni, 96.9 id, 0.0 wa, 0.0 hi, 1.6 si, 0.0 st
MiB Mem : 5668.6 total, 4655.6 free, 470.1 used, 542.9 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 4942.6 avail Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	172180	16232	10328	S	0.0	0.3	0:01.49	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.01	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp

El comando `ps` muestra los valores nice del proceso cuando se usan las opciones de formato predeterminadas.

El siguiente comando `ps` muestra una lista de todos los procesos, con su ID del proceso, nombre del proceso, valor nice y clase de programación, ordenados de forma descendente por valor nice. En la columna de clase de programación `CLS`, `TS` significa *tiempo de uso compartido*, que es otro nombre para las políticas de programación normales, incluida `SCHED_NORMAL`. Otros valores `CLS`, como `FF` para *primero en entrar, primero en salir* y `RR` para *round robin*, indican procesos en tiempo real. A los procesos en tiempo real no se les asignan valores nice, como lo indica el guión (-) en la columna `NI`. Las políticas de programación avanzada se enseñan en el curso *Red Hat Performance Tuning: Linux in Physical, Virtual, and Cloud (RH442)*.

```
[user@host ~]$ ps axo pid,comm,nice,cls --sort=-nice
```

PID	COMMAND	NI	CLS
33	khugepaged	19	TS
32	ksmd	5	TS
814	rtkit-daemon	1	TS
1	systemd	0	TS
2	kthreadd	0	TS
5	kworker/0:0-cgr	0	TS
7	kworker/0:1-rcu	0	TS

```
8 kworker/u4:0-ev    0    TS
15 migration/0      -    FF
...output omitted...
```

Iniciar procesos con valores nice definidos por el usuario

Cuando se crea un proceso, hereda el valor nice del proceso superior. Cuando un proceso se inicia desde la línea de comandos, hereda el valor nice del proceso de la shell desde donde se inició. Por lo general, los procesos nuevos se ejecutan con el valor nice predeterminado de 0.

El siguiente ejemplo inicia un proceso desde la shell y muestra el valor nice del proceso. Tenga en cuenta el uso de la opción PID en el comando `ps` para especificar la salida solicitada.



nota

El comando de ejemplo se usa aquí únicamente para demostrar valores nice y se eligió por su bajo consumo de recursos.

```
[user@host ~]$ sleep 60 &
[1] 2667
[user@host ~]$ ps -o pid,comm,nice 2667
  PID COMMAND      NI
 2667 sleep          0
```

Todos los usuarios pueden usar el comando `nice` para iniciar comandos con un valor nice predeterminado o superior. Sin opciones, el comando `nice` inicia un proceso con el valor nice predeterminado de 10. Definir un valor más alto de manera predeterminada garantiza que el nuevo proceso tenga una prioridad más baja que su shell de trabajo actual y es menos probable que afecte su sesión interactiva actual.

En el siguiente ejemplo, se inicia el mismo comando que una tarea en segundo plano con el valor nice predeterminado y se muestra el valor nice del proceso:

```
[user@host ~]$ nice sleep 60 &
[1] 2736
[user@host ~]$ ps -o pid,comm,nice 2736
  PID COMMAND      NI
 2736 sleep         10
```

Use la opción `-n` del comando `nice` para aplicar un valor nice definido por el usuario para el proceso de arranque. El proceso predeterminado es agregar 10 al valor nice actual del proceso. En el siguiente ejemplo, se inicia una tarea en segundo plano con el valor nice definido por el usuario de 15 y muestra el resultado:

```
[user@host ~]$ nice -n 15 sleep 60 &
[1] 2673
[user@host ~]$ ps -o pid,comm,nice 2740
  PID COMMAND      NI
 2740 sleep         15
```

**Importante**

Los usuarios sin privilegios solo pueden aumentar el valor nice de su valor actual a un máximo de 19. Una vez que aumenta el valor, los usuarios sin privilegios no pueden reducir el valor para revertir al valor nice anterior. Sin embargo, un usuario privilegiado puede reducir el valor nice de cualquier valor actual a un mínimo de -20.

Cambio del valor nice de un proceso existente

Puede cambiar el valor nice de un proceso existente con el comando `renice`. En este ejemplo, se usa el ID del proceso del ejemplo anterior para cambiar el valor nice actual de 15 a un nuevo valor nice de 19.

```
[user@host ~]$ renice -n 19 2740  
2740 (process ID) old priority 15, new priority 19
```

También puede usar el comando `top` para cambiar el valor nice en un proceso existente. Desde la interfaz interactiva `top`, presione la opción `r` para acceder al comando `renice`. Ingrese el ID del proceso y, luego, el nuevo valor nice.

**Referencias**

Páginas del manual: `nice(1)`, `renice(1)`, `top(1)` y `sched_setscheduler(2)`

► Ejercicio Guiado

Influencia en la programación de procesos

En este ejercicio, ajustará la prioridad de programación de los procesos con los comandos `nice` y `renice`, y observará los efectos en la ejecución de procesos.

Resultados

- Ajuste las prioridades de programación para los procesos.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start tuning-procscheduling
```



Importante

Este ejercicio usa comandos que realizan una suma de comprobación infinita en un archivo de dispositivo mientras usan intencionalmente recursos de CPU significativos. Verifique que haya finalizado todos los procesos del ejercicio antes de dejar este ejercicio.

Instrucciones

- 1. Use el comando `ssh` para iniciar sesión en la máquina `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Determine la cantidad de núcleos de CPU en la máquina `servera` y, a continuación, inicie dos instancias del comando `sha1sum /dev/zero &` para cada núcleo.

- 2.1. Use el comando `grep` para analizar la cantidad de procesadores virtuales existentes (núcleos de CPU) del archivo `/proc/cpuinfo`.

```
[student@servera ~]$ grep -c '^processor' /proc/cpuinfo
2
```

- 2.2. Use un comando de bucle para iniciar varias instancias del comando `sha1sum /dev/zero &`. Inicie dos instancias para cada procesador virtual que se indicó en el

paso anterior. En este ejemplo, un bucle `for` crea cuatro instancias. Los valores de PID en su salida podrían variar con respecto al ejemplo.

```
[student@servera ~]$ for i in {1..4}; do sha1sum /dev/zero & done
[1] 1132
[2] 1133
[3] 1134
[4] 1135
```

- ▶ 3. Verifique que las tareas en segundo plano se estén ejecutando para cada uno de los procesos `sha1sum`.

```
[student@servera ~]$ jobs
[1]  Running          sha1sum /dev/zero &
[2]  Running          sha1sum /dev/zero &
[3]- Running          sha1sum /dev/zero &
[4]+ Running          sha1sum /dev/zero &
```

- ▶ 4. Use los comandos `ps` y `pgrep` para mostrar el porcentaje de uso de CPU para cada proceso `sha1sum`.

```
[student@servera ~]$ ps u $(pgrep sha1sum)
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
student	1132	49.6	0.1	225336	2288	pts/0	R	11:40	2:40	sha1sum /dev/zero
student	1133	49.6	0.1	225336	2296	pts/0	R	11:40	2:40	sha1sum /dev/zero
student	1134	49.6	0.1	225336	2264	pts/0	R	11:40	2:40	sha1sum /dev/zero
student	1135	49.6	0.1	225336	2280	pts/0	R	11:40	2:40	sha1sum /dev/zero

- ▶ 5. Cierre todos los procesos `sha1sum` y, a continuación, verifique que no haya trabajos en ejecución.

- 5.1. Use el comando `kill` para cerrar todos los procesos en ejecución con el patrón de nombre `sha1sum`.

```
[student@servera ~]$ kill sha1sum
[2]  Terminated          sha1sum /dev/zero
[4]+ Terminated          sha1sum /dev/zero
[1]- Terminated          sha1sum /dev/zero
[3]+ Terminated          sha1sum /dev/zero
```

- 5.2. Verifique que no haya trabajos en ejecución.

```
[student@servera ~]$ jobs
[student@servera ~]$
```

- ▶ 6. Inicie varias instancias del comando `sha1sum /dev/zero &` y, a continuación, inicie una instancia adicional del comando `sha1sum /dev/zero &` con un nivel `nice` de 10. Inicie tantas instancias como procesadores virtuales tenga el sistema. En este ejemplo, se inician tres instancias normales y otra con el nivel `nice` más alto.

- 6.1. Use los bucles para iniciar tres instancias del comando `sha1sum /dev/zero &`.

```
[student@servera ~]$ for i in {1..3}; do sha1sum /dev/zero & done
[1] 1207
[2] 1208
[3] 1209
```

6.2. Use el comando `nice` para iniciar la cuarta instancia con un nivel `nice` de 10.

```
[student@servera ~]$ nice -n 10 sha1sum /dev/zero &
[4] 1210
```

- 7. Use los comandos `ps` y `pgrep` para mostrar el PID, el porcentaje de uso de CPU, el valor `nice` y el nombre ejecutable para cada proceso. La instancia con el valor `nice` de 10 debe mostrar un porcentaje menor de uso de CPU que las otras instancias.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
PID %CPU  NI COMMAND
1207 64.2   0 sha1sum
1208 65.0   0 sha1sum
1209 63.9   0 sha1sum
1210  8.2   10 sha1sum
```

- 8. Use el comando `sudo renice` para bajar el nivel `nice` de un proceso del paso anterior. Use el valor PID de la instancia del proceso con el nivel `nice` de 10 para reducir su nivel `nice` a 5.

```
[student@servera ~]$ sudo renice -n 5 1210
[sudo] password for student: student
1210 (process ID) old priority 10, new priority 5
```

- 9. Repita los comandos `ps` y `pgrep` para mostrar el porcentaje de CPU y el nivel `nice`.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
PID %CPU  NI COMMAND
1207 62.9   0 sha1sum
1208 63.2   0 sha1sum
1209 63.2   0 sha1sum
1210 10.9   5 sha1sum
```

- 10. Use el comando `pkill` para cerrar todos los procesos en ejecución con el patrón de nombre `sha1sum`.

```
[student@servera ~]$ pkill sha1sum
...output omitted...
```

- 11. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish tuning-procscheduling
```

Esto concluye la sección.

► Trabajo de laboratorio

Ajuste del rendimiento del sistema

En este trabajo de laboratorio, aplica un perfil de ajuste específico y ajustará la prioridad de programación de un proceso existente con un alto uso de CPU.

Resultados

- Activar un perfil de ajuste específico para un sistema de cómputo.
- Ajustar la prioridad de programación de CPU de un proceso.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start tuning-review
```



Importante

Este trabajo de laboratorio usa comandos que realizan una suma de comprobación infinita en un archivo de dispositivo mientras usan intencionalmente recursos de CPU significativos. Verifique que haya finalizado todos los procesos del trabajo de laboratorio antes de dejar este trabajo de laboratorio.

Instrucciones

1. Cambie el perfil de ajuste actual para la máquina `serverb` al perfil `balanced`, un perfil de ajuste general no especializado. Enumere la información para el perfil de ajuste `balanced` cuando es el perfil de ajuste actual.
2. Dos procesos en `serverb` están consumiendo un alto porcentaje de uso de CPU. Ajuste el nivel de `nice` de cada proceso en 10 para permitir más tiempo de CPU para otros procesos.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade tuning-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish tuning-review
```

Esto concluye la sección.

► Solución

Ajuste del rendimiento del sistema

En este trabajo de laboratorio, aplica un perfil de ajuste específico y ajustará la prioridad de programación de un proceso existente con un alto uso de CPU.

Resultados

- Activar un perfil de ajuste específico para un sistema de cómputo.
- Ajustar la prioridad de programación de CPU de un proceso.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start tuning-review
```



Importante

Este trabajo de laboratorio usa comandos que realizan una suma de comprobación infinita en un archivo de dispositivo mientras usan intencionalmente recursos de CPU significativos. Verifique que haya finalizado todos los procesos del trabajo de laboratorio antes de dejar este trabajo de laboratorio.

Instrucciones

1. Cambie el perfil de ajuste actual para la máquina `serverb` al perfil `balanced`, un perfil de ajuste general no especializado. Enumere la información para el perfil de ajuste `balanced` cuando es el perfil de ajuste actual.

- 1.1. Inicie sesión en la máquina `serverb` como el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Verifique que el paquete `tuned` esté instalado.

```
[student@serverb ~]$ dnf list tuned
...output omitted...
Installed Packages
tuned.noarch                2.18.0-1.el9                @System
```

- 1.3. Verifique el estado del servicio **tuned**.

```
[student@serverb ~]$ systemctl is-active tuned
active
```

- 1.4. Enumere todos los perfiles de ajuste disponibles y sus descripciones. Tenga en cuenta que el perfil activo actual es **virtual-guest**.

```
[student@serverb ~]$ sudo tuned-adm list
[sudo] password for student: student
Available profiles:
- accelerator-performance - Throughput performance based tuning with disabled
                           higher latency STOP states
- balanced                 - General non-specialized tuned profile
- desktop                 - Optimize for the desktop use-case
- hpc-compute              - Optimize for HPC compute workloads
- intel-sst                - Configure for Intel Speed Select Base Frequency
- latency-performance     - Optimize for deterministic performance at the cost
                           of increased power consumption
- network-latency         - Optimize for deterministic performance at the cost
                           of increased power consumption, focused on low
                           latency network performance
- network-throughput      - Optimize for streaming network throughput, generally
                           only necessary on older CPUs or 40G+ networks
- optimize-serial-console - Optimize for serial console use.
- powersave               - Optimize for low power consumption
- throughput-performance  - Broadly applicable tuning that provides excellent
                           performance across a variety of common server
                           workloads
- virtual-guest            - Optimize for running inside a virtual guest
- virtual-host             - Optimize for running KVM guests
Current active profile: virtual-guest
```

- 1.5. Cambie el perfil de ajuste activo actual al perfil **balanced**.

```
[student@serverb ~]$ sudo tuned-adm profile balanced
```

- 1.6. Enumere la información de resumen del perfil activo de ajuste actual. Verifique que el perfil **balanced** sea el perfil activo.

```
[student@serverb ~]$ sudo tuned-adm profile_info
Profile name:
balanced

Profile summary:
General non-specialized tuned profile
...output omitted...
```

2. Dos procesos en **serverb** están consumiendo un alto porcentaje de uso de CPU. Ajuste el nivel de **nice** de cada proceso en 10 para permitir más tiempo de CPU para otros procesos.

- 2.1. Determine los dos consumidores de CPU principales en la máquina `serverb`. El comando `ps` enumera los principales consumidores de CPU en la parte inferior de la salida. Los valores de porcentaje de CPU pueden variar en su máquina.

```
[student@serverb ~]$ ps aux --sort=pcpu
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
...output omitted...
root      1079  98.5   0.1 225340   2300 ?        RN    06:25   4:29 sha1sum /dev/zero
root      1095  99.0   0.1 225340   2232 ?        R<    06:25   4:30 md5sum /dev/zero
```

- 2.2. Identifique el nivel actual de `nice` para los dos consumidores de CPU principales.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm \
$(pgrep sha1sum;pgrep md5sum)
  PID %CPU  NI COMMAND
 1079  98.8   2 sha1sum
 1095  99.1  -2 md5sum
```

- 2.3. Ajuste el nivel de `nice` de cada proceso en `10`. Use los valores de PID correctos para sus procesos de la salida del comando anterior.

```
[student@serverb ~]$ sudo renice -n 10 1079 1095
[sudo] password for student: student
1079 (process ID) old priority 2, new priority 10
1095 (process ID) old priority -2, new priority 10
```

- 2.4. Verifique que el nivel actual de `nice` de cada proceso sea `10`.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm \
$(pgrep sha1sum;pgrep md5sum)
  PID %CPU  NI COMMAND
 1079  98.9  10 sha1sum
 1095  99.2  10 md5sum
```

- 2.5. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade tuning-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish tuning-review
```

Esto concluye la sección.

Resumen

- El servicio `tuned` modifica automáticamente la configuración del dispositivo para satisfacer las necesidades específicas del sistema en función del perfil de ajuste predefinido seleccionado.
- Para revertir todos los cambios realizados en la configuración del sistema por el perfil seleccionado, cambie a otro perfil o desactive el servicio `tuned`.
- El sistema asigna una prioridad relativa a un proceso para determinar su acceso a la CPU. Esta prioridad se denomina el valor `nice` de un proceso.
- El comando `nice` asigna una prioridad a un proceso cuando se inicia.
- El comando `renice` modifica la prioridad de un proceso en ejecución.

Administración de seguridad de SELinux

Meta

Proteger y administrar la seguridad del servidor con SELinux.

Objetivos

- Explicar cómo SELinux protege los recursos, cambiar el modo de SELinux actual de un sistema y definir el modo de SELinux predeterminado de un sistema.
- Administrar las reglas de política de SELinux que determinan el contexto predeterminado para archivos y directorios con el comando `semanage fcontext` y aplicar el contexto definido por la política de SELinux a archivos y directorios con el comando `restorecon`.
- Activar y desactivar las reglas de política de SELinux con el comando `setsebool`, administrar el valor persistente de los booleanos de SELinux con el comando `semanage boolean -l` y consultar las páginas `man` que terminan con `_selinux` para encontrar información útil acerca de los booleanos de SELinux.
- Usar las herramientas de análisis de registros de SELinux y visualizar información útil durante la solución de problemas de SELinux con el comando `sealert`.

Secciones

- Cambio del modo de cumplimiento (enforcement) de SELinux (y ejercicio guiado)
- Control de contextos de archivo de SELinux (y ejercicio guiado)
- Ajuste de la política de SELinux con booleanos (y ejercicio guiado)
- Investigación y resolución de problemas de SELinux (y ejercicio guiado)

Trabajo de laboratorio

Administración de seguridad de SELinux

Cambio del modo de cumplimiento (enforcement) de SELinux

Objetivos

Explicar cómo SELinux protege los recursos, cambiar el modo de SELinux actual de un sistema y definir el modo de SELinux predeterminado de un sistema.

Descripción de la arquitectura de SELinux

Security Enhanced Linux (SELinux) es una característica de seguridad fundamental de Linux. El acceso a los archivos, puertos y otros recursos se controla a un nivel granular. Los procesos tienen permitido el acceso solo a los recursos especificados por su política SELinux o parámetros booleanos.

Los permisos de archivos controlan el acceso a los archivos para un usuario o grupo específico. Sin embargo, los permisos de archivos no impiden que un usuario autorizado con acceso a archivos use un archivo para un propósito no deseado.

Por ejemplo, con acceso de escritura a un archivo, otros editores o programas aún pueden abrir y modificar un archivo de datos estructurado que está diseñado solo para escribir a un programa específico, lo que podría provocar daños o un problema de seguridad de datos. Los permisos de archivos no detienen dicho acceso no deseado porque no controlan *cómo* se usa un archivo, sino solo *quién* tiene permitido leer, escribir o ejecutar un archivo.

SELinux consta de conjuntos de políticas que definen los desarrolladores de la aplicación para declarar exactamente qué acciones y accesos son adecuados y permitidos para cada ejecutable binario, archivo de configuración y archivo de datos usados por una aplicación. Esta política se conoce como *política dirigida*, ya que una política define las actividades de una aplicación. Las políticas declaran etiquetas predefinidas que se configuran en programas, archivos y puertos de red individuales.

Uso de SELinux

SELinux impone (enforces) un conjunto de reglas de acceso que definen explícitamente las acciones permitidas entre procesos y recursos. No se permite ninguna acción que no esté definida en una regla de acceso. Debido a que solo se permiten acciones definidas, las aplicaciones con un diseño de seguridad deficiente aún están protegidas del uso malintencionado. Las aplicaciones o los servicios con una política específica se ejecutan en un dominio *confinado*, mientras que una aplicación sin una política se ejecuta de forma *ilimitada* pero sin ninguna protección de SELinux. Las políticas específicas individuales se pueden deshabilitar para ayudar con el desarrollo y la depuración de políticas de seguridad y aplicaciones.

SELinux tiene los siguientes modos operativos:

- **Enforcing:** SELinux hace cumplir las políticas cargadas. Este modo es el predeterminado en Red Hat Enterprise Linux.
- **Permissive:** SELinux carga las políticas y está activo, pero en lugar de aplicar (enforcing) las reglas de control de acceso, registra las infracciones de acceso. Este modo es útil para probar y solucionar problemas de aplicaciones y reglas.
- **Disabled:** SELinux está desactivado. Las infracciones de SELinux no se rechazan ni se registran. Se desaconseja deshabilitar SELinux.



Importante

A partir de Red Hat Enterprise Linux 9, SELinux se puede deshabilitar por completo solo mediante el uso del parámetro de kernel `selinux=0` en el arranque. RHEL ya no soporta la configuración de la opción `SELINUX=disabled` en el archivo `/etc/selinux/config`.

A partir de RHEL 9, si se deshabilita SELinux en el archivo `/etc/selinux/config`, SELinux se inicia y ejecuta la aplicación activa (enforcing), pero sin cargar ninguna política. Debido a que las reglas de política definen acciones permitidas, si no se carga ninguna política, se rechazan todas las acciones. Este comportamiento es intencional y está diseñado para bloquear los intentos maliciosos de eludir la protección de SELinux.

Conceptos básicos de SELinux

El objetivo principal de SELinux es proteger los datos del usuario del uso inadecuado por parte de aplicaciones comprometidas o servicios de sistemas. La mayoría de los administradores de Linux están familiarizados con el modelo estándar de seguridad de permisos de archivos de usuarios, grupos y globales, que se conoce como *control de acceso discrecional (DAC)* porque los administradores definen los permisos de archivo según su necesidad. SELinux proporciona una capa adicional de seguridad basada en objetos, que se define en reglas granulares y se conoce como *control de acceso obligatorio (MAC)* porque las políticas de MAC se aplican a todos los usuarios y no se pueden omitir para usuarios específicos mediante ajustes de configuración discrecionales.

Por ejemplo, el puerto de firewall abierto de un servidor web permite el acceso anónimo remoto a un cliente web. Sin embargo, un usuario malintencionado que accede a ese puerto puede intentar poner en peligro un sistema a través de una vulnerabilidad existente. Si una vulnerabilidad de ejemplo compromete los permisos para el usuario y el grupo `apache`, un usuario malintencionado puede acceder directamente al contenido raíz (root) del documento `/var/www/html`, a los directorios `/tmp` y `/var/tmp` del sistema, u otros archivos y directorios accesibles.

Las políticas de SELinux son reglas de seguridad que definen cómo los procesos específicos acceden a archivos, directorios y puertos relevantes. Cada entidad de recursos, como un archivo, proceso, directorio o puerto, tiene una etiqueta denominada *contexto de SELinux*. La etiqueta de contexto coincide con una regla de política de SELinux definida para permitir que un proceso acceda al recurso etiquetado. De forma predeterminada, la política no permite ningún acceso a menos que una regla explícita otorgue acceso. Cuando no se define ninguna regla de permiso, no se permiten todos los accesos.

Las etiquetas de SELinux tienen campos `user`, `role`, `type` y `security level`. La política específica, que está habilitada en RHEL de manera predeterminada, define reglas mediante el uso del contexto `type`. Por lo general, los nombres de contexto de tipo finalizan en `_t`.

SELinux User	Role	Type	Level	File
unconfined_u	:object_r	:httpd_sys_content_t	s0	/var/www/html/file2

Figura 4.1: Contexto de archivo de SELinux

Conceptos de reglas de acceso a políticas

Por ejemplo, un proceso de servidor web se etiqueta con el contexto de tipo `httpd_t`. Los archivos y directorios del servidor web en el directorio `/var/www/html` y otras ubicaciones

se etiquetan con el contexto de tipo `httpd_sys_content_t`. Los archivos temporales en los directorios `/tmp` y `/var/tmp` tienen los contextos de tipo `tmp_t` como etiqueta. Los puertos del servidor web tienen el contexto de tipo `http_port_t` como etiqueta.

Un proceso de servidor web Apache se ejecuta con el contexto de tipo `httpd_t`. Una regla de política permite que el servidor Apache acceda a archivos y directorios con la etiqueta de contexto de tipo `httpd_sys_content_t`. De manera predeterminada, los archivos en el directorio `/var/www/html` tienen el contexto de tipo `httpd_sys_content_t`. De manera predeterminada, una política de servidor web no tiene reglas `allow` para usar archivos con la etiqueta `tmp_t`, como en los directorios `/tmp` y `/var/tmp`, lo que impide el acceso. Con SELinux habilitado, un usuario malintencionado que usa un proceso Apache comprometido aún no podría acceder al directorio `/tmp`.

Un proceso de servidor web MariaDB se ejecuta con el contexto de tipo `mysqld_t`. De manera predeterminada, los archivos en el directorio `/data/mysql` tienen el contexto de tipo `mysqld_db_t`. Un servidor MariaDB puede acceder a los archivos etiquetados `mysqld_db_t`, pero no tiene reglas para permitir el acceso a los archivos para otros servicios, como los archivos etiquetados `httpd_sys_content_t`.

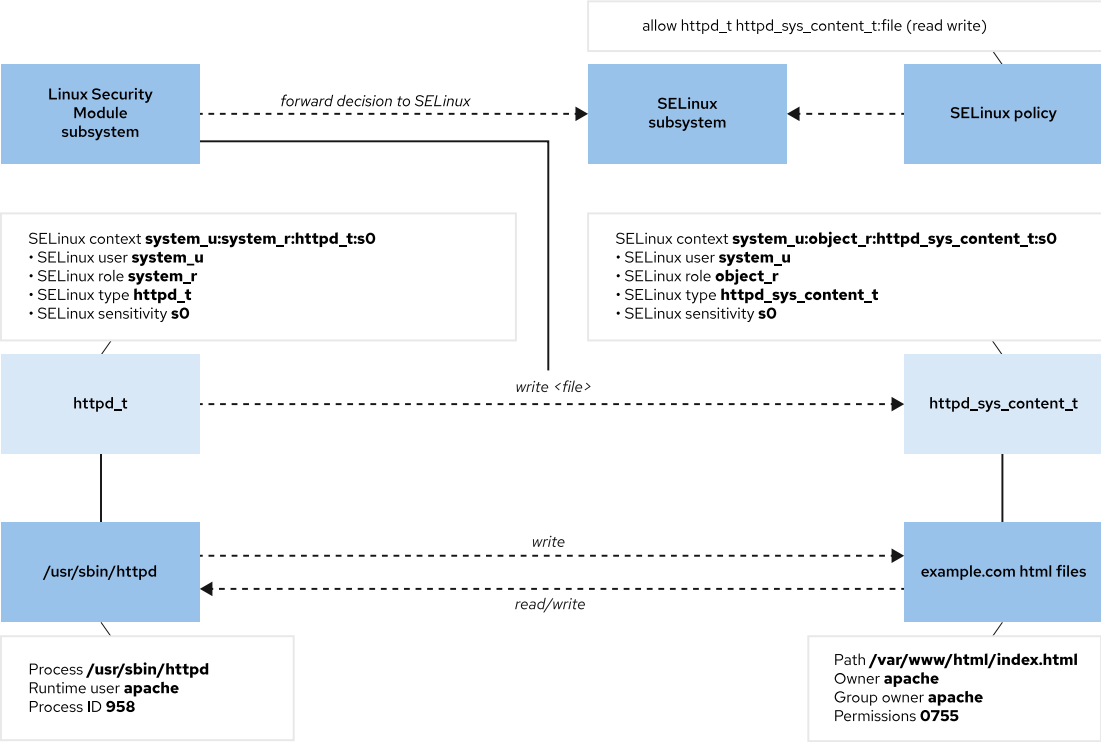


Figura 4.2: Flujo de toma de decisiones de SELinux

Muchos comandos que enumeran recursos usan la opción `-Z` para administrar contextos de SELinux. Por ejemplo, los comandos `ps`, `ls`, `cp` y `mkdir` usan la opción `-Z`.

```
[root@host ~]# ps axZ
LABEL                                PID TTY          STAT TIME COMMAND
system_u:system_r:kernel_t:s0        2 ?           S    0:00 [kthreadd]
system_u:system_r:kernel_t:s0        3 ?           I<   0:00 [rcu_gp]
system_u:system_r:kernel_t:s0        4 ?           I<   0:00 [rcu_par_gp]
...output omitted...
[root@host ~]# systemctl start httpd
[root@host ~]# ps -ZC httpd
```

```

LABEL                                PID TTY          TIME CMD
system_u:system_r:httpd_t:s0        1550 ?              00:00:00 httpd
system_u:system_r:httpd_t:s0        1551 ?              00:00:00 httpd
system_u:system_r:httpd_t:s0        1552 ?              00:00:00 httpd
system_u:system_r:httpd_t:s0        1553 ?              00:00:00 httpd
system_u:system_r:httpd_t:s0        1554 ?              00:00:00 httpd
[root@host ~]# ls -Z /var/www
system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
system_u:object_r:httpd_sys_content_t:s0 html

```

Cambio del modo de SELinux

Use el comando `getenforce` para ver el modo actual de SELinux. Use el comando `setenforce` para cambiar el modo actual de SELinux.

```

[root@host ~]# getenforce
Enforcing
[root@host ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@host ~]# setenforce 0
[root@host ~]# getenforce
Permissive
[root@host ~]# setenforce Enforcing
[root@host ~]# getenforce
Enforcing

```

De manera alternativa, defina el modo SELinux en el momento del arranque con un parámetro del kernel. Pase el parámetro del kernel `enforcing=0` para arrancar el sistema en modo `permissive` o pase `enforcing=1` para arrancar en modo `enforcing`. Deshabilite SELinux pasando el parámetro del kernel `selinux=0` o pase `selinux=1` para habilitar SELinux.

Red Hat recomienda reiniciar el servidor cuando cambie el modo de SELinux de `Permissive` a `Enforcing`. Este reinicio garantiza que los servicios iniciados en modo `permissive` estén limitados en el siguiente arranque.

Configuración del modo predeterminado de SELinux

Para configurar SELinux de manera persistente, use el archivo `/etc/selinux/config`. En el siguiente ejemplo predeterminado, la configuración establece SELinux en el modo `enforcing`. Los comentarios enumeran otros valores válidos, como los modos `permissive` y `disabled`.

```

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
...output omitted...
#
# NOTE: In earlier Fedora kernel builds, SELINUX=disabled would also
# fully disable SELinux during boot. If you need a system with SELinux
# fully disabled instead of SELinux running with no policy loaded, you
# need to pass selinux=0 to the kernel command line. You can use grubby
# to persistently set the bootloader to boot with selinux=0:
#

```

```
# grubby --update-kernel ALL --args selinux=0
#
# To revert back to SELinux enabled:
#
# grubby --update-kernel ALL --remove-args selinux
#
SELINUX=enforcing
# SELINUXTYPE= can take one of these three values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are
#             protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

El sistema lee este archivo en el momento del arranque y configura SELinux en consecuencia. Lo argumentos del kernel `selinux=0|1` y `enforcing=0|1` anulan esta configuración.



Referencias

Páginas del manual: `getenforce(8)`, `setenforce(8)` y `selinux_config(5)`

► Ejercicio Guiado

Cambio del modo de cumplimiento (enforcement) de SELinux

En este trabajo de laboratorio, administra modos de SELinux, tanto de forma temporal como de forma persistente.

Resultados

- Ver y configurar el modo de SELinux actual.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-opsmode
```

Instrucciones

- 1. En la máquina `workstation`, use el comando `ssh` para iniciar sesión en la máquina `servera` como el usuario `student` y, luego, cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Cambie el modo predeterminado de SELinux a `permissive`.
 - 2.1. Use el comando `getenforce` para verificar el modo actual de SELinux en la máquina `servera`.

```
[root@servera ~]# getenforce
Enforcing
```

- 2.2. Use el comando `vim /etc/selinux/config` para editar el archivo de configuración. Cambie el parámetro `SELINUX` del modo `enforcing` a `permissive`.

```
[root@servera ~]# vim /etc/selinux/config
```

- 2.3. Use el comando `grep` para confirmar que el parámetro `SELINUX` se muestra el modo `permissive`.


```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=permissive
SELINUXTYPE=targeted
```

- 2.4. Use el comando `getenforce` para confirmar que el parámetro SELINUX se muestra el modo `enforcing`.

```
[root@servera ~]# getenforce
Enforcing
```

- 2.5. Use el comando `setenforce` para cambiar el modo SELINUX al modo `permissive` y verifique el cambio.

```
[root@servera ~]# setenforce 0
[root@servera ~]# getenforce
Permissive
```

- ▶ 3. Cambie el modo predeterminado de SELinux al modo `enforcing` en el archivo de configuración.
 - 3.1. Use el comando `vim /etc/selinux/config` para editar el archivo de configuración. Cambie el parámetro SELINUX del modo `permissive` a `enforcing`.

```
[root@servera ~]# vim /etc/selinux/config
```

- 3.2. Use el comando `grep` para confirmar que el parámetro SELINUX define el modo `enforcing` en inicio

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=enforcing
SELINUXTYPE=targeted
```

- ▶ 4. Defina el modo SELinux en `enforcing` en la línea de comandos. Reinicie la máquina servera y verifique el modo SELinux.
 - 4.1. Use el comando `setenforce` para definir el modo actual de SELinux en el modo `enforcing`. Use el comando `getenforce` para confirmar que SELinux está en el modo `enforcing`.

```
[root@servera ~]# setenforce 1
[root@servera ~]# getenforce
Enforcing
```

- 4.2. Reinicie la máquina servera para implementar la configuración persistente.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

4.3. Inicie sesión en la máquina `servera` y verifique el modo SELinux.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]# getenforce
Enforcing
```

► 5. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-opsmode
```

Esto concluye la sección.

Control de contextos de archivo de SELinux

Objetivos

Administrar las reglas de política de SELinux que determinan el contexto predeterminado para archivos y directorios con el comando `semanage fcontext` y aplicar el contexto definido por la política de SELinux a archivos y directorios con el comando `restorecon`.

Contexto inicial de SELinux

Todos los recursos, como procesos, archivos y puertos, se etiquetan con un *contexto* de SELinux. SELinux mantiene una base de datos basada en archivos de políticas de etiquetado de archivos en el directorio `/etc/selinux/targeted/contexts/files/`. Los archivos nuevos obtienen una etiqueta predeterminada cuando su nombre de archivo coincide con una política de etiquetado existente.

Cuando el nombre de un archivo nuevo no coincide con una política de etiquetado existente, el archivo hereda la misma etiqueta que el directorio principal. Con la herencia de etiquetado, todos los archivos siempre se etiquetan cuando se crean, independientemente de si existe una política explícita para un archivo.

Cuando se crean archivos en ubicaciones predeterminadas que tienen una política de etiquetado existente, o cuando existe una política para una ubicación personalizada, los archivos nuevos se etiquetan con un contexto SELinux correcto. Sin embargo, si se crea un archivo en una ubicación inesperada sin una política de etiquetado existente, es posible que la etiqueta heredada no sea correcta para el propósito previsto del nuevo archivo.

Además, copiar un archivo en una nueva ubicación puede hacer que el contexto de SELinux de ese archivo cambie, con el nuevo contexto determinado por la política de etiquetado de la nueva ubicación o por la herencia del directorio principal si no existe una política. El contexto de SELinux de un archivo se puede conservar durante una copia para conservar la etiqueta de contexto que se determinó para la ubicación original del archivo. Por ejemplo, el comando `cp -p` conserva todos los atributos del archivo cuando sea posible y el comando `cp -c` conserva solo los contextos de SELinux durante una copia.



nota

Copiar un archivo siempre crea un nuevo inodo de archivo, y los atributos de ese inodo, incluido el contexto de SELinux, deben establecerse inicialmente, como se analizó anteriormente.

Sin embargo, mover un archivo generalmente no crea un nuevo inodo si el movimiento ocurre dentro del mismo sistema de archivos, sino que mueve el nombre de archivo del inodo existente a una nueva ubicación. Debido a que los atributos del inodo existente no necesitan inicializarse, un archivo que se mueve con `mv` conserva su contexto de SELinux a menos que usted establezca un nuevo contexto en el archivo con la opción `-Z`.

Después de copiar o mover un archivo, verifique que tenga el contexto de SELinux adecuado y configúrelo correctamente si es necesario.

El siguiente ejemplo demuestra cómo funciona este proceso.

Cree dos archivos vacíos en el directorio `/tmp`. Ambos archivos reciben el tipo de contexto `user_tmp_t`.

Mueva el primer archivo y copie el segundo archivo al directorio `/var/www/html`.

- El archivo movido conserva el contexto de archivo que se etiquetó desde el directorio `/tmp` original.
- El archivo copiado tiene un nuevo inodo y hereda el contexto de SELinux del directorio de destino `/var/www/html`.

El comando `ls -Z` muestra el contexto de SELinux de un archivo. Observe la etiqueta de los archivos creados en el directorio `/tmp`.

```
[root@host ~]# touch /tmp/file1 /tmp/file2
[root@host ~]# ls -Z /tmp/file*
unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```

El comando `ls -Zd` muestra el contexto de SELinux del directorio especificado. Observe la etiqueta en el directorio `/var/www/html` y los archivos que contiene.

```
[root@host ~]# ls -Zd /var/www/html/
system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
[root@host ~]# ls -Z /var/www/html/index.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/index.html
```

Mueva un archivo del directorio `/tmp` al directorio `/var/www/html`. Copie el otro archivo al mismo directorio. Observe la etiqueta resultante en cada archivo.

```
[root@host ~]# mv /tmp/file1 /var/www/html/
[root@host ~]# cp /tmp/file2 /var/www/html/
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

El archivo movido conservó su etiqueta original y el archivo copiado heredó la etiqueta del directorio de destino. Aunque no es importante para este análisis, `unconfined_u` es el usuario de SELinux, `object_r` es el rol de SELinux y `s0` es el nivel de sensibilidad (más bajo posible). Las configuraciones y características avanzadas de SELinux usan estos valores.

Cambio del contexto de SELinux

Puede cambiar los contextos de SELinux en los archivos con los comandos `semanage fcontext`, `restorecon` y `chcon`.

El método recomendado para definir el contexto para un archivo es crear una política de contexto de archivo con el comando `semanage fcontext` y, luego, aplicar el contexto especificado en la política al archivo con el comando `restorecon`. Este método garantiza que pueda volver a etiquetar fácilmente el archivo en su contexto correcto con el comando `restorecon` siempre que sea necesario. La ventaja de este método es que no necesita recordar cuál se supone que es el contexto y puede corregir fácilmente el contexto en un conjunto de archivos.

El comando `chcon` define el contexto de SELinux directamente en los archivos, pero sin hacer referencia a la política de SELinux del sistema. Si bien `chcon` es útil para probar y depurar, la configuración manual de contextos con este método es temporal. Los contextos de archivos que se establecen manualmente sobreviven a un reinicio, pero pueden reemplazarse si ejecuta `restorecon` para volver a etiquetar el contenido del sistema de archivos.



Importante

Cuando se produce una *nueva etiqueta* del sistema SELinux, todos los archivos en un sistema se etiquetan con sus valores predeterminados de política. Cuando usa `restorecon` en un archivo, cualquier contexto que configure manualmente en el archivo se reemplaza si no coincide con las reglas de la política de SELinux.

El siguiente ejemplo crea un directorio con un contexto de SELinux `default_t`, que heredó del directorio principal `/`.

```
[root@host ~]# mkdir /virtual
[root@host ~]# ls -Zd /virtual
unconfined_u:object_r:default_t:s0 /virtual
```

El comando `chcon` define el contexto de archivo del directorio `/virtual` en el tipo `httpd_sys_content_t`.

```
[root@host ~]# chcon -t httpd_sys_content_t /virtual
[root@host ~]# ls -Zd /virtual
unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
```

La ejecución del comando `restorecon` restablece el contexto al valor predeterminado de `default_t`. Observe el mensaje `Relabeled`.

```
[root@host ~]# restorecon -v /virtual
Relabeled /virtual from unconfined_u:object_r:httpd_sys_content_t:s0 to
unconfined_u:object_r:default_t:s0
[root@host ~]# ls -Zd /virtual
unconfined_u:object_r:default_t:s0 /virtual
```

Definición de las políticas de contextos de archivos predeterminados de SELinux

El comando `semanage fcontext` muestra o modifica las políticas que determina los contextos de archivos predeterminados. Puede enumerar todas las reglas de políticas de contexto de archivos ejecutando el comando `semanage fcontext -l`. Estas reglas usan la sintaxis de expresiones regulares para especificar los nombres de archivo y las rutas de acceso.

Al visualizar políticas, la expresión regular extendida más común es `(/.*)?`, que generalmente se anexa al nombre de un directorio. Esta notación se denomina humorísticamente *pirata*, porque tiene el aspecto de una cara con un parche en el ojo y una mano en forma de gancho al lado.

Esta sintaxis se describe como "un conjunto de caracteres que comienza con una barra y va seguido de cualquier número de caracteres, donde el conjunto puede existir o no". Dicho de manera más simple, esta sintaxis coincide con el directorio en sí, incluso cuando está vacío, pero también coincide con casi cualquier nombre de archivo que se crea dentro de ese directorio.

Por ejemplo, la siguiente regla especifica que el directorio `/var/www/cgi-bin` y cualquier archivo en él o en sus subdirectorios (y sus subdirectorios, etc.) deben tener el contexto de SELinux `system_u:object_r:httpd_sys_script_exec_t:s0`, salvo que una regla más específica lo anule.

```
/var/www/cgi-bin(/.*)? all files system_u:object_r:httpd_sys_script_exec_t:s0
```

Operaciones básicas de contexto de archivo

La siguiente tabla es una referencia para las opciones del comando `semanage fcontext` para agregar, eliminar o enumerar políticas de contextos de archivos de SELinux.

Comandos de `semanage fcontext`

opción	descripción
-a, --add	Agregar un registro del tipo de objeto especificado
-d, --delete	Eliminar un registro del tipo de objeto especificado
-l, --list	Mostrar registros del tipo de objeto especificado

Para administrar contextos de SELinux, instale los paquetes `policycoreutils` y `policycoreutils-python-utils`, que contienen los comandos `restorecon` y `semanage`.

Para restablecer todos los archivos en un directorio al contexto de política predeterminado, primero use el comando `semanage fcontext -l` para localizar y verificar que la política correcta existe con el contexto de archivo deseado. A continuación, use el comando `restorecon` en el nombre del directorio con comodines para restablecer todos los archivos de manera recursiva. En el siguiente ejemplo, vea los contextos de archivos antes y después de usar los comandos `semanage` y `restorecon`.

Primero, verifique el contexto de SELinux para los archivos:

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

A continuación, use el comando `semanage fcontext -l` para enumerar los contextos de archivos SELinux predeterminados:

```
[root@host ~]# semanage fcontext -l
...output omitted...
/var/www(/.*)? all files system_u:object_r:httpd_sys_content_t:s0
...output omitted...
```

La salida del comando `semanage` indica que todos los archivos y subdirectorios en el directorio `/var/www/` tendrán el contexto `httpd_sys_content_t` de manera predeterminada. La ejecución del comando `restorecon` en la carpeta con comodines restaura el contexto predeterminado en todos los archivos y subdirectorios.

```
[root@host ~]# restorecon -Rv /var/www/
Relabeled /var/www/html/file1 from unconfined_u:object_r:user_tmp_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

El siguiente ejemplo usa el comando `semanage` para agregar una política de contexto para un directorio nuevo. Primero, cree el directorio `/virtual` con un archivo `index.html` en él. Vea el contexto de SELinux para el archivo y el directorio.

```
[root@host ~]# mkdir /virtual
[root@host ~]# touch /virtual/index.html
[root@host ~]# ls -Zd /virtual/
unconfined_u:object_r:default_t:s0 /virtual
[root@host ~]# ls -Z /virtual/
unconfined_u:object_r:default_t:s0 index.html
```

A continuación, use el comando `semanage fcontext` para agregar una política de contexto de archivo SELinux para el directorio.

```
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'
```

Use el comando `restorecon` en el directorio con comodines para definir el contexto predeterminado en el directorio y todos los archivos dentro de él.

```
[root@host ~]# restorecon -RFvv /virtual
Relabeled /virtual from unconfined_u:object_r:default_t:s0 to
system_u:object_r:httpd_sys_content_t:s0
Relabeled /virtual/index.html from unconfined_u:object_r:default_t:s0 to
system_u:object_r:httpd_sys_content_t:s0
[root@host ~]# ls -Zd /virtual/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/
[root@host ~]# ls -Z /virtual/
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```

Use el comando `semanage fcontext -l -C` para ver las personalizaciones locales de la política predeterminada.

```
[root@host ~]# semanage fcontext -l -C
SELinux fcontext      type      Context

/virtual(/.*)?       all files system_u:object_r:httpd_sys_content_t:s0
```



Referencias

Páginas del manual: `chcon(1)`, `restorecon(8)`, `semanage(8)` y `semanage-fcontext(8)`

► Ejercicio Guiado

Control de contextos de archivo de SELinux

En este trabajo de laboratorio, realiza un cambio persistente en el contexto de SELinux de un directorio y su contenido.

Resultados

- Configure el servidor HTTP Apache para publicar contenido web desde una raíz (root) de documento no estándar.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-filecontexts
```

Instrucciones

- 1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Configure Apache para usar un directorio de documento en una ubicación no estándar.

- 2.1. Cree el directorio `/custom`.

```
[root@servera ~]# mkdir /custom
```

- 2.2. Cree el archivo `index.html` en el directorio `/custom`. El archivo `index.html` debe contener el texto `This is SERVERA..`

```
[root@servera ~]# echo 'This is SERVERA.' > /custom/index.html
```

- 2.3. Configure Apache para que use la nueva ubicación del directorio. Edite el archivo de configuración de Apache `/etc/httpd/conf/httpd.conf` y reemplace las dos apariciones del directorio `/var/www/html` con el directorio `/custom`. Puede usar

el comando `vim /etc/httpd/conf/httpd.conf` para hacerlo. En el siguiente ejemplo, se muestra el contenido esperado del archivo `/etc/httpd/conf/httpd.conf`.

```
[root@servera ~]# cat /etc/httpd/conf/httpd.conf
...output omitted...
DocumentRoot "/custom"
...output omitted...
<Directory "/custom">
...output omitted...
```

- ▶ 3. Inicie y habilite el servicio web Apache y confirme que el servicio se está ejecutando.

3.1. Inicie y habilite el servicio web Apache con el comando `systemctl`.

```
[root@servera ~]# systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

3.2. Verifique que el servicio se esté ejecutando.

```
[root@servera ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2022-04-06 05:21:19 EDT; 22s ago
     Docs: man:httpd.service(8)
    Main PID: 1676 (httpd)
...output omitted...
Apr 06 05:21:19 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 06 05:21:19 servera.lab.example.com systemd[1]: Started The Apache HTTP Server.
Apr 06 05:21:19 servera.lab.example.com httpd[1676]: Server configured, listening on: port 80
```

- ▶ 4. Abra un navegador web en `workstation` e intente ver la página web `http://servera/index.html`. Recibe un mensaje de error indicando que no tiene permiso para acceder al archivo.
- ▶ 5. Para permitir el acceso al archivo `index.html` en `servera`, debe configurar el contexto de SELinux. Defina una regla de contextos de archivos de SELinux que defina el tipo de contexto en `httpd_sys_content_t` para el directorio `/custom` y todos los archivos en él.

```
[root@servera ~]# semanage fcontext -a \
-t httpd_sys_content_t '/custom(/.*)?'
```

- ▶ 6. Corrija los contextos de archivos en el directorio `/custom`.

```
[root@servera ~]# restorecon -Rv /custom
Relabeled /custom from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
```

- ▶ 7. Intente ver `http://servera/index.html` nuevamente en el navegador web en la máquina `workstation`. Deberá ver el mensaje `This is SERVERA..`
- ▶ 8. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-filecontexts
```

Esto concluye la sección.

Ajuste de la política de SELinux con booleanos

Objetivos

Activar y desactivar las reglas de política de SELinux con el comando `setsebool`, administrar el valor persistente de los booleanos de SELinux con el comando `semanage boolean -l` y consultar las páginas `man` que terminan con `_selinux` para encontrar información útil acerca de los booleanos de SELinux.

Booleanos de SELinux

Un desarrollador de aplicaciones o servicios escribe una política específica de SELinux para definir el comportamiento permitido de la aplicación de destino. Un desarrollador puede incluir un comportamiento de aplicación opcional en la política de SELinux que se puede habilitar cuando el comportamiento está permitido en un sistema específico. Los booleanos de SELinux habilitan o deshabilitan el comportamiento opcional de la política de SELinux. Con los booleanos, puede ajustar selectivamente el comportamiento de una aplicación.

Estos comportamientos opcionales son específicos de la aplicación y deben detectarse y seleccionarse para cada aplicación de destino. Los booleanos específicos del servicio están documentados en la página de manual de SELinux de ese servicio. Por ejemplo, el servicio `httpd` del servidor web tiene su página de manual `httpd(8)` y una página de manual `httpd_selinux(8)` para documentar su política de SELinux, incluidos los tipos de procesos soportados, los contextos de archivos y los comportamientos booleanos disponibles. Las páginas del manual de SELinux se proporcionan en el paquete `selinux-policy-doc`.

Use el comando `getsebool` para enumerar los booleanos disponibles para las políticas objetivo en este sistema y el estado booleano actual. Use el comando `setsebool` para habilitar o deshabilitar el estado de ejecución de estos comportamientos. La opción del comando `setsebool -P` hace que la configuración sea persistente al escribir en el archivo de políticas. Solo los usuarios con privilegios pueden definir booleanos de SELinux.

```
[root@host ~]# getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
...output omitted...
```

Ejemplo de booleano de política `httpd`

La política de servicio `httpd` incluye el booleano `httpd_enable_homedirs` que permite compartir directorios de inicio con `httpd`. Por lo general, el directorio de inicio local de un usuario solo es accesible para el usuario cuando está conectado al sistema local. De manera alternativa, los directorios de inicio se comparten y se accede a ellos mediante un protocolo de uso compartido de archivos remoto, como NFS. En ambas situaciones, los directorios de inicio no se comparten mediante el uso de `https`, de manera predeterminada, y no están disponibles para el usuario a través de un navegador.

```
[root@host ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

Puede habilitar el uso compartido y permitir que los usuarios accedan a sus directorios de inicio con un navegador. Cuando está habilitado, el servicio `httpd` comparte directorios de inicio que están etiquetados con el contexto de archivo `user_home_dir_t`. Los usuarios pueden acceder y administrar sus archivos de directorio de inicio desde un navegador.

Administración del booleano de la política

La configuración de booleanos de SELinux con el comando `setsebool` sin la opción `-P` es temporal, y la configuración volverá a los valores persistentes después del reinicio. Vea información adicional con el comando `semanage boolean -l`, que enumera los booleanos de los archivos de política, incluido si un booleano es persistente, los valores predeterminados y actuales y una breve descripción.

```
[root@host ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs      (off , off) Allow httpd to enable homedirs
[root@host ~]# setsebool httpd_enable_homedirs on
[root@host ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs      (on , off) Allow httpd to enable homedirs
[root@host ~]# getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

Para enumerar solo los booleanos con una configuración actual diferente de la configuración predeterminada en el arranque, use el comando `semanage boolean -l -C`. Este ejemplo tiene el mismo resultado que el ejemplo anterior, sin requerir el filtrado `grep`.

```
[root@host ~]# semanage boolean -l -C
SELinux boolean           State Default Description

httpd_enable_homedirs      (on , off) Allow httpd to enable homedirs
```

En el ejemplo anterior se definió temporalmente el valor actual para el booleano `httpd_enable_homedirs` en `on`, hasta que el sistema se reinicia. Para cambiar la configuración predeterminada, use el comando `setsebool -P` para hacer que la configuración sea persistente. El siguiente ejemplo define un valor persistente y, luego, visualiza la información booleana del archivo de política.

```
[root@host ~]# setsebool -P httpd_enable_homedirs on
[root@host ~]# semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs      (on , on) Allow httpd to enable homedirs
```

Use el comando `semanage boolean -l -C` de nuevo. El booleano se muestra a pesar de que la configuración actual y predeterminada es la misma. Sin embargo, la opción `-C` coincide cuando la configuración actual es diferente de la configuración predeterminada del último arranque. Para este ejemplo de `httpd_enable_homedirs`, la configuración de arranque predeterminada original era `off`.

```
[root@host ~]# semanage boolean -l -C
SELinux boolean          State  Default Description
httpd_enable_homedirs    (on   ,   on) Allow httpd to enable homedirs
```



Referencias

Páginas del manual: `booleans(8)`, `getsebool(8)`, `setsebool(8)`, `semanage(8)` y `semanage-boolean(8)`

► Ejercicio Guiado

Ajuste de la política de SELinux con booleanos

En este ejercicio, configurará Apache para publicar contenido web desde los directorios de inicio de los usuarios.

Resultados

- Configure el servicio web Apache para publicar contenido web desde el directorio de inicio del usuario.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-boolean
```

Instrucciones

- 1. En la máquina `workstation`, use el comando `ssh` para iniciar sesión en la máquina `servera` como el usuario `student` y, luego, cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Edite el archivo de configuración `/etc/httpd/conf.d/userdir.conf` para habilitar la función Apache para que los usuarios puedan publicar contenido web desde sus directorios de inicio. Comente la línea en la sección `IfModule` que define la variable `UserDir` en el valor `disabled` y quite el comentario de la línea que define la variable `UserDir` en el valor `public_html`.

```
[root@servera ~]# vim /etc/httpd/conf.d/userdir.conf
<IfModule mod_userdir.c>
...output omitted...
    # UserDir disabled

...output omitted...
    UserDir public_html
```

```
...output omitted...
</IfModule>
```

- ▶ 3. Inicie y habilite el servicio web Apache.

```
[root@servera ~]# systemctl enable --now httpd
```

- ▶ 4. Abra otra ventana de terminal y use el comando `ssh` para iniciar sesión en la máquina `servera` con el usuario `student`. Cree el archivo de contenido web `index.html` en el directorio `~/public_html`.

- 4.1. En otra ventana de terminal, use el comando `ssh` para iniciar sesión en la máquina `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 4.2. Use el comando `mkdir` para crear el directorio `~/public_html`.

```
[student@servera ~]$ mkdir ~/public_html
```

- 4.3. Cree el archivo `index.html` con el siguiente contenido:

```
[student@servera ~]$ echo 'This is student content on SERVERA.' > \
~/public_html/index.html
```

- 4.4. Para que el servicio web Apache proporcione el contenido del directorio `/home/student/public_html`, debe tener permiso para compartir archivos y subdirectorios en el directorio `/home/student`. Cuando creó el directorio `/home/student/public_html`, se configuró automáticamente con permisos que permiten a cualquier persona con permiso de directorio de inicio acceder a su contenido.

Cambie los permisos del directorio `/home/student` para permitir que el servicio web Apache acceda al subdirectorio `public_html`.

```
[student@servera ~]$ chmod 711 ~
[student@servera ~]$ ls -ld ~
drwx--x--x. 16 student student 4096 Nov  3 09:28 /home/student
```

- ▶ 5. Abra un navegador web en la máquina `workstation` e ingrese la dirección `http://servera/~student/index.html`. Un mensaje de error indica que no tiene permiso para acceder al archivo.
- ▶ 6. Cambie al otro terminal y use el comando `getsebool` para ver si algún booleano restringe el acceso a los directorios para el servicio `httpd`.

```
[root@servera ~]# getsebool -a | grep home
...output omitted...
httpd_enable_homedirs --> off
...output omitted...
```

- ▶ 7. Use el comando `setsebool` para habilitar el acceso persistente al directorio de inicio para el servicio `httpd`.

```
[root@servera ~]# setsebool -P httpd_enable_homedirs on
```

- ▶ 8. Verifique que ahora pueda ver el mensaje `This is student content on SERVERA.` en el navegador web después de ingresar la dirección `http://servera/~student/index.html`.
- ▶ 9. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-booleans
```

Esto concluye la sección.

Investigación y resolución de problemas de SELinux

Objetivos

Usar las herramientas de análisis de registros de SELinux y visualizar información útil durante la solución de problemas de SELinux con el comando `sealert`.

Solución de problemas de SELinux

Cuando las aplicaciones inesperadamente no funcionan debido a denegaciones de acceso a SELinux, existen métodos y herramientas disponibles para resolver estos problemas. Es útil comenzar por comprender algunos conceptos y comportamientos fundamentales cuando SELinux está habilitado.

- SELinux consta de políticas específicas que definen explícitamente las acciones permitidas.
- Una entrada de política define un proceso etiquetado y un recurso etiquetado que interactuará.
- La política establece el tipo de proceso y el contexto de archivo o puerto mediante el uso de etiquetas.
- La entrada de política define un tipo de proceso, una etiqueta de recurso y la acción explícita para permitir.
- Una acción puede ser una llamada al sistema, una función del kernel u otra rutina de programación específica.
- Si no se crea una entrada para una relación específica entre el proceso, el recurso y la acción, se rechaza la acción.
- Cuando se rechaza una acción, el intento se registra con información útil de contexto.

Red Hat Enterprise Linux proporciona una política de SELinux específica y estable para casi todos los servicios de la distribución. Por lo tanto, es inusual tener problemas de acceso de SELinux con los servicios comunes de RHEL cuando están configurados correctamente. Los problemas de acceso a SELinux ocurren cuando los servicios se implementan incorrectamente o cuando las nuevas aplicaciones tienen políticas incompletas. Tenga en cuenta estos conceptos de solución de problemas antes de realizar cambios generales en la configuración de SELinux.

- La mayoría de las denegaciones de acceso indican que SELinux funciona correctamente al bloquear acciones incorrectas.
- La evaluación de las acciones denegadas requiere cierta familiaridad con las acciones de servicio normales y esperadas.
- El problema más frecuente de SELinux es un contexto de archivos incorrecto en archivos nuevos, copiados o movidos.
- Los contextos de archivos se corrigen fácilmente cuando una política existente hace referencia a su ubicación.
- Las características opcionales de la política booleana están documentadas en las páginas del manual `_selinux`.
- La implementación de funciones booleanas generalmente requiere establecer una configuración adicional que no sea de SELinux.
- Las políticas de SELinux no reemplazan ni eluden los permisos de archivos o las restricciones de la lista de control de acceso.

Cuando una aplicación o servicio común falla, y se sabe que el servicio tiene una política de SELinux en funcionamiento, primero revise la página de manual `_selinux` del servicio para

verificar la etiqueta de tipo de contexto correcta. Vea el proceso afectado y los atributos del archivo para verificar que se definan las etiquetas correctas.

Monitoreo de las violaciones de SELinux

El servicio de solución de problemas de SELinux, del paquete `setroubleshoot-server` brinda herramientas para diagnosticar problemas de SELinux. Cuando SELinux rechaza una acción, se registra un mensaje de caché de vector de acceso (AVC) en el archivo de registro de seguridad `/var/log/audit/audit.log`. El servicio de solución de problemas de SELinux monitorea los eventos de AVC y envía un resumen de eventos al archivo `/var/log/messages`.

El resumen de AVC incluye un identificador único de evento (UUID). Use el comando `sealert -l UUID` para ver los detalles completos del informe para el evento específico. Use el comando `sealert -a /var/log/audit/audit.log` para ver todos los eventos existentes.

Considere el siguiente ejemplo de secuencia de comandos en un servidor web Apache estándar: Puede crear `/root/mypage` y moverlo a la carpeta predeterminada de contenido de Apache (`/var/www/html`). A continuación, después de iniciar el servicio Apache, intente recuperar el contenido del archivo.

```
[root@host ~]# touch /root/mypage
[root@host ~]# mv /root/mypage /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/mypage
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access this resource.</p>
</body></html>
```

El servidor web no muestra el contenido y devuelve un error `permission denied`. Se registra un evento de AVC en los archivos `/var/log/audit/audit.log` y `/var/log/messages`. Observe el comando `sealert` sugerido y el UUID en el mensaje de evento `/var/log/messages`.

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1649249057.067:212): avc: denied { getattr }
for pid=2332 comm="httpd" path="/var/www/html/mypage"
dev="vda4" ino=9322502 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file permissive=0
...output omitted
[root@host ~]# tail /var/log/messages
...output omitted...
Apr  6 08:44:19 host setroubleshoot[2547]: SELinux is preventing /usr/sbin/httpd
from getattr access on the file /var/www/html/mypage. For complete SELinux
messages run: sealert -l 95f41f98-6b56-45bc-95da-ce67ec9a9ab7
...output omitted...
```

La salida `sealert` describe el evento, incluido el proceso afectado, el archivo al que se accedió y la acción intentada y denegada. La salida incluye consejos para corregir la etiqueta del archivo, si corresponde. Los consejos adicionales describen cómo generar una nueva política para permitir la acción denegada. Use los consejos proporcionados solo cuando sea adecuado para su situación.

**Importante**

La salida `sealert` incluye una calificación de confianza, que indica el nivel de confianza de que el consejo dado mitigará la denegación. Sin embargo, es posible que ese consejo no sea adecuado para su situación.

Por ejemplo, si la denegación de AVC se debe a que el archivo denegado se encuentra en la ubicación incorrecta, el consejo que indica que se debe ajustar la etiqueta de contexto del archivo o crear una nueva política para esta ubicación y acción es técnicamente correcto, pero no es el correcto para su situación. Si la causa raíz es una ubicación o nombre de archivo incorrectos, entonces la solución adecuada es mover o cambiar el nombre del archivo y luego restaurar un contexto de archivo correcto.

```
[root@host ~]# sealert -l 95f41f98-6b56-45bc-95da-ce67ec9a9ab7
SELinux is preventing /usr/sbin/httpd from getattr access on the file /var/www/html/mypage.

***** Plugin restorecon (99.5 confidence) suggests *****

If you want to fix the label.
/var/www/html/mypage default label should be httpd_sys_content_t.
Then you can run restorecon. The access attempt may have been stopped due to
insufficient permissions to access a parent directory in which case try to change
the following command accordingly.
Do
# /sbin/restorecon -v /var/www/html/mypage

***** Plugin catchall (1.49 confidence) suggests *****

If you believe that httpd should be allowed getattr access on the mypage file by
default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp

Additional Information:
Source Context      system_u:system_r:httpd_t:s0
Target Context      unconfined_u:object_r:admin_home_t:s0
Target Objects      /var/www/html/mypage [ file ]
Source              httpd
Source Path          /usr/sbin/httpd
...output omitted...

Raw Audit Messages
type=AVC msg=audit(1649249057.67:212): avc: denied { getattr }
for pid=2332 comm="httpd" path="/var/www/html/mypage"
dev="vda4" ino=9322502 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file permissive=0
```

```
type=SYSCALL msg=audit(1649249057.67:212): arch=x86_64 syscall=newfstatat
success=no exit=EACCES a0=ffffff9c a1=7fe9c00048f8 a2=7fe9ccfc8830 a3=100
items=0 ppid=2329 pid=2332 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48
egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm=httpd exe=/usr/sbin/httpd
subj=system_u:system_r:httpd_t:s0 key=(null)
```

Hash: httpd,httpd_t,admin_home_t,file,getattr

En este ejemplo, el archivo al que se accedió está en la ubicación correcta, pero no tiene el contexto de archivo SELinux correcto. La sección **Raw Audit Messages** muestra información de la entrada del evento `/var/log/audit.log`. Use el comando `restorecon /var/www/html/mypage` para definir la etiqueta de contexto correcta. Para corregir varios archivos de manera recursiva, use el comando `restorecon -R` en el directorio principal.

Use el comando `ausearch` para buscar eventos de AVC en el archivo de registro `/var/log/audit.log`. Use la opción `-m` para especificar el tipo de mensaje AVC y la opción `-ts` para proporcionar una sugerencia de tiempo, como `recent`.

```
[root@host ~]# ausearch -m AVC -ts recent
----
time->Tue Apr 6 13:13:07 2019
type=PROCTITLE msg=audit(1554808387.778:4002):
proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1554808387.778:4002): arch=c000003e syscall=49
success=no exit=-13 a0=3 a1=55620b8c9280 a2=10 a3=7ffed967661c items=0
ppid=1 pid=9340 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1554808387.778:4002): avc: denied { name_bind }
for pid=9340 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
```

Solución de problemas de SELinux con la consola web

La consola web de RHEL incluye herramientas para solucionar problemas de SELinux. Seleccione **SELinux** del menú a la izquierda. La ventana de política de SELinux muestra el estado de cumplimiento (enforcing) actual. La sección **SELinux access control errors** enumera los problemas actuales de SELinux.

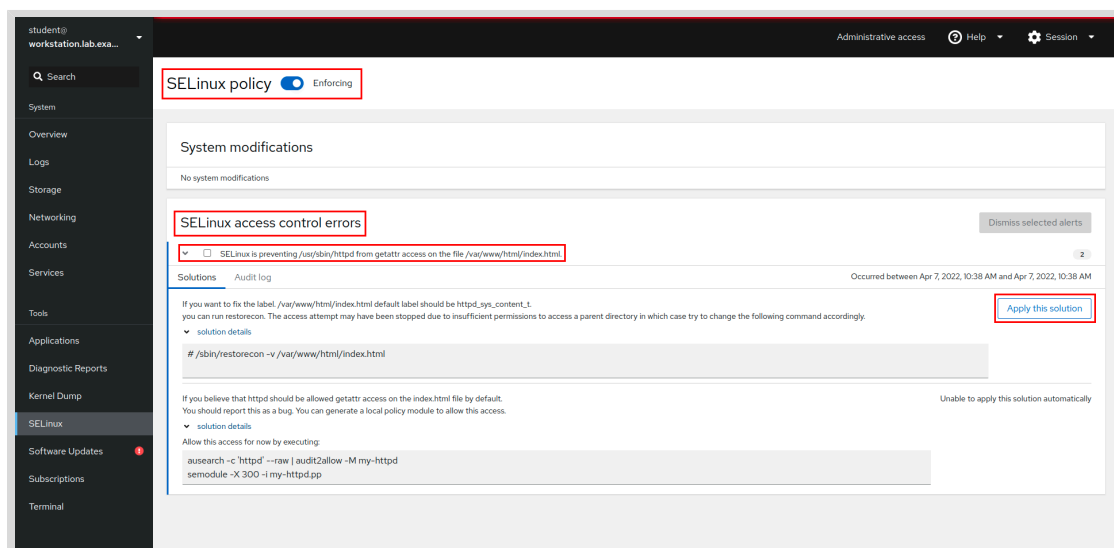


Figura 4.3: Política de SELinux y errores en la consola web

Haga clic en el carácter > para mostrar los detalles del evento. Haga clic en **solution details** (detalles de la solución) para mostrar todos los detalles y consejos del evento. Puede hacer clic en **Apply the solution** (Aplicar la solución) para aplicar los consejos dados.

Después de corregir el problema, en la sección **SELinux access control errors**, se deberá eliminar ese evento de la vista. Si aparece el mensaje **No SELinux alerts**, significa que ha corregido todos los problemas actuales de SELinux.



Referencias

Página del manual: `sealert(8)`

► Ejercicio Guiado

Investigación y resolución de problemas de SELinux

En este trabajo de laboratorio, aprende cómo solucionar problemas de denegaciones por seguridad de SELinux.

Resultados

- Obtenga experiencia con las herramientas para la solución de problemas de SELinux.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-issues
```

Instrucciones

1. Desde un navegador web en la máquina `workstation`, abra la página web `http://servera/index.html`. Un mensaje de error indica que no tiene permiso para acceder al archivo.
2. Use el comando `ssh` para iniciar sesión en `servera` con el usuario `student`. Use el comando `sudo -i` para cambiar al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

3. Use el comando `less` para visualizar el contenido del archivo `/var/log/messages`. Use el carácter `/` y busque el texto `sealert`. Presione la tecla `n` hasta llegar a la última aparición, ya que los ejercicios anteriores también pueden haber generado mensajes de SELinux. Copie el comando `sealert` sugerido para que pueda usarlo en el siguiente paso. Use la tecla `q` para salir del comando `less`.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Apr  7 04:52:18 servera setroubleshoot[20715]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html. For complete SELinux
messages run: sealert -l 9a96294a-239b-4568-8f1e-9f35b5fb472b
...output omitted...
```

- 4. Ejecute el comando sugerido `sealert`. Tenga en cuenta el contexto de origen, los objetos de destino, la política y el modo de cumplimiento (enforcing). Busque la etiqueta de contexto SELinux correcta para el archivo que el servicio `httpd` intenta proporcionar.

4.1. Ejecute el comando `sealert`.

La salida explica que el archivo `/custom/index.html` tiene una etiqueta de contexto incorrecta.

```
[root@servera ~]# sealert -l 9a96294a-239b-4568-8f1e-9f35b5fb472b
SELinux is preventing /usr/sbin/httpd from getattr access on the file /custom/
index.html.

***** Plugin catchall_labels (83.8 confidence) suggests *****

If you want to allow httpd to have getattr access on the index.html file
Then you need to change the label on /custom/index.html
Do
# semanage fcontext -a -t FILE_TYPE '/custom/index.html'
where FILE_TYPE is one of the following: NetworkManager_exec_t,
NetworkManager_log_t, NetworkManager_tmp_t, abrt_dump_oops_exec_t,
abrt_etc_t, abrt_exec_t, abrt_handle_event_exec_t, abrt_helper_exec_t,
abrt_retrace_coredump_exec_t, abrt_retrace_spool_t, abrt_retrace_worker_exec_t,
abrt_tmp_t, abrt_upload_watch_tmp_t, abrt_var_cache_t, abrt_var_log_t,
abrt_var_run_t, accountsd_exec_t, acct_data_t, acct_exec_t, admin_crontab_tmp_t,
admin_passwd_exec_t, afs_logfile_t, aide_exec_t, aide_log_t, alsa_exec_t,
alsa_tmp_t, amanda_exec_t, amanda_log_t, amanda_recover_exec_t, amanda_tmp_t,
amtu_exec_t, anacron_exec_t, anon_inodefs_t
...output omitted...

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:default_t:s0
Target Objects          /custom/index.html [ file ]
Source                  httpd
Source Path              /usr/sbin/httpd
Port                    <Unknown>
Host                    servera.lab.example.com
Source RPM Packages      httpd-2.4.51-7.el9_0.x86_64
Target RPM Packages
SELinux Policy RPM      selinux-policy-targeted-34.1.27-1.el9.noarch
Local Policy RPM        selinux-policy-targeted-34.1.27-1.el9.noarch
SELinux Enabled         True
Policy Type             targeted
Enforcing Mode          Enforcing
Host Name                servera.lab.example.com
```

```
Platform                                Linux servera.lab.example.com
                                         5.14.0-70.2.1.el9_0.x86_64 #1 SMP PREEMPT Wed Mar
                                         16 18:15:38 EDT 2022 x86_64 x86_64

Alert Count                             4
First Seen                             2022-04-07 04:51:38 EDT
Last Seen                              2022-04-07 04:52:13 EDT
Local ID                               9a96294a-239b-4568-8f1e-9f35b5fb472b

Raw Audit Messages
type=AVC msg=audit(1649321533.406:1024): avc: denied { getattr } for
pid=20464 comm="httpd" path="/custom/index.html" dev="vda4" ino=25571802
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0

...output omitted...
```

- 4.2. Verifique el contexto de SELinux para el directorio desde donde el servicio `httpd` proporciona el contenido de manera predeterminada, `/var/www/html`. El contexto de SELinux `httpd_sys_content_t` es apropiado para el archivo `/custom/index.html`.

```
[root@servera ~]# ls -ldZ /var/www/html
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 Mar 21 11:47 /
var/www/html
```

- ▶ 5. La sección `Raw Audit Messages` del comando `sealert` contiene información del archivo `/var/log/audit/audit.log`. Use el comando `ausearch` para buscar el archivo `/var/log/audit/audit.log`. La opción `-m` busca en el tipo de mensaje. La opción `-ts` busca en función del tiempo. La siguiente entrada identifica el proceso relevante y el archivo que causa la alerta. El proceso es el servidor web Apache `httpd`, el archivo es `/custom/index.html` y el contexto es `system_r:httpd_t`.

```
[root@servera ~]# ausearch -m AVC -ts today
...output omitted...
----
time->Thu Apr  7 04:52:13 2022
type=PROCTITLE msg=audit(1649321533.406:1024):
  proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1649321533.406:1024): arch=c000003e syscall=262 success=no
  exit=-13 a0=ffffff9c a1=7fefc403d850 a2=7fefc89bc830 a3=100 items=0 ppid=20461
  pid=20464 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
  sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
  subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1649321533.406:1024): avc: denied
{ getattr } for pid=20464 comm="httpd" path="/custom/index.html"
dev="vda4" ino=25571802 scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

- ▶ 6. Resuelva el problema aplicando el contexto `httpd_sys_content_t`.


```
[root@servera ~]# semanage fcontext -a \
-t httpd_sys_content_t '/custom(/.*)?'
[root@servera ~]# restorecon -Rv /custom
Relabeled /custom from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to
unconfined_u:object_r:httpd_sys_content_t:s0
```

- ▶ 7. Nuevamente, intente visualizar `http://servera/index.html`. Se muestra el mensaje `This is SERVERA..`
- ▶ 8. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-issues
```

Esto concluye la sección.

► Trabajo de laboratorio

Administración de seguridad de SELinux

En este trabajo de laboratorio, identificará problemas en los archivos de registro del sistema y ajustará la configuración de SELinux.

Resultados

- Identificar problemas en los archivos de registro del sistema.
- Ajustar la configuración de SELinux.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-review
```

Instrucciones

1. Inicie sesión en la máquina `serverb` como el usuario `student` y cambie al usuario `root`.
2. Desde un navegador web en la máquina `workstation`, vea la página web `http://serverb/lab.html`. Debe ver el siguiente mensaje de error: `You do not have permission to access this resource.`
3. Investigue e identifique el problema de SELinux que impide que el servicio Apache proporcione el contenido web.
4. Muestre el contexto de SELinux del nuevo directorio de documentos HTTP y el directorio de documentos HTTP original. Resuelva el problema de SELinux que evita que el servidor Apache proporcione el contenido web.
5. Verifique que el servidor Apache ahora pueda proporcionar contenido web.
6. Regrese a la máquina `workstation` como el usuario `student`.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade selinux-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-review
```

Esto concluye la sección.

► Solución

Administración de seguridad de SELinux

En este trabajo de laboratorio, identificará problemas en los archivos de registro del sistema y ajustará la configuración de SELinux.

Resultados

- Identificar problemas en los archivos de registro del sistema.
- Ajustar la configuración de SELinux.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start selinux-review
```

Instrucciones

1. Inicie sesión en la máquina `serverb` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

2. Desde un navegador web en la máquina `workstation`, vea la página web `http://serverb/lab.html`. Debe ver el siguiente mensaje de error: `You do not have permission to access this resource.`
3. Investigue e identifique el problema de SELinux que impide que el servicio Apache proporcione el contenido web.
 - 3.1. Vea el contenido del archivo `/var/log/messages`. Use la tecla `/` y busque la cadena `sealert`. Use la tecla `q` para salir del comando `less`.

```
[root@serverb ~]# less /var/log/messages
...output omitted...
Apr  7 06:16:15 serverb setroubleshoot[26509]: failed to retrieve rpm info for /
lab-content/lab
b.html
Apr  7 06:16:17 serverb setroubleshoot[26509]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html. For complete SELinux
messages run: sealert -l 35c9e452-2552-4ca3-8217-493b72ba6d0b
Apr  7 06:16:17 serverb setroubleshoot[26509]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html
...output omitted...
```

- 3.2. Ejecute el comando sugerido `sealert`. Tenga en cuenta el contexto de origen, los objetos de destino, la política y el modo de cumplimiento (enforcing).

```
[root@serverb ~]# sealert -l 35c9e452-2552-4ca3-8217-493b72ba6d0b
SELinux is preventing /usr/sbin/httpd from getattr access on the file /lab-
content/lab.html.

***** Plugin catchall_labels (83.8 confidence) suggests *****

If you want to allow httpd to have getattr access on the lab.html file
Then you need to change the label on /lab-content/lab.html
Do
# semanage fcontext -a -t FILE_TYPE '/lab-content/lab.html'
where FILE_TYPE is one of the following:
...output omitted...

Additional Information:
Source Context      system_u:system_r:httpd_t:s0
Target Context      unconfined_u:object_r:default_t:s0
Target Objects      /lab-content/lab.html [ file ]
Source              httpd
Source Path          /usr/sbin/httpd
Port                <Unknown>
Host                serverb.lab.example.com
Source RPM Packages httpd-2.4.51-7.el9_0.x86_64
Target RPM Packages
SELinux Policy RPM  selinux-policy-targeted-34.1.27-1.el9.noarch
Local Policy RPM    selinux-policy-targeted-34.1.27-1.el9.noarch
Selinux Enabled     True
Policy Type         targeted
Enforcing Mode      Enforcing
Host Name           serverb.lab.example.com
Platform            Linux serverb.lab.example.com
                    5.14.0-70.2.1.el9_0.x86_64 #1 SMP PREEMPT Wed Mar
                    16 18:15:38 EDT 2022 x86_64 x86_64

Alert Count         8
First Seen          2022-04-07 06:14:45 EDT
Last Seen           2022-04-07 06:16:12 EDT
Local ID            35c9e452-2552-4ca3-8217-493b72ba6d0b

Raw Audit Messages
```

```
type=AVC msg=audit(1649326572.86:407): avc: denied { getattr } for
pid=10731 comm="httpd" path="/lab-content/lab.html" dev="vda4" ino=18192752
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0
```

```
type=SYSCALL msg=audit(1649326572.86:407): arch=x86_64 syscall=newfstatat
success=no exit=EACCES a0=ffffff9c a1=7f7c8c0457c0 a2=7f7c887f7830 a3=100 items=0
ppid=10641 pid=10731 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48
egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm=httpd exe=/usr/sbin/httpd
subj=system_u:system_r:httpd_t:s0 key=(null)
```

Hash: httpd,httpd_t,default_t,file,getattr

- 3.3. La sección **Raw Audit Messages** del comando **sealert** contiene información del archivo `/var/log/audit/audit.log`. Busque el archivo `/var/log/audit/audit.log`. La opción **-m** busca en el tipo de mensaje. La opción **ts** busca en función del tiempo. La siguiente entrada identifica el proceso relevante y el archivo que causa la alerta. El proceso es el servidor web Apache **httpd**, el archivo es `/lab-content/lab.html` y el contexto es **system_r:httpd_t**.

```
[root@serverb ~]# ausearch -m AVC -ts recent
...output omitted...
----
time->Thu Apr  7 06:16:12 2022
type=PROCTITLE msg=audit(1649326572.086:407):
proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1649326572.086:407): arch=c000003e syscall=262 success=no
exit=-13 a0=ffffff9c a1=7f7c8c0457c0 a2=7f7c887f7830 a3=100 items=0 ppid=10641
pid=10731 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48
sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1649326572.086:407): avc: denied { getattr } for
pid=10731 comm="httpd" path="/lab-content/lab.html" dev="vda4" ino=18192752
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0
```

4. Muestre el contexto de SELinux del nuevo directorio de documentos HTTP y el directorio de documentos HTTP original. Resuelva el problema de SELinux que evita que el servidor Apache proporcione el contenido web.

- 4.1. Compare el contexto de SELinux para los directorios `/lab-content` y `/var/www/html`.

```
[root@serverb ~]# ls -dZ /lab-content /var/www/html
unconfined_u:object_r:default_t:s0 /lab-content
system_u:object_r:httpd_sys_content_t:s0 /var/www/html
```

- 4.2. Cree una regla de contextos de archivos que establezca el tipo predeterminado en `httpd_sys_content_` para el directorio `/lab-content` y todos los archivos en él.

```
[root@serverb ~]# semanage fcontext -a \
-t httpd_sys_content_t '/lab-content(/.*)?'
```

- 4.3. Corrija el contexto de SELinux para los archivos en el directorio `/lab-content`.

```
[root@serverb ~]# restorecon -R /lab-content/
```

5. Verifique que el servidor Apache ahora pueda proporcionar contenido web.

- 5.1. Use su navegador web para actualizar el enlace `http://serverb/lab.html`. Si se muestra el contenido, su problema está resuelto.

```
This is the html file for the SELinux final lab on SERVERB.
```

6. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade selinux-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish selinux-review
```

Esto concluye la sección.

Resumen

- Use los comandos `getenforce` y `setenforce` para administrar el modo de SELinux de un sistema.
- El comando `semanage` administra las reglas de políticas de SELinux. El comando `restorecon` aplica el contexto que define la política.
- Los booleanos son opciones que modifican el comportamiento de la política de SELinux. Puede habilitarlos o deshabilitarlos para ajustar la política.
- El comando `sealert` muestra información útil que ayuda con la solución de problemas de SELinux.

capítulo 5

Administración de almacenamiento básico

Meta

Crear y administrar dispositivos de almacenamiento, particiones, sistemas de archivos y espacios de intercambio (swap) desde la línea de comandos.

Objetivos

- Crear particiones de almacenamiento, formatearlas con sistemas de archivos y montarlas para su uso.
- Crear y administrar espacios de intercambio (swap) para complementar la memoria física.

Secciones

- Adición de particiones, sistemas de archivos y montajes persistentes (y ejercicio guiado)
- Administración de espacio de intercambio (swap) (y ejercicio guiado)

Trabajo de laboratorio

Administración de almacenamiento básico

Adición de particiones, sistemas de archivos y montajes persistentes

Objetivos

Crear particiones de almacenamiento, formatearlas con sistemas de archivos y montarlas para su uso.

Discos de partición

La partición del disco divide un disco duro en varias *particiones* de almacenamiento lógico. Puede usar particiones para dividir el almacenamiento en función de diferentes requisitos y esta división proporciona numerosas ventajas:

- Limitar espacio disponible para aplicaciones o usuarios.
- Separar archivos de programa y de sistemas operativos de archivos de usuarios.
- Crear un área separada para el intercambio (swap) de memoria.
- Limitar el uso de espacio en disco para mejorar el rendimiento de herramientas de diagnóstico e imágenes de copia de seguridad.

Esquema de partición MBR

El esquema de partición *Master Boot Record* (MBR) es el estándar en los sistemas que ejecutan el firmware del BIOS. Este esquema soporta un máximo de cuatro particiones primarias. En sistemas Linux, con las particiones ampliadas y lógicas, puede crear un máximo de 15 particiones. Con un tamaño de partición de 32 bits, los discos que están particionados con MBR pueden tener un tamaño de hasta 2 TiB.



Figura 5.1: Particiones MBR del dispositivo de almacenamiento /dev/vdb

El límite de tamaño de disco y partición de 2 TiB ahora es una limitación común y restrictiva. En consecuencia, el esquema de MBR heredado es reemplazado por el esquema de partición *GUID Partition Table* (GPT).

Esquema de partición GPT

Para los sistemas que ejecutan el firmware de *Unified Extensible Firmware Interface* (UEFI), GPT es el estándar para la partición del disco y aborda las limitaciones del esquema de MBR. Un GPT proporciona un máximo de 128 particiones. El esquema GPT asigna 64 bits para direcciones de bloques lógicos, para soportar particiones y discos de hasta ocho zebibytes (ZiB) u ocho mil millones de tebibytes (TiB).



Figura 5.2: Particiones GPT del dispositivo de almacenamiento /dev/vdb

La partición GPT ofrece características y ventajas adicionales sobre MBR. Un GPT usa un *identificador único global* (GUID) para identificar cada disco y partición. Un GPT hace que la tabla de particiones sea redundante, con el GPT principal en el cabezal del disco y un GPT secundario de respaldo en el extremo del disco. GPT usa la suma de comprobación para detectar errores en la tabla de particiones y el encabezado de GPT.

Administrar particiones

Un administrador puede usar un programa *editor de particiones* para cambiar particiones de un disco, como crear y eliminar particiones y cambiar los tipos de partición.



nota

¿Qué editor de particiones debería usar? Si bien los profesionales de TI tienen opiniones sólidas sobre las distinciones de funciones, cada editor aquí detallado realiza correctamente tareas comunes de preparación de discos.

- `fdisk` es uno de los favoritos históricos y ha soportado particiones GPT durante años.
- `gdisk` y otras variantes `fdisk` se crearon inicialmente para soportar GPT.
- `parted` y la librería `libparted` han sido el estándar de RHEL durante años.
- El instalador Anaconda continúa usando la librería `libparted`.
- `gnome-disk` es la herramienta gráfica predeterminada de GNOME, que reemplaza al upstream `gparted`.
- Casi todos los editores de la CLI son buenos para el scripting y `parted` fue diseñado para ello.

Los administradores pueden usar el editor de particiones `parted` tanto para el esquema de partición GPT como MBR. El comando `parted` toma el nombre del dispositivo de todo el disco como primer argumento seguido de subcomandos. En el siguiente ejemplo, se usa el subcomando `print` para visualizar la tabla de particiones en el disco `/dev/vda`.

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
  1      1049kB  10.7GB  10.7GB  primary  xfs          boot
  2      10.7GB  53.7GB  42.9GB  primary  xfs
```

Use el comando `parted` sin un subcomando para abrir una sesión de partición interactiva.

```
[root@host ~]# parted /dev/vda
GNU Parted 3.4
Using /dev/vda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  10.7GB  10.7GB  primary xfs          boot
  2      10.7GB  53.7GB  42.9GB  primary xfs
```

```
(parted) quit
[root@host ~]#
```

De forma predeterminada, el comando `parted` muestra los tamaños en potencias de 10 (KB, MB, GB). Puede cambiar el tamaño de la unidad con el parámetro `unit`, que acepta los siguientes valores:

- **S** para sector
- **B** para byte
- **MiB**, **GiB** o **TiB** (potencias de 2)
- **MB**, **GB** o **TB** (potencias de 10)

```
[root@host ~]# parted /dev/vda unit s print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start      End          Size          Type    File system  Flags
  1      2048s      20971486s    20969439s     primary xfs          boot
  2      20971520s  104857535s   83886016s     primary xfs
```

Como se muestra en el ejemplo anterior, también puede especificar varios subcomandos (aquí, `unit` y `print`) en la misma línea.

Escritura de la tabla de particiones en un disco nuevo

Para particionar una nueva unidad, primero escriba una etiqueta de disco. La etiqueta del disco indica qué esquema de partición usar. Use `parted` para escribir una etiqueta de disco MBR o una etiqueta de disco GPT.

```
[root@host ~]# parted /dev/vdb mklabel msdos

[root@host ~]# parted /dev/vdb mklabel gpt
```

**Advertencia**

El subcomando `mklabel` borra la tabla de particiones existente. Use el subcomando `mklabel` cuando la intención sea volver a usar el disco sin tener en cuenta los datos existentes. Si una nueva etiqueta cambia los límites de la partición, todos los datos en los sistemas de archivos existentes se vuelven inaccesibles.

Creación de particiones de MBR

Las siguientes instrucciones crean una partición de disco MBR. Especifique el dispositivo de disco donde creará la partición.

Ejecute el comando `parted` y especifique el nombre del dispositivo de disco como un argumento, para comenzar en el modo interactivo. La sesión muestra (`parted`) como prompt de subcomando.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Use el subcomando `mkpart` para crear una partición primaria o ampliada.

```
(parted) mkpart
Partition type? primary/extended? primary
```

**nota**

Si necesitan más de cuatro particiones en un disco particionado con MBR, cree tres particiones primarias y una partición ampliada. La partición ampliada sirve como contenedor dentro del cual se pueden crear varias particiones lógicas.

Indique el tipo de sistema de archivos que desea crear en la partición, como `xfs` o `ext4`. Este valor no crea el sistema de archivos, pero es solo una etiqueta de tipo de partición útil.

```
File system type? [ext2]? xfs
```

Para obtener la lista de los tipos de sistemas de archivos soportados, use el siguiente comando:

```
[root@host ~]# parted /dev/vdb help mkpart
...output omitted...
mkpart PART-TYPE [FS-TYPE] START END      make a partition

PART-TYPE is one of: primary, logical, extended
FS-TYPE is one of: udf, btrfs, nilfs2, ext4, ext3, ext2, f2fs, fat32, fat16,
hfsx, hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs,
reiserfs, hp-ufs, sun-ufs, xfs, apfs2, apfs1, asfs, amufs5, amufs4, amufs3,
amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
```

```
'mkpart' makes a partition without creating a new file system on the
partition. FS-TYPE may be specified to set an appropriate partition
ID.
```

Especifique el sector en el disco donde se iniciará la nueva partición.

```
Start? 2048s
```

El sufijo `s` proporciona el valor en sectores o usa los sufijos `MiB`, `GiB`, `TiB`, `MB`, `GB` o `TB`. El comando `parted` toma de manera predeterminada el sufijo `MB`. El comando `parted` redondea los valores proporcionados para satisfacer las restricciones del disco.

Cuando se inicia el comando `parted`, recupera la topología del disco del dispositivo, como el tamaño del bloque físico del disco. El comando `parted` asegura que la posición de inicio que proporcione alinea correctamente la partición con la estructura del disco para optimizar el rendimiento. Si la posición de inicio genera una partición desalineada, el comando `parted` muestra una advertencia. Con la mayoría de los discos, un sector de inicio que es un múltiplo de 2048 es seguro.

Especifique el sector del disco donde debería terminar la nueva partición y salga de `parted`. Puede especificar el final como tamaño o como ubicación final.

```
End? 1000MB
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

Al proporcionar la posición final, el comando `parted` actualiza la tabla de particiones en el disco con los detalles de la nueva partición.

Ejecute el comando `udevadm settle`. Este comando espera a que el sistema detecte la nueva partición y cree el archivo de dispositivo asociado en el directorio `/dev`. El prompt regresa cuando se realiza la tarea.

```
[root@host ~]# udevadm settle
```

Como alternativa al modo interactivo, puede crear una partición en un solo comando:

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

Creación de particiones GPT

El esquema GTP también usa el comando `parted` para crear particiones. Especifique el dispositivo de disco donde creará la partición.

Como usuario `root`, ejecute el comando `parted` y especifique el nombre del dispositivo de disco como un argumento.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Use el subcomando `mkpart` para comenzar a crear la partición. Con el esquema GPT, cada partición recibe un nombre.

```
(parted) mkpart
Partition name? []? userdata
```

Indique el tipo de sistema de archivos que desea crear en la partición, como `xfs` o `ext4`. Este valor no crea el sistema de archivos, pero es una etiqueta de tipo de partición útil.

```
File system type? [ext2]? xfs
```

Especifique el sector del disco donde se iniciará la nueva partición.

```
Start? 2048s
```

Especifique el sector del disco donde debería terminar la nueva partición y salga de `parted`. Al proporcionar la posición final, el comando `parted` actualiza la tabla de particiones el GPT en el disco con los detalles de la nueva partición.

```
End? 1000MB
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

Ejecute el comando `udevadm settle`. Este comando espera a que el sistema detecte la nueva partición y cree el archivo de dispositivo asociado en el directorio `/dev`. El prompt regresa cuando se realiza la tarea.

```
[root@host ~]# udevadm settle
```

Como alternativa al modo interactivo, puede crear una partición en un solo comando:

```
[root@host ~]# parted /dev/vdb mkpart userdata xfs 2048s 1000MB
```

Eliminación de particiones

Las siguientes instrucciones se aplican a los esquemas de partición MBR y GPT. Especifique el disco que contiene la partición que se eliminará.

Ejecute el comando `parted` con el dispositivo de disco como único argumento.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Identifique el número de partición de la partición que se eliminará.

```
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number   Start    End      Size    File system  Name      Flags
  1       1049kB   1000MB   999MB   xfs          usersdata
```

Elimine la partición y salga de `parted`. El subcomando `rm` elimina inmediatamente la partición de la tabla de particiones en el disco.

```
(parted) rm 1
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

Como alternativa al modo interactivo, puede eliminar una partición en un solo comando:

```
[root@host ~]# parted /dev/vdb rm 1
```

Creación de sistemas de archivos

Después de la creación de un dispositivo de bloque, el siguiente paso es agregarle un sistema de archivos. Red Hat Enterprise Linux soporta varios tipos de sistemas de archivos y XFS es el valor predeterminado recomendado.

Con el usuario `root`, use el comando `mkfs.xfs` para aplicar un sistema de archivos XFS a un dispositivo de bloque. Para un sistema de archivos `ext4`, use el comando `mkfs.ext4`.

```
[root@host ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1            isize=512    agcount=4, agsize=60992 blks
                               =               sectsz=512   attr=2, projid32bit=1
                               =               crc=1     finobt=1, sparse=1, rmapbt=0
                               =               reflink=1  bigtime=1 inobtcount=1
data      =                     bsize=4096   blocks=243968, imaxpct=25
                               =               sunit=0    swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log       bsize=4096   blocks=1566, version=2
                               =               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none               extsz=4096   blocks=0, rtextents=0
```


Montaje de sistemas de archivos

Después de agregar el sistema de archivos, el último paso es montar el sistema de archivos en un directorio en la estructura de directorios. Cuando monta un sistema de archivos en la jerarquía del directorio, las utilidades de espacio de usuarios pueden acceder o escribir archivos en el dispositivo.

Montaje manual de sistemas de archivos

Use el comando `mount` para conectar manualmente un dispositivo a una ubicación de directorio de *punto de montaje*. El comando `mount` requiere un dispositivo y un punto de montaje y puede incluir opciones de montaje del sistema de archivos. Las opciones del sistema de archivos personalizan el comportamiento del sistema de archivos.

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

También puede usar el comando `mount` para ver los sistemas de archivos montados actualmente, los puntos de montaje y sus opciones.

```
[root@host ~]# mount | grep vdb1
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Montaje de manera persistente de sistemas de archivos

Montar manualmente un sistema de archivos es una buena manera de verificar que un dispositivo formateado sea accesible y funcione de la manera esperada. Sin embargo, cuando el servidor se reinicia, el sistema no vuelve a montar automáticamente el sistema de archivos.

Para configurar el sistema para que monte automáticamente el sistema de archivos durante el arranque del sistema, agregue una entrada al archivo `/etc/fstab`. Este archivo de configuración enumera los sistemas de archivos para montar en el arranque del sistema.

El archivo `/etc/fstab` es un archivo delimitado por espacios en blanco con seis campos por línea.

```
[root@host ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Thu Apr 5 12:05:19 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=a8063676-44dd-409a-b584-68be2c9f5570 / xfs defaults 0 0
UUID=7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata xfs defaults 0 0
```

El primer campo especifica el dispositivo. Este ejemplo usa un UUID para especificar el dispositivo. Los sistemas de archivos crean y almacenan el UUID en su superbloque de partición en el momento de la creación. O puede usar el archivo del dispositivo, como `/dev/vdb1`.

El segundo campo es el punto de montaje del directorio, desde el cual se podrá acceder al dispositivo de bloque en la estructura del directorio. El punto de montaje debe existir; de lo contrario, créelo con el comando `mkdir`.

El tercer campo contiene el tipo del sistema de archivos, como `xfs` o `ext4`.

El cuarto campo es la lista de opciones separada por comas para aplicar al dispositivo. `defaults` es un conjunto de opciones que se usan comúnmente. La página del manual `mount(8)` documenta las otras opciones disponibles.

El quinto campo es usado por el comando `dump` para hacer una copia de seguridad del dispositivo. Otras aplicaciones de copia de seguridad no suelen usar este campo.

El último campo, el campo de orden de `fsck`, determina si debería ejecutarse el comando `fsck` en el arranque del sistema para verificar que los sistemas de archivos estén limpios. El valor en este campo indica el orden en el que debe ejecutarse `fsck`. Para sistemas de archivos XFS, establezca este campo en `0` porque XFS no usa `fsck` para comprobar el estado de su sistema de archivos. Para sistemas de archivos `ext4`, configúrelo en `1` para el sistema de archivos `root` y en `2` para los otros sistemas de archivos `ext4`. Al usar esta notación, la utilidad `fsck` primero procesa el sistema de archivos `root` y, luego, verifica los sistemas de archivos en discos separados al mismo tiempo, y los sistemas de archivos en el mismo disco en secuencia.



nota

Si hay una entrada incorrecta en `/etc/fstab`, es posible que la máquina no pueda volver a arrancarse. Verifique que una entrada sea válida desmontando manualmente el nuevo sistema de archivos y, luego, use `mount /mountpoint` para leer el archivo `/etc/fstab` y vuelva a montar el sistema de archivos con las opciones de montaje de esa entrada. Si el comando `mount` arroja un error, corrijalo antes de volver a arrancar la máquina.

De manera alternativa, use el comando `findmnt --verify` para analizar el archivo `/etc/fstab` para la usabilidad de la partición.

Cuando agregue o elimine una entrada en el archivo `/etc/fstab`, ejecute el comando `systemctl daemon-reload`, o reinicie el servidor, para asegurar que el daemon `systemd` cargue y use la nueva configuración.

```
[root@host ~]# systemctl daemon-reload
```

Red Hat recomienda el uso de UUID para montar los sistemas de archivos de forma persistente, dado que los nombres de dispositivos de bloques pueden cambiar en determinadas situaciones, como en el caso de que un proveedor de la nube cambie la capa de almacenamiento subyacente de una máquina virtual, o que los discos se detecten en un orden diferente en un arranque del sistema. El nombre del archivo del dispositivo de bloque puede cambiar, pero el UUID permanece constante en el superbloque del sistema de archivos.

Use el comando `lsblk --fs` para escanear los dispositivos de bloque que están conectados a una máquina y recuperar los UUID del sistema de archivos.

```
[root@host ~]# lsblk --fs
```

NAME	FSTYPE	FSVER	LABEL	UUID	FSAVAIL	FSUSE%	MOUNTPOINTS
vda							
└─vda1							
└─vda2	xfs		boot	49dd...75fdf	312M	37%	/boot
└─vda3	xfs		root	8a90...ce0da	4.8G	48%	/



Referencias

info parted (*GNU Parted User Manual*)

Páginas del manual: parted(8), mkfs(8), mount(8), lsblk(8) y fstab(5)

Para obtener más información, consulte la guía *Configuring and Managing File Systems* en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/managing_file_systems/index

► Ejercicio Guiado

Adición de particiones, sistemas de archivos y montajes persistentes

En este ejercicio, crea una partición en un nuevo dispositivo de almacenamiento, la formateará con un sistema de archivos XFS, la configurará para que se monte en el arranque y la montará para su uso.

Resultados

- Use `parted`, `mkfs.xfs` y otros comandos para crear una partición en un disco nuevo, formatearla y montarla de manera persistente.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start storage-partitions
```

Instrucciones

- 1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Cree una etiqueta de disco `msdos` en el dispositivo `/dev/vdb`.

```
[root@servera ~]# parted /dev/vdb mklabel msdos
Information: You may need to update /etc/fstab.
```

- 3. Agregue una partición principal de 1 GB. Para una correcta alineación, inicie la partición en el sector 2048. Establezca el tipo de sistema de archivo de la partición en XFS.

- 3.1. Use el modo interactivo `parted` para poder crear la partición.

```
[root@servera ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

```
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? xfs
Start? 2048s
End? 1001MB
(parted) quit
Information: You may need to update /etc/fstab.
```

Dado que la partición comienza en el sector 2048, el comando anterior establece la posición final en 1001 MB para obtener un tamaño de partición de 1000 MB (1 GB).

Como alternativa, puede realizar la misma operación con el siguiente comando no interactivo: `parted /dev/vdb mkpart primary xfs 2048s 1001 MB`

3.2. Verifique su trabajo enumerando las particiones en el dispositivo `/dev/vdb`.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type    File system  Flags
  1      1049kB  1001MB  1000MB  primary
```

3.3. Ejecute el comando `udevadm settle`. Este comando espera a que el sistema registre la nueva partición y regresa cuando finaliza.

```
[root@servera ~]# udevadm settle
```

► 4. Formatee la nueva partición con el sistema de archivos XFS.

```
[root@servera ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1             isize=512    agcount=4, agsize=61056 blks
=                               sectsz=512   attr=2, projid32bit=1
=                               crc=1       finobt=1, sparse=1, rmapbt=0
=                               reflink=1   bigtime=1 inobtcount=1
data      =                     bsize=4096   blocks=244224, imaxpct=25
=                               sunit=0        swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log      bsize=4096   blocks=1566, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
```

► 5. Configure el nuevo sistema de archivos para montarlo de forma persistente en el directorio `/archive`.

5.1. Cree el directorio `/archive`.

```
[root@servera ~]# mkdir /archive
```

- 5.2. Descubra el UUID del dispositivo `/dev/vdb1`. El UUID en la salida probablemente sea diferente en su sistema.

```
[root@servera ~]# lsblk --fs /dev/vdb
NAME      FSTYPE FSVER LABEL UUID                               FSAVAIL FSUSE%
MOUNTPOINTS
vdb
└─vdb1 xfs               881e856c-37b1-41e3-b009-ad526e46d987
```

- 5.3. Agregue una entrada al archivo `/etc/fstab`. Reemplace el UUID con el que descubrió en el paso anterior.

```
...output omitted...
UUID=881e856c-37b1-41e3-b009-ad526e46d987 /archive xfs defaults 0 0
```

- 5.4. Actualice el daemon `systemd` para que el sistema registre la nueva configuración del archivo `/etc/fstab`.

```
[root@servera ~]# systemctl daemon-reload
```

- 5.5. Monte el nuevo sistema de archivos con la nueva entrada en el archivo `/etc/fstab`.

```
[root@servera ~]# mount /archive
```

- 5.6. Verifique que el nuevo sistema de archivos esté montado en el directorio `/archive`.

```
[root@servera ~]# mount | grep /archive
/dev/vdb1 on /archive type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

- 6. Reinicie `servera`. Después de reiniciar el servidor, inicie sesión y verifique que el dispositivo `/dev/vdb1` esté montado en el directorio `/archive`. Cuando finalice, cierre sesión de `servera`.

- 6.1. Reinicie `servera`.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- 6.2. Espere a que se reinicie `servera` e inicie sesión con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 6.3. Verifique que el dispositivo `/dev/vdb1` esté montado en el directorio `/archive`.

```
[student@servera ~]$ mount | grep /archive  
/dev/vdb1 on /archive type xfs  
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

6.4. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish storage-partitions
```

Esto concluye la sección.

Administración de espacio de intercambio (swap)

Objetivos

Crear y administrar espacios de intercambio (swap) para complementar la memoria física.

Conceptos de espacio de intercambio (swap)

Un *espacio de intercambio (swap)* es un área del disco bajo el control del subsistema de administración de memoria del kernel Linux. El kernel usa espacio de intercambio (swap) para complementar la memoria RAM del sistema al contener páginas inactivas en la memoria. La *memoria virtual* de un sistema abarca la memoria RAM del sistema y el espacio de intercambio (swap) combinados.

Cuando el uso de la memoria en un sistema supera un límite definido, el kernel busca en la memoria RAM páginas de memoria que están asignadas a los procesos, pero inactivas. El kernel escribe las páginas inactivas en el área de intercambio (swap) y, luego, reasigna la página RAM a otros procesos. Si un programa requiere acceso a una página en el disco, el kernel localiza otra página de memoria inactiva, la escribe en el disco y vuelve a convocar la página necesaria desde el área de intercambio (swap).

Dado que las áreas de intercambio (swap) residen en el disco, el intercambio es lento cuando se lo compara con la memoria RAM. Si bien el espacio de intercambio (swap) aumenta la memoria RAM del sistema, no debe considerar el espacio de intercambio como una solución sostenible para una RAM insuficiente para su carga de trabajo.

Cálculo del espacio de intercambio (swap)

Los administradores deben establecer el tamaño del espacio de intercambio (swap) en función de la carga de trabajo de memoria en el sistema. Los proveedores de aplicaciones a veces ofrecen recomendaciones para calcular el espacio de intercambio (swap). La siguiente tabla proporciona instrucciones basadas en la memoria física total.

Recomendaciones de memoria RAM y espacio de intercambio (swap)

RAM	Espacio de intercambio (swap)	Espacio de intercambio (swap) si se permite la hibernación
2 GB o menos	Dos veces la memoria RAM	Tres veces la memoria RAM
Entre 2 GB y 8 GB	Igual que la memoria RAM	Dos veces la memoria RAM
Entre 8 GB y 64 GB	Al menos 4 GB	1,5 veces la memoria RAM
Más de 64 GB	Al menos 4 GB	No se recomienda la hibernación.

La función de hibernación de los equipos portátiles y de escritorio usa el espacio de intercambio (swap) para guardar el contenido de la memoria RAM antes de apagar el sistema. Cuando vuelve a encender el sistema, el kernel restaura el contenido de la memoria RAM desde el espacio

de intercambio (swap) y no necesita un arranque completo. Para esos sistemas, el espacio de intercambio (swap) debe ser mayor que la cantidad de memoria RAM.

El artículo de la base de conocimiento en Referencias al final de esta sección brinda más orientación sobre el tamaño del espacio de intercambio (swap).

Creación de un espacio de intercambio (swap)

Para crear un espacio de intercambio (swap), debe realizar los siguientes pasos:

- Cree una partición con un tipo de sistema de archivos de `linux-swap`.
- Coloque una firma de intercambio (swap) en el dispositivo.

Creación de una partición de intercambio (swap)

Use el comando `parted` para crear una partición del tamaño adecuado y establezca su tipo de sistema de archivos en `linux-swap`. Anteriormente, las herramientas determinaban a partir del tipo de sistema de archivos de la partición si el dispositivo debía activarse; no obstante, eso ya no sucede. Si bien las utilidades ya no usan el tipo de sistema de archivos de la partición, los administradores pueden determinar rápidamente el propósito de la partición a partir de ese tipo.

En el siguiente ejemplo, se crea una partición de 256 MB.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.4
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049KB  1001MB  1000MB                data

(parted) mkpart
Partition name? []? swap1
File system type? [ext2]? linux-swap
Start? 1001MB
End? 1257MB
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049KB  1001MB  1000MB                data
  2      1001MB  1257MB  256MB    linux-swap(v1) swap1

(parted) quit
```

```
Information: You may need to update /etc/fstab.
```

```
[root@host ~]#
```

Después de crear la partición, ejecute el comando `udevadm settle`. Este comando espera a que el sistema detecte la nueva partición y cree el archivo de dispositivo asociado en el directorio `/dev`. El comando regresa solo cuando finaliza.

```
[root@host ~]# udevadm settle
```

Formateo del espacio de intercambio (swap)

El comando `mkswap` aplica una firma de intercambio (swap) al dispositivo. A diferencia de otras utilidades de formateo, el comando `mkswap` escribe un único bloque de datos al inicio del dispositivo, dejando el resto del dispositivo sin formatear de modo que el kernel pueda usarlo para almacenar páginas de memoria.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

Activación de un espacio de intercambio (swap)

Puede usar el comando `swapon` para activar un espacio de intercambio (swap) formateado.

Use `swapon` con el dispositivo como parámetro o use `swapon -a` para activar todos los espacios de intercambio (swap) detallados en el archivo `/etc/fstab`. Use los comandos `swapon --show` y `free` para inspeccionar los espacios de intercambio (swap) disponibles.

```
[root@host ~]# free
```

	total	used	free	shared	buff/cache	available
Mem:	1873036	134688	1536436	16748	201912	1576044
Swap:	0	0	0			

```
[root@host ~]# swapon /dev/vdb2
[root@host ~]# free
```

	total	used	free	shared	buff/cache	available
Mem:	1873036	135044	1536040	16748	201952	1575680
Swap:	249852	0	249852			

Puede desactivar un espacio de intercambio (swap) con el comando `swapoff`. Si el espacio de intercambio (swap) tiene páginas escritas, el comando `swapoff` intenta mover esas páginas a otros espacios de intercambio activos o volver a la memoria. Si el comando `swapoff` no puede escribir datos en otros lugares, falla con un error y el espacio de intercambio (swap) permanece activo.

Activación del espacio de intercambio (swap) de forma persistente

Cree una entrada en el archivo `/etc/fstab` para garantizar un espacio de intercambio (swap) activo en el arranque del sistema. En el siguiente ejemplo, se muestra una línea típica en el archivo `/etc/fstab` basada en el espacio de intercambio (swap) creado anteriormente.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881  swap  swap  defaults  0 0
```

El ejemplo anterior usa el UUID como el primer campo. Cuando formatea el dispositivo, el comando `mkswap` muestra ese UUID. Si perdió la salida de `mkswap`, luego use el comando `lsblk - -fs`. Como alternativa, puede usar el nombre del dispositivo en el primer campo.

El segundo campo se reserva típicamente para el punto de montaje. Sin embargo, para dispositivos de intercambio (swap), que no son accesibles a través de la estructura del directorio, este campo toma el valor del marcador de posición `swap`. La página del manual `fstab(5)` usa un valor de marcador de posición de `none`; sin embargo, un valor de `swap` permite mensajes de error más informativos en el caso de que algo salga mal.

El tercer campo es el tipo de sistema de archivos. El tipo de sistemas de archivos para un espacio de intercambio (swap) es `swap`.

El cuarto campo es para opciones. El ejemplo usa la opción `defaults`. La opción `defaults` incluye la opción de montaje `auto`, que activa el espacio de intercambio (swap) automáticamente en el arranque del sistema.

Los dos campos finales son el indicador `dump` y el orden `fsck`. Los espacios de intercambio (swap) no requieren copias de seguridad ni revisión del sistema de archivos y, por lo tanto, estos campos deben establecerse en cero.

Cuando agregue o elimine una entrada en el archivo `/etc/fstab`, ejecute el comando `systemctl daemon-reload`, o reinicie el servidor, para `systemd` para registrar la nueva configuración.

```
[root@host ~]# systemctl daemon-reload
```

Configuración de la prioridad de espacio de intercambio (swap)

De forma predeterminada, el sistema usa espacios de intercambio (swap) en serie, lo que significa que el kernel usa el primer espacio de intercambio activado hasta que esté lleno y, luego, el kernel empieza a usar el segundo espacio de intercambio. En cambio, puede definir una prioridad para cada espacio de intercambio (swap) para forzar un orden particular.

Para establecer la prioridad, use la opción `pri` en el archivo `/etc/fstab`. El kernel usa el espacio de intercambio (swap) con la prioridad más alta primero. La prioridad predeterminada es `-2`.

En el siguiente ejemplo, se muestran tres espacios de intercambio (swap) definidos en el archivo `/etc/fstab`. El kernel usa la última entrada primero, ya que su prioridad se establece en 10. Cuando ese espacio está lleno, usa la segunda entrada, ya que su prioridad está definida en 4. Por último, usa la primera entrada, que tiene una prioridad predeterminada de `-2`.

```
UUID=af30cbb0-3866-466a-825a-58889a49ef33  swap  swap  defaults  0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881  swap  swap  pri=4     0 0
UUID=fbd7fa60-b781-44a8-961b-37ac3ef572bf  swap  swap  pri=10    0 0
```

Use el comando `swapon - -show` para visualizar las prioridades de espacio de intercambio (swap).

Cuando los espacios de intercambio (swap) tienen la misma prioridad, el kernel los escribe con el método Round-Robin.



Referencias

Páginas del manual: `mkswap(8)`, `swapon(8)`, `swapoff(8)`, `mount(8)` y `parted(8)`

Base de conocimiento: ¿Cuál es el tamaño de intercambio (swap) recomendado para las plataformas Red Hat?

<https://access.redhat.com/solutions/15244>

► Ejercicio Guiado

Administración de espacio de intercambio (swap)

En este ejercicio, crea y formatea una partición para usar como espacio de intercambio (swap), la formateará como intercambio y la activará de forma persistente.

Resultados

- Crear una partición y un espacio de intercambio (swap) en un disco con el esquema de partición GPT.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start storage-swap
```

Instrucciones

- 1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Inspeccione el disco `/dev/vdb`. El disco ya tiene una tabla de particiones y usa el esquema de partición GPT. Además, tiene una partición existente de 1 GB.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB  1001MB  1000MB                 data
```

- 3. Agregue una nueva partición de 500 MB para usar como espacio de intercambio (swap). Establezca el tipo de partición en `linux-swap`.

- 3.1. Cree la partición `myswap`. Dado que el disco usa el esquema de partición GPT, debe asignarle un nombre a la partición. Observe que la posición inicial, 1001 MB, es el final de la primera partición existente. El comando `parted` se asegura de que la nueva partición siga inmediatamente a la anterior, sin ninguna brecha. Dado que la partición comienza en la posición 1001 MB, el comando establece la posición final en 1501 MB para obtener un tamaño de partición de 500 MB.

```
[root@servera ~]# parted /dev/vdb mkpart myswap linux-swap \
1001MB 1501MB
Information: You may need to update /etc/fstab.
```

- 3.2. Verifique su trabajo enumerando las particiones en el disco `/dev/vdb`. El tamaño de la nueva partición no es exactamente 500 MB. La diferencia de tamaño se debe a que el comando `parted` debe alinear la partición con el diseño del disco.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049KB  1001MB  1000MB                data
  2      1001MB  1501MB  499MB                myswap swap
```

- 3.3. Ejecute el comando `udevadm settle`. Este comando espera a que el sistema registre la nueva partición y regresa cuando finaliza.

```
[root@servera ~]# udevadm settle
```

- ▶ 4. Inicialice la nueva partición como espacio de intercambio (swap).

```
[root@servera ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 476 MiB (499118080 bytes)
no label, UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328
```

- ▶ 5. Habilite el nuevo espacio de intercambio (swap).

- 5.1. Verifique que la creación e inicialización del espacio de intercambio (swap) no lo habilita aún para su uso.

```
[root@servera ~]# swapon --show
```

- 5.2. Habilite el nuevo espacio de intercambio (swap).

```
[root@servera ~]# swapon /dev/vdb2
```

- 5.3. Verifique que el nuevo espacio de intercambio (swap) ahora esté disponible.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRI0
/dev/vdb2 partition 476M  0B   -2
```

5.4. Deshabilite el espacio de intercambio (swap).

```
[root@servera ~]# swapoff /dev/vdb2
```

5.5. Confirme que el espacio de intercambio (swap) está deshabilitado.

```
[root@servera ~]# swapon --show
```

► 6. Habilite el nuevo espacio de intercambio (swap) en el arranque del sistema.

6.1. Use el comando `lsblk` con la opción `-fs` para descubrir el UUID del dispositivo `/dev/vdb2`. El UUID en la salida será diferente en su sistema.

```
[root@servera ~]# lsblk --fs /dev/vdb2
NAME FSTYPE FSVER LABEL UUID                               FSAVAIL FSUSE%
MOUNTPOINTS
vdb2 swap 1          762735cb-a52a-4345-9ed0-e3a68aa8bb97
```

6.2. Agregue una entrada al archivo `/etc/fstab`. En el siguiente comando, reemplace el UUID con el que descubrió en el paso anterior.

```
...output omitted...
UUID=762735cb-a52a-4345-9ed0-e3a68aa8bb97 swap swap defaults 0 0
```

6.3. Actualice el daemon `systemd` para que el sistema registre la nueva configuración del archivo `/etc/fstab`.

```
[root@servera ~]# systemctl daemon-reload
```

6.4. Habilite el espacio de intercambio (swap) usando la entrada en el archivo `/etc/fstab`.

```
[root@servera ~]# swapon -a
```

6.5. Verifique que el nuevo espacio de intercambio (swap) esté habilitado.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRI0
/dev/vdb2 partition 476M  0B   -2
```

► 7. Reinicie la máquina `servera`. Después de reiniciar el servidor, inicie sesión y verifique que el espacio de intercambio (swap) esté habilitado. Cuando finalice, cierre sesión de `servera`.

7.1. Reinicie la máquina `servera`.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

7.2. Espere a que se reinicie `servera` e inicie sesión con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

7.3. Verifique que el espacio de intercambio (swap) esté habilitado.

```
[student@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

7.4. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish storage-swap
```

Esto concluye la sección.

► Trabajo de laboratorio

Administración de almacenamiento básico

En este trabajo de laboratorio, crea varias particiones en un disco nuevo, formateará algunas con sistemas de archivos y las montará, y activará otras como espacios de intercambio (swap).

Resultados

- Visualizar y crear particiones con el comando `parted`.
- Crear sistemas de archivos en particiones y montarlos de forma persistente.
- Crear espacios de intercambio (swap) y activarlos en el arranque.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start storage-review
```

Instrucciones

1. La máquina `serverb` tiene varios discos sin usar. En el primer disco sin usar, cree una partición `backup` GPT de 2 GB. Dado que puede ser difícil establecer el tamaño exacto, un tamaño entre 1,8 GB y 2,2 GB es aceptable. Configure la partición `backup` para alojar un sistema de archivos XFS.
2. Formatee la partición `backup` de 2 GB con un sistema de archivos XFS y móntelo de manera persistente en el directorio `/backup`.
3. En el mismo disco, cree dos particiones GPT de 512 MB denominadas `swap1` y `swap2`. Un tamaño entre 460 MB y 564 MB es aceptable. Configure los tipos de sistemas de archivos de las particiones para alojar espacios de intercambio (swap).
4. Inicialice las dos particiones de 512 MiB como espacios de intercambio (swap) y configúrelas para que se activen en el arranque. Establezca el espacio de intercambio (swap) en la partición `swap2` para que se prefiera sobre el otro.
5. Para verificar su trabajo, reinicie la máquina `serverb`. Confirme que el sistema monta automáticamente la primera partición en el directorio `/backup`. Además, confirme que el sistema activa los dos espacios de intercambio (swap).

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade storage-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish storage-review
```

Esto concluye la sección.

► Solución

Administración de almacenamiento básico

En este trabajo de laboratorio, crea varias particiones en un disco nuevo, formateará algunas con sistemas de archivos y las montará, y activará otras como espacios de intercambio (swap).

Resultados

- Visualizar y crear particiones con el comando `parted`.
- Crear sistemas de archivos en particiones y montarlos de forma persistente.
- Crear espacios de intercambio (swap) y activarlos en el arranque.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start storage-review
```

Instrucciones

1. La máquina `serverb` tiene varios discos sin usar. En el primer disco sin usar, cree una partición `backup` GPT de 2 GB. Dado que puede ser difícil establecer el tamaño exacto, un tamaño entre 1,8 GB y 2,2 GB es aceptable. Configure la partición `backup` para alojar un sistema de archivos XFS.

- 1.1. Inicie sesión en `serverb` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 1.2. Identifique los discos sin usar. El primer disco sin usar, `/dev/vdb`, no tiene particiones.

```
[root@serverb ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
vda   252:0    0  10G  0 disk
├─vda1 252:1    0   1M  0 part
├─vda2 252:2    0 200M  0 part /boot/efi
├─vda3 252:3    0 500M  0 part /boot
└─vda4 252:4    0 9.3G  0 part /
```

```
vdb    252:16    0    5G    0 disk
vdc    252:32    0    5G    0 disk
vdd    252:48    0    5G    0 disk
```

- 1.3. Confirme que el disco `/dev/vdb` no tiene etiqueta.

```
[root@serverb ~]# parted /dev/vdb print
Error: /dev/vdb: unrecognised disk label
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: unknown
Disk Flags:
```

- 1.4. Defina el esquema de partición GPT

```
[root@serverb ~]# parted /dev/vdb mklabel gpt
Information: You may need to update /etc/fstab.
```

- 1.5. Cree una partición `backup` de 2 GB con un tipo de sistema de archivos de `xfs`. Inicie la partición en el sector 2048.

```
[root@serverb ~]# parted /dev/vdb mkpart backup xfs 2048s 2GB
Information: You may need to update /etc/fstab.
```

- 1.6. Confirme la correcta creación de la partición `backup`.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2000MB	1999MB		backup	

- 1.7. Ejecute el comando `udevadm settle`. Este comando espera que el sistema detecte la nueva partición y cree el archivo de dispositivo `/dev/vdb1`.

```
[root@serverb ~]# udevadm settle
```

2. Formatee la partición `backup` de 2 GB con un sistema de archivos XFS y móntelo de manera persistente en el directorio `/backup`.

- 2.1. Formatee la partición `/dev/vdb1`.

```
[root@serverb ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1            isize=512    agcount=4, agsize=121984 blks
=                               sectsz=512   attr=2, projid32bit=1
=                               crc=1       finobt=1, sparse=1, rmapbt=0
=                               reflink=1   bigtime=1 inobtcount=1
```

```
data      =                bsize=4096   blocks=487936, imaxpct=25
          =                sunit=0      swidth=0 blks
naming    =version 2       bsize=4096   ascii-ci=0,  ftype=1
log       =internal log    bsize=4096   blocks=2560, version=2
          =                sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none           extsz=4096   blocks=0,  rtextents=0
```

2.2. Cree el punto de montaje `/backup`.

```
[root@serverb ~]# mkdir /backup
```

2.3. Antes de agregar el nuevo sistema de archivos al archivo `/etc/fstab`, recupere su UUID. El UUID en su sistema podría ser diferente.

```
[root@serverb ~]# lsblk --fs /dev/vdb1
NAME FSTYPE FSVER LABEL UUID                               FSAVAIL FSUSE%
MOUNTPOINTS
vdb1 xfs                f74ed805-b1fc-401a-a5ee-140f97c6757d
```

2.4. Edite el archivo `/etc/fstab` y defina el nuevo sistema de archivos.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=f74ed805-b1fc-401a-a5ee-140f97c6757d /backup xfs defaults 0 0
```

2.5. Fuerce al daemon `systemd` para que vuelva a leer el archivo `/etc/fstab`.

```
[root@serverb ~]# systemctl daemon-reload
```

2.6. Monte manualmente el directorio `/backup` para verificar su trabajo. Confirme que el montaje sea correcto.

```
[root@serverb ~]# mount /backup
[root@serverb ~]# mount | grep /backup
/dev/vdb1 on /backup type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

3. En el mismo disco, cree dos particiones GPT de 512 MB denominadas `swap1` y `swap2`. Un tamaño entre 460 MB y 564 MB es aceptable. Configure los tipos de sistemas de archivos de las particiones para alojar espacios de intercambio (swap).

3.1. Recupere la posición final de la primera partición visualizando la tabla de particiones actual en el disco `/dev/vdb`. En el siguiente paso, use ese valor como el inicio de la partición `swap1`.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2000MB	1999MB	xfs	backup	

- 3.2. Cree la primera partición **swap1** de 512 MB. Establezca su tipo en **linux-swap**. Use la posición final de la primera partición como punto de partida. La posición final es 2000 MB + 512 MB = 2512 MB

```
[root@serverb ~]# parted /dev/vdb mkpart swap1 linux-swap 2000M 2512M
Information: You may need to update /etc/fstab.
```

- 3.3. Cree la segunda partición **swap2** de 512 MB. Establezca su tipo en **linux-swap**. Use la posición final de la partición anterior como punto de partida: 2512M. La posición final es 2512 MB + 512 MB = 3024 MB

```
[root@serverb ~]# parted /dev/vdb mkpart swap2 linux-swap 2512M 3024M
Information: You may need to update /etc/fstab.
```

- 3.4. Visualice la tabla de particiones para verificar su trabajo.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
  1      1049kB  2000MB  1999MB  xfs          backup
  2      2000MB  2512MB  513MB
  3      2512MB  3024MB  512MB  swap1       swap
  3      2512MB  3024MB  512MB  swap2       swap
```

- 3.5. Ejecute el comando **udevadm settle**. El comando espera que el sistema registre las nuevas particiones y cree los archivos de dispositivo.

```
[root@serverb ~]# udevadm settle
```

4. Inicialice las dos particiones de 512 MiB como espacios de intercambio (swap) y configúrelas para que se activen en el arranque. Establezca el espacio de intercambio (swap) en la partición **swap2** para que se prefiera sobre el otro.

- 4.1. Use el comando **mkswap** para inicializar las particiones de intercambio (swap). Observe los UUID de los dos espacios de intercambio (swap), ya que usará esa información en el siguiente paso. Si borra la salida **mkswap**, use el comando **lsblk --fs** para recuperar los UUID.

```
[root@serverb ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 489 MiB (512749568 bytes)
no label, UUID=87976166-4697-47b7-86d1-73a02f0fc803
[root@serverb ~]# mkswap /dev/vdb3
Setting up swapspace version 1, size = 488 MiB (511700992 bytes)
no label, UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942
```

- 4.2. Edite el archivo `/etc/fstab` y defina los nuevos espacios de intercambio (swap). Para configurar el espacio de intercambio (swap) en la partición `swap2` para que se prefiera sobre la partición `swap1`, otórguele a la partición `swap2` una mayor prioridad con la opción `pri`.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4bfb-49b6-a528-74e268b058dd /backup xfs defaults 0 0
UUID=87976166-4697-47b7-86d1-73a02f0fc803 swap swap pri=10 0 0
UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942 swap swap pri=20 0 0
```

- 4.3. Fuerce al daemon `systemd` para que vuelva a leer el archivo `/etc/fstab`.

```
[root@serverb ~]# systemctl daemon-reload
```

- 4.4. Active los nuevos espacios de intercambio (swap). Verifique la correcta activación de los espacios de intercambio (swap).

```
[root@serverb ~]# swapon -a
[root@serverb ~]# swapon --show
NAME      TYPE      SIZE USED  PRIO
/dev/vdb2 partition 489M  0B    10
/dev/vdb3 partition 488M  0B    20
```

5. Para verificar su trabajo, reinicie la máquina `serverb`. Confirme que el sistema monta automáticamente la primera partición en el directorio `/backup`. Además, confirme que el sistema activa los dos espacios de intercambio (swap).

- 5.1. Reinicie `serverb`.

```
[root@serverb ~]# systemctl reboot
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

- 5.2. Espere a que se reinicie `serverb` y, luego, inicie sesión con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 5.3. Verifique que el sistema monta automáticamente la partición `/dev/vdb1` en el directorio `/backup`.

```
[student@serverb ~]$ mount | grep /backup
/dev/vdb1 on /backup type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

- 5.4. Verifique que el sistema active ambos espacios de intercambio (swap).

```
[student@serverb ~]$ swapon --show
NAME      TYPE      SIZE USED  PRIO
/dev/vdb2 partition 489M   0B    10
/dev/vdb3 partition 488M   0B    20
```

5.5. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade storage-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish storage-review
```

Esto concluye la sección.

Resumen

- El comando `parted` agrega, modifica y elimina particiones en discos con esquemas de partición MBR o GPT.
- El comando `mkfs.xfs` crea sistemas de archivos XFS en particiones de disco.
- El archivo `/etc/fstab` contiene dispositivos que deben montarse de forma persistente.
- El comando `mkswap` inicializa espacios de intercambio (swap).

Administración de la pila (stack) de almacenamiento

Meta

Crear y administrar volúmenes lógicos que contengan sistemas de archivos o espacios de intercambio (swap) desde la línea de comandos.

Objetivos

- Describir los componentes y conceptos del administrador de volúmenes lógicos e implementar el almacenamiento de LVM y mostrar la información de los componentes de LVM.
- Analizar los múltiples componentes de almacenamiento que conforman las capas de la pila (stack) de almacenamiento.

Secciones

- Creación y ampliación de volúmenes lógicos (y ejercicio guiado)
- Administración de almacenamiento en capas (y ejercicio guiado)

Trabajo de laboratorio

Administración de la pila (stack) de almacenamiento

Creación y ampliación de volúmenes lógicos

Objetivos

Describir los componentes y conceptos del administrador de volúmenes lógicos e implementar el almacenamiento de LVM y mostrar la información de los componentes de LVM.

Descripción general del administrador de volúmenes lógicos

Use el sistema *administrador de volúmenes lógicos (LVM)* para crear volúmenes de almacenamiento lógico como una capa en el almacenamiento físico. Este sistema de almacenamiento proporciona una mayor flexibilidad que el uso de almacenamiento físico directamente. LVM oculta la configuración de almacenamiento de hardware del software y le permite cambiar el tamaño de los volúmenes sin detener las aplicaciones ni desmontar los sistemas de archivos. LVM proporciona herramientas completas de línea de comandos para administrar el almacenamiento.

Dispositivos físicos

Los volúmenes lógicos usan dispositivos físicos para almacenar datos. Estos dispositivos podrían ser particiones de discos, discos enteros, arreglos RAID o discos SAN. Debe inicializar el dispositivo como volumen físico LVM. Un volumen físico de LVM debe usar todo el dispositivo físico.

Volúmenes físicos (PV)

LVM usa el dispositivo físico subyacente como volumen físico de LVM. Las herramientas de LVM segmentan volúmenes físicos en *extensiones físicas (PE)* para formar pequeños conjuntos de datos que actúan como el bloque de almacenamiento más pequeño en un PV.

Grupos de volúmenes (VG)

Los grupos de volúmenes son conjuntos (pools) de almacenamiento conformados por uno o más PV. Se trata del equivalente funcional de un disco completo en comparación con el almacenamiento físico. Un PV solo puede ser asignado a un único VG. LVM establece el tamaño de la PE automáticamente, aunque es posible especificarlo. Un VG puede constar de espacio sin usar y de varios volúmenes lógicos.

Volúmenes lógicos (LV)

Los volúmenes lógicos se crean desde extensiones físicas libres en un VG y se proporcionan como el dispositivo de almacenamiento para aplicaciones, usuarios y sistemas operativos. Los LV son una colección de *extensiones lógicas (LE)*, que se asignan a extensiones físicas. De forma predeterminada, cada LE se asigna a una PE. La configuración de opciones de LV específicas cambia esta asignación; por ejemplo, la duplicación hace que cada LE se asigne a dos PE.

Flujo de trabajo del administrador de volúmenes lógicos

La creación del almacenamiento de LVM requiere la creación de estructuras en un flujo de trabajo lógico.

- Determine los dispositivos físicos usados para crear volúmenes físicos e inicialice estos dispositivos como volúmenes físicos de LVM.
- Cree un grupo de volúmenes a partir de varios volúmenes físicos.

- Cree los volúmenes lógicos del espacio disponible en el grupo de volúmenes.
- Formatee el volumen lógico con un sistema de archivos y móntelo o actívelo como espacio de intercambio (swap), o pase el volumen sin formato a una base de datos o servidor de almacenamiento para estructuras avanzadas.

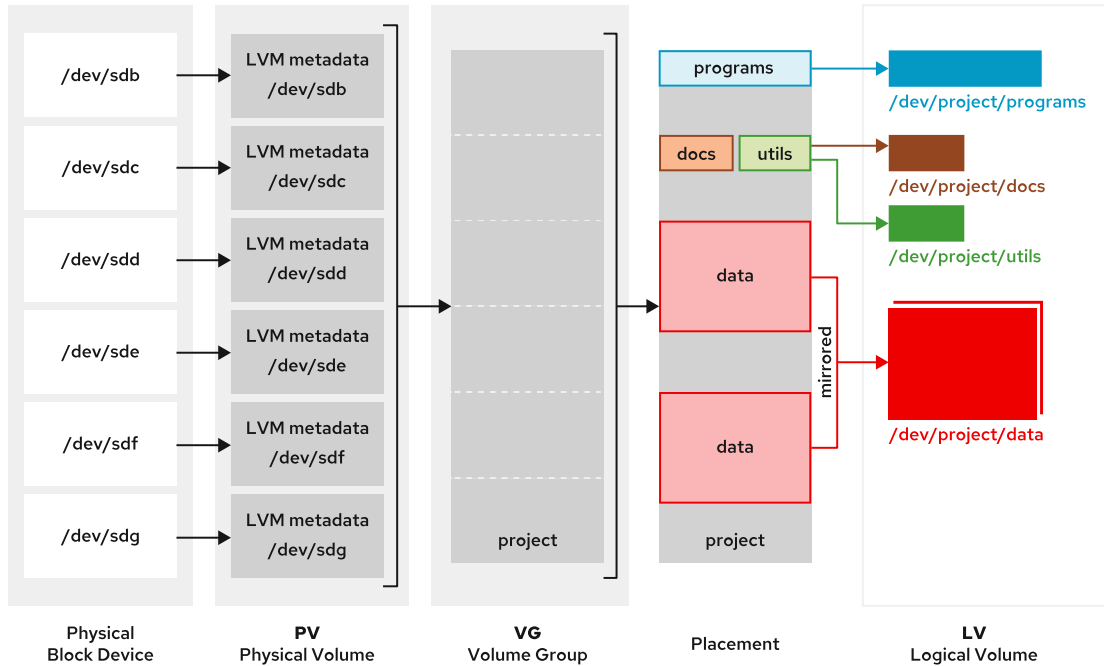


Figura 6.1: Flujo de trabajo del administrador de volúmenes lógicos



nota

Los ejemplos aquí usan un nombre de dispositivo `/dev/vdb` y sus particiones de almacenamiento. Los nombres de los dispositivos en el sistema en el aula pueden ser diferentes. Use los comandos `lsblk`, `blkid` o `cat /proc/partitions` para identificar los dispositivos en su sistema.

Creación del almacenamiento de LVM

La creación de un volumen lógico implica la creación de particiones de dispositivos físicos, volúmenes físicos y grupos de volúmenes. Después de crear un LV, formatee el volumen y móntelo para acceder a él como almacenamiento.

Preparación de dispositivos físicos

La partición es opcional cuando ya está presente. Use el comando `parted` para crear una nueva partición en el dispositivo físico. Establezca el dispositivo físico en el tipo de partición Linux LVM. Use el comando `udevadm settle` para registrar la nueva partición con el kernel.

```
[root@host ~]# parted /dev/vdb mklabel gpt mkpart primary 1MiB 769MiB
...output omitted...
[root@host ~]# parted /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted /dev/vdb set 1 lvm on
[root@host ~]# parted /dev/vdb set 2 lvm on
[root@host ~]# udevadm settle
```

Creación de volúmenes físicos

Use el comando `pvcreate` para etiquetar la partición física como volumen físico de LVM. Etiquete múltiples dispositivos al mismo tiempo al usar nombres de dispositivos delimitados por espacios como argumentos para el comando `pvcreate`. Este ejemplo etiqueta los dispositivos `/dev/vdb1` y `/dev/vdb2` como PV que están listos para crear grupos de volúmenes.

```
[root@host ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
Creating devices file /etc/lvm/devices/system.devices
```

Creación de un grupo de volúmenes

El comando `vgcreate` crea uno o más volúmenes físicos en un grupo de volúmenes. El primer comando es un nombre de grupo de volúmenes seguido de uno o más volúmenes físicos para asignar a este VG. Este ejemplo crea el VG `vg01` usando los PV `/dev/vdb1` y `/dev/vdb2`.

```
[root@host ~]# vgcreate vg01 /dev/vdb1 /dev/vdb2
Volume group "vg01" successfully created
```

Creación de un volumen lógico

El comando `lvcreate` crea un nuevo volumen lógico desde las PE disponibles en un grupo de volúmenes. Use el comando `lvcreate` para establecer el nombre y el tamaño del LV y el nombre del VG que contendrá este volumen lógico. Este ejemplo crea el LV `lv01` con un tamaño de 700 MiB en el VG `vg01`.

```
[root@host ~]# lvcreate -n lv01 -L 300M vg01
Logical volume "lv01" created.
```

Este comando podría fallar si el grupo de volúmenes no tiene suficientes extensiones físicas libres. El tamaño de LV se redondea al siguiente valor de tamaño de PE cuando el tamaño no coincide exactamente.

La opción `-L` del comando `lvcreate` requiere tamaños en bytes, mebibytes (megabytes binarios, 1048576 bytes), gibibytes (gigabytes binarios) o similar. La `-l` minúscula requiere tamaños especificados como una cantidad de extensiones físicas. Los siguientes comandos son dos opciones para crear el mismo LV con el mismo tamaño:

- `lvcreate -n lv01 -L 128M vg01` : crea un LV de 128 MiB, redondeado al siguiente PE.
- `lvcreate -n lv01 -l 32 vg01` : crea un LV de 32 PE a 4 MiB cada uno es 128 MiB.

Creación de un volumen lógico con deduplicación y compresión

RHEL 9 usa implementación de VDO de LVM para administrar volúmenes de VDO. Las herramientas de administración de VDO basadas en Python anteriores aún están disponibles, pero ya no son necesarias.

El *optimizador de datos virtual (VDO)* proporciona deduplicación de nivel de bloque en línea, compresión y aprovisionamiento ligero para el almacenamiento. Configure un volumen de VDO para usar hasta 256 TB de almacenamiento físico. Administrar VDO como un tipo de volumen

lógico (LV) de LVM, similar a los volúmenes con aprovisionamiento ligero de LVM. Un VDO de LVM está compuesto por dos volúmenes lógicos:

LV del conjunto (pool) de VDO

Este LV almacena, deduplica y comprime datos y establece el tamaño del volumen de VDO respaldado por el dispositivo físico. VDO se deduplica y comprime cada LV de VDO por separado porque cada LV de conjunto (pool) de VDO puede contener solo un LV de VDO.

LV de VDO

Un dispositivo virtual se aprovisiona sobre el LV del conjunto (pool) de VDO y establece el tamaño lógico del volumen de VDO que almacena los datos antes de que se produzca la deduplicación y la compresión.

VDO de LVM presenta el almacenamiento deduplicado como un volumen lógico regular (LV). El volumen de VDO puede formatearse con sistemas de archivos estándares, compartirse como un dispositivo de bloque o usarse para crear otras capas de almacenamiento, al igual que cualquier volumen lógico normal.

Para poder usar la deduplicación y compresión de VDO, instale los paquetes `vdo` y `kmod-kvdo`.

```
[root@host ~]# dnf install vdo kmod-kvdo
```

Verifique que el grupo de volúmenes de LVM seleccionado tenga suficiente capacidad de almacenamiento libre. Use el comando `lvcreate` con el parámetro `--type vdo` para crear un LV de VDO.

```
[root@host ~]# lvcreate --type vdo --name vdo-lv01 --size 5G vg01
Logical blocks defaulted to 523108 blocks.
The VDO volume can address 2 GB in 1 data slab.
It can grow to address at most 16 TB of physical storage in 8192 slabs.
If a larger maximum size might be needed, use bigger slabs.
Logical volume "vdo-lv01" created.
```

Creación de un sistema de archivos en volúmenes lógicos

Especifique el volumen lógico, ya sea usando el nombre tradicional `/dev/vgname/lvname` o el nombre del asignador de dispositivos del kernel `/dev/mapper/_vgname-lvname_`.

Use el comando `mkfs` para crear un sistema de archivos en el nuevo volumen lógico.

```
[root@host ~]# mkfs -t xfs /dev/vg01/lv01
...output omitted...
```

Cree un punto de montaje con el comando `mkdir`.

```
[root@host ~]# mkdir /mnt/data
```

Para que el sistema de archivos esté disponible de manera persistente, agregue una entrada al archivo `/etc/fstab`.

```
/dev/vg01/lv01 /mnt/data xfs defaults 0 0
```

Monte el LV con el comando `mount`.

```
[root@host ~]# mount /mnt/data/
```

**nota**

Puede montar un volumen lógico por nombre o por UUID, ya que LVM analiza los PV en busca del UUID. Este comportamiento se realizó correctamente incluso cuando el VG se creó con un nombre, ya que el PV siempre contendrá un UUID.

Visualización del estado de los componentes de LVM

LVM proporciona varias utilidades para mostrar la información de estado de PV, VG y LV. Use los comandos `pvdisplay`, `vgdisplay` y `lvdisplay` para mostrar la información de estado de los componentes de LVM.

Los comandos `pvs`, `vgs` y `lvs` asociados se usan comúnmente y muestran un subconjunto de la información de estado, con una línea para cada entidad.

Visualización de la información de volumen físico

El comando `pvdisplay` muestra esta información acerca de PV. Use el comando sin argumentos para enumerar la información de todos los PV presentes en el sistema. Al proporcionar el nombre del PV como argumento para el comando, se muestra la información específica del PV.

```
[root@host ~]# pvdisplay /dev/vdb1
--- Physical volume ---
PV Name                /dev/vdb1
VG Name                vg01
PV Size                731.98 MiB / not usable 3.98 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               182
Free PE                107
Allocated PE           75
PV UUID                zP0gD9-NxTV-Qtoi-yfQD-TGpL-0Yj0-wExh2N
```

- ❶ PV Name muestra el nombre del dispositivo.
- ❷ VG Name muestra el grupo de volúmenes donde se encuentra asignado el PV.
- ❸ PV Size muestra el tamaño físico del PV, incluido todo el espacio no usable.
- ❹ PE Size muestra el tamaño de la extensión física.
- ❺ Free PE muestra el tamaño de PE disponible en el VG para crear nuevos LV o ampliar los LV existentes.

Visualización de la información de grupos de volúmenes

El comando `vgdisplay` muestra la información acerca de los grupos de volúmenes. Para mostrar información sobre todos los VG, use el comando sin argumentos. Proporcione el nombre del VG como argumento para enumerar la información específica del VG.


```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name                vg01
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   2
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
Open LV                1
Max PV                 0
Cur PV                2
Act PV                 2
VG Size                1012.00 MiB
PE Size                4.00 MiB
Total PE                253
Alloc PE / Size        75 / 300.00 MiB
Free PE / Size         178 / 712.00 MiB
VG UUID                jK5M1M-Yv1k-kxU2-bxMS-dNjQ-Bs3L-DRLJNc
```

- ❶ VG Name muestra el nombre del grupo de volúmenes.
- ❷ VG Size es el tamaño total del conjunto (pool) de almacenamiento disponible para la asignación de LV.
- ❸ Total PE muestra el tamaño total de unidades de PE.
- ❹ Free PE / Size muestra el espacio disponible en el VG para crear nuevos LV o ampliar los LV existentes.

Visualización de la información de volumen lógico

El comando `lvdisplay` muestra esta información acerca de volúmenes lógicos. Use el comando sin argumentos para enumerar la información de todos los LV. Proporcionar el nombre del LV como argumento muestra información específica del LV.

```
[root@host ~]# lvdisplay /dev/vg01/lv01
--- Logical volume ---
LV Path                /dev/vg01/lv01
LV Name                lv01
VG Name                vg01
LV UUID                FVmNel-u25R-dt3p-C5L6-VP2w-QRNP-scqrbq
LV Write Access        read/write
LV Creation host, time servera.lab.example.com, 2022-04-07 10:45:34 -0400
LV Status               available
# open                 1
LV Size                300.00 MiB
Current LE              75
Segments               1
Allocation              inherit
```

Read ahead sectors	auto
- currently set to	8192
Block device	253:0

- ❶ LV Path muestra el nombre del dispositivo del LV.
- ❷ VG Name muestra el VG usado para crear este LV.
- ❸ LV Size muestra el tamaño total del LV. Use herramientas del sistema de archivos para determinar el espacio libre y usado para el LV.
- ❹ Current LE muestra el número de extensiones lógicas usadas por este LV.

Amplíe y reduzca el almacenamiento de LVM

Después de crear un volumen lógico, puede ampliar el volumen para ampliar el sistema de archivos. Es posible que necesite ampliar el PV o VG para aumentar la capacidad de almacenamiento del LV.

Ampliar el tamaño de un grupo de volúmenes

Es posible que deba agregar más espacio en disco para ampliar un VG. Puede agregar volúmenes físicos adicionales a un VG para ampliar su tamaño disponible.

Prepare el dispositivo físico y cree el volumen físico cuando no esté presente.

```
[root@host ~]# parted /dev/vdb mkpart primary 1072MiB 1648MiB
...output omitted...
[root@host ~]# parted /dev/vdb set 3 lvm on
...output omitted...
[root@host ~]# udevadm settle
[root@host ~]# pvcreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

El comando `vgextend` agrega el nuevo PV al VG. Proporcione los nombres del VG y del PV como argumentos para el comando `vgextend`.

```
[root@host ~]# vgextend vg01 /dev/vdb3
Volume group "vg01" successfully extended
```

Este comando amplía el VG `vg01` al tamaño del PV `/dev/vdb3`.

Ampliación del volumen lógico

Un beneficio de usar volúmenes lógicos es aumentar su tamaño sin experimentar tiempo de inactividad. Agregue extensiones físicas libres al LV en el VG para ampliar su capacidad para ampliar el sistema de archivos del LV. Use el comando `vgdisplay` para confirmar que el grupo de volúmenes tenga suficiente espacio libre para la extensión del LV.

Use el comando `lvextend` para ampliar el LV.

```
[root@host ~]# lvextend -L +500M /dev/vg01/lv01
Size of logical volume vg01/lv01 changed from 300.00 MiB (75 extents) to 800.00
MiB (200 extents).
Logical volume vg01/lv01 successfully resized.
```

Esto aumenta el tamaño del volumen lógico lv01 en 500 MiB. El signo más (+) delante del tamaño significa que desea agregar este valor al tamaño existente; de lo contrario, sin el signo más, el valor define el tamaño final del LV.

El comando `lvextend` usa diferentes métodos para especificar el tamaño. La opción `-l` del comando `lvextend` espera el número de PE como argumento. La opción `-L` del comando `lvextend` espera tamaños en bytes, mebibytes, gibibytes y similares.

Ampliación de un sistema de archivos XFS al tamaño de volumen lógico

El comando `xfs_growfs` ayuda a ampliar el sistema de archivos para que ocupe el LV ampliado. El sistema de archivos de destino se debe montar antes de usar el comando `xfs_growfs`. Puede seguir usando el sistema de archivos mientras se cambia el tamaño.

```
[root@host ~]# xfs_growfs /mnt/data/
...output omitted...
data blocks changed from 76800 to 204800
```



Importante

Ejecute siempre el comando `xfs_growfs` después de ejecutar el comando `lvextend`. Use la opción `-r` del comando `lvextend` para ejecutar los dos pasos consecutivamente. Después de ampliar el LV, cambia el tamaño del sistema de archivos con el comando `fsadm`. Esta opción soporta varios otros sistemas de archivos.

Ampliación de un sistema de archivos EXT4 al tamaño de volumen lógico

Los pasos para ampliar el LV con el sistema de archivos `ext4` son los mismos para LV con el sistema de archivos XFS, excepto por el paso para cambiar el tamaño del sistema de archivos.

El comando `resize2fs` amplía el sistema de archivos para ocupar el nuevo LV ampliado. Puede seguir usando el sistema de archivos mientras se cambia el tamaño.

```
[root@host ~]# resize2fs /dev/vg01/lv01
resize2fs 1.46.5 (30-Dec-2021)
Resizing the filesystem on /dev/vg01/lv01 to 256000 (4k) blocks.
The filesystem on /dev/vg01/lv01 is now 256000 (4k) blocks long.
```

La principal diferencia entre `xfs_growfs` y `resize2fs` es el argumento que se pasó para identificar el sistema de archivos. El comando `xfs_growfs` toma el punto de montaje como argumento y el comando `resize2fs` toma el nombre del LV como argumento. El comando `xfs_growfs` solo soporta un cambio de tamaño en línea, mientras que el comando `resize2fs` soporta el cambio de tamaño en línea y sin conexión. Puede cambiar el tamaño de un sistema de

archivos ext4 hacia arriba o hacia abajo, pero solo puede cambiar el tamaño de un sistema de archivos XFS.

Ampliación de volúmenes lógicos del espacio de intercambio (swap)

Puede ampliar los LV usados como espacio de intercambio (swap); sin embargo, el proceso difiere de ampliar el sistema de archivos ext4 o XFS. Los volúmenes lógicos usados como espacio de intercambio (swap) se deben desconectar para ampliarse.

Use el comando `swapoff` para desactivar el espacio de intercambio (swap) en el LV.

```
[root@host ~]# swapoff -v /dev/vg01/swap
swapoff /dev/vg01/swap
```

Use el comando `lvextend` para ampliar el LV.

```
[root@host ~]# lvextend -L +300M /dev/vg01/swap
Size of logical volume vg01/swap changed from 500.00 MiB (125 extents) to 800.00
MiB (200 extents).
Logical volume vg01/swap successfully resized.
```

Use el comando `mkswap` para formatear el LV como espacio de intercambio (swap).

```
[root@host ~]# mkswap /dev/vg01/swap
mkswap: /dev/vg01/swap: warning: wiping old swap signature.
Setting up swapspace version 1, size = 800 MiB (838856704 bytes)
no label, UUID=25b4d602-6180-4b1c-974e-7f40634ad660
```

Use el comando `swapon` para activar el espacio de intercambio (swap) en el LV.

```
[root@host ~]# swapon /dev/vg01/swap
```

Reducción del almacenamiento del grupo de volúmenes

Reducir un VG implica eliminar el PV no usado del VG. El comando `pvmove` mueve datos de extensiones en un PV a extensiones en otro PV con suficientes extensiones libres en el mismo VG. Puede continuar usando el LV desde el mismo VG mientras reduce. Use la opción `-A` del comando `pvmove` para hacer una copia de seguridad automática de los metadatos del VG después de un cambio. Esta opción usa el comando `vgcfgbackup` para hacer una copia de seguridad de los metadatos automáticamente.

```
[root@host ~]# pvmove /dev/vdb3
```



Advertencia

Antes de usar el comando `pvmove`, realice una copia de seguridad de los datos almacenados en todos los LV en el VG. Una pérdida de energía inesperada durante la operación puede dejar al VG en un estado inconsistente que podría causar una pérdida de datos en los LV.

Use el comando `vgreduce` para eliminar un PV de un VG.

```
[root@host ~]# vgreduce vg01 /dev/vdb3
Removed "/dev/vdb3" from volume group "vg01"
```



Importante

Los sistemas de archivos GFS2 y XFS no soportan la reducción, por lo que no puede reducir el tamaño de un LV.

Eliminar almacenamiento de LVM

Use los comandos `lvremove`, `vgremove` y `pvremove` para eliminar un componente de LVM que ya no es necesario.

Preparación del sistema de archivos

Traslade todos los datos que se deben conservar a otro sistema de archivos. Use el comando `umount` para desmontar el sistema de archivos y, luego, eliminar cualquier entrada de `/etc/fstab` asociada a este sistema de archivos.

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



Advertencia

Al eliminar un volumen lógico, se destruyen todos los datos almacenados en este. Realice una copia de seguridad de los datos o trasládelos **antes** de eliminar el volumen lógico.

Eliminación del volumen lógico

Use el comando `lvremove` *DEVICE-NAME* para eliminar un volumen lógico que ya no es necesario. Desmonte el sistema de archivos del LV antes de ejecutar este comando. El comando solicita confirmación antes de eliminar el LV.

```
[root@host ~]# lvremove /dev/vg01/lv01
Do you really want to remove active logical volume vg01/lv01? [y/n]: y
Logical volume "lv01" successfully removed.
```

Las extensiones físicas del LV se liberan y están disponibles para ser asignadas a LV existentes o nuevos en el grupo de volúmenes.

Eliminación del grupo de volúmenes

Use el comando `vgremove` *VG-NAME* para eliminar un grupo de volúmenes que ya no es necesario.

```
[root@host ~]# vgremove vg01
Volume group "vg01" successfully removed
```

Los volúmenes físicos del VG se liberan y están disponibles para ser asignados a VG existentes o nuevos en el sistema.

Eliminación de volúmenes lógicos

Use el comando `pvremove` para eliminar volúmenes físicos que ya no son necesarios. Use una lista delimitada por espacios de dispositivos del PV para eliminar más de uno a la vez. Este comando elimina los metadatos del PV de la partición (o disco). Ahora, la partición está libre para una nueva asignación o para ser formateada.

```
[root@host ~]# pvremove /dev/vdb1 /dev/vdb2
Labels on physical volume "/dev/vdb1" successfully wiped.
Labels on physical volume "/dev/vdb2" successfully wiped.
```



Referencias

Páginas del manual: `fdisk(8)`, `gdisk(8)`, `parted(8)`, `partprobe(8)`, `lvm(8)`, `pvcreate(8)`, `vgcreate(8)`, `lvcreate(8)`, `mkfs(8)`, `pvdisplay(8)`, `vgdisplay(8)`, `lvdisplay(8)`, `vgextend(8)`, `lvextend(8)`, `xfs_growfs(8)`, `resize2fs(8)`, `swapoff(8)`, `mkswap(8)`, `swapon(8)`, `pvmove(8)`, `vgcfgbackup(8)`, `vgreduce(8)`, `lvremove(8)`, `vgremove(8)`, `pvremove(8)`

Para obtener más información, consulte *Configuring and Managing Logical Volumes* en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/configuring_and_managing_logical_volumes/index

► Ejercicio Guiado

Creación y ampliación de volúmenes lógicos

En este trabajo de laboratorio, crea y amplía un volumen físico, un grupo de volúmenes, un volumen lógico y un sistema de archivos XFS. También monta de forma persistente el sistema de archivos de volúmenes lógicos.

Resultados

- Crear volúmenes físicos, grupos de volúmenes y volúmenes lógicos con herramientas de LVM.
- Crear nuevos sistemas de archivos en volúmenes lógicos y montarlos de manera persistente.
- Ampliar el grupo de volúmenes para incluir un volumen físico adicional.
- Cambiar el tamaño del volumen lógico mientras el sistema de archivos aún está montado y en uso.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start lvm-manage
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Cree la partición del dispositivo físico en el dispositivo de almacenamiento `/dev/vdb`.

- 2.1. Cree dos particiones de 256 MiB cada una y configúrelas con el tipo Linux LVM. Use `first` y `second` como nombres para estas particiones.

```
[root@servera ~]# parted /dev/vdb mklabel gpt
[root@servera ~]# parted /dev/vdb mkpart first 1MiB 258MiB
[root@servera ~]# parted /dev/vdb set 1 lvm on
[root@servera ~]# parted /dev/vdb mkpart second 258MiB 514MiB
[root@servera ~]# parted /dev/vdb set 2 lvm on
```

- 2.2. Registre las nuevas particiones con el kernel.

```
[root@servera ~]# udevadm settle
```

- 2.3. Enumere las particiones en el dispositivo de almacenamiento `/dev/vdb`. En la columna `Number`, los valores 1 y 2 corresponden a las particiones del dispositivo `/dev/vdb1` y `/dev/vdb2`. La columna `Flags` indica el tipo de partición.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	271MB	269MB		first	lvm
2	271MB	539MB	268MB		second	lvm

- ▶ 3. Etiquete las dos nuevas particiones como volúmenes físicos.

```
[root@servera ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
Creating devices file /etc/lvm/devices/system.devices
```

- ▶ 4. Cree el grupo de volúmenes `servera_group` con los dos nuevos PV.

```
[root@servera ~]# vgcreate servera_group /dev/vdb1 /dev/vdb2
Volume group "servera_group" successfully created
```

- ▶ 5. Cree el volumen lógico `servera_volume` con un tamaño de 400 MiB. Este comando crea el LV `/dev/servera_group/servera_volume` sin un sistema de archivos.

```
[root@servera ~]# lvcreate -n servera_volume -L 400M servera_group
Logical volume "servera_volume" created.
```

- ▶ 6. Formatee el LV recién creado y móntelo de forma persistente.

- 6.1. Formatee el nuevo LV `servera_volume` con el sistema de archivos XFS.

```
[root@servera ~]# mkfs -t xfs /dev/servera_group/servera_volume
...output omitted...
```

- 6.2. Cree el directorio `/data` como punto de montaje.

```
[root@servera ~]# mkdir /data
```

- 6.3. Para montar de manera persistente el sistema de archivos creado recientemente, agregue el siguiente contenido en el archivo `/etc/fstab`.


```
/dev/servera_group/servera_volume /data xfs defaults 0 0
```

6.4. Monte el LV `servera_volume`.

```
[root@servera ~]# mount /data
```

- 7. Verifique que se pueda acceder al sistema de archivos montado y visualice la información de estado del LVM.

7.1. Verifique que pueda copiar archivos en el directorio `/data`.

```
[root@servera ~]# cp -a /etc/*.conf /data
[root@servera ~]# ls /data | wc -l
32
```

7.2. Vea la información de estado de PV. La salida muestra que el PV usa el VG `servera_group`. El PV tiene un tamaño de 256 MiB y el tamaño de la extensión física de 4 MiB.

Hay 63 PE, con 27 PE disponibles para la asignación y 36 PE actualmente asignadas al LV. Al calcular el tamaño en MiB:

- Total de 252 MiB (63 PE x 4 MiB).
- 108 MiB libres (27 PE x 4 MiB)
- 144 MiB asignados (36 PE x 4 MiB)

```
[root@servera ~]# pvdiskdisplay /dev/vdb2
--- Physical volume ---
PV Name                /dev/vdb2
VG Name                servera_group
PV Size                256.00 MiB / not usable 4.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               63
Free PE                27
Allocated PE           36
PV UUID                FKKFYJ-wJiR-1jt2-sfy3-yjPy-Tyln-LG92jj
```

7.3. Vea la información de estado de PV del VG `servera_group`. La salida muestra el VG de 508 MiB con el PE de 4 MiB. El tamaño disponible del VG es 108 MiB.

```
[root@servera ~]# vgdisplay servera_group
--- Volume group ---
VG Name                servera_group
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   2
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
```

```

Open LV          1
Max PV           0
Cur PV          2
Act PV           2
VG Size          508.00 MiB
PE Size          4.00 MiB
Total PE         127
Alloc PE / Size  100 / 400.00 MiB
Free PE / Size   27 / 108.00 MiB
VG UUID          g0ahyT-90J5-iGic-nnb5-G6T9-tLdK-dX8c9M

```

- 7.4. Vea la información de estado del LV `servera_volume`. La salida muestra el nombre de VG usado para crear el LV. También muestra el LV de 400 MiB y el LE de 100.

```

[root@servera ~]# lvdisplay /dev/servera_group/servera_volume
--- Logical volume ---
LV Path                /dev/servera_group/servera_volume
LV Name                servera_volume
VG Name                servera_group
LV UUID                93MfUt-esgT-B5HM-r1p5-DVZH-n5cn-J5e2tw
LV Write Access        read/write
LV Creation host, time servera.lab.example.com, 2022-04-11 03:25:12 -0400
LV Status              available
# open                 1
LV Size                400.00 MiB
Current LE             100
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     8192
Block device           253:0

```

- 7.5. Vea el espacio libre en disco en unidades legibles por humanos. La salida muestra el tamaño total de 395 MiB con el tamaño disponible de 372 MiB.

```

[root@servera ~]# df -h /data

```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/servera_group-servera_volume	395M	24M	372M	6%	/data

- 8. Cree los recursos físicos en el dispositivo de almacenamiento `/dev/vdb`.

- 8.1. Cree una partición adicional de 512 MiB y configúrela con el tipo Linux LVM. Use `third` como el nombre para esta partición.

```

[root@servera ~]# parted /dev/vdb mkpart third 514MiB 1026MiB
[root@servera ~]# parted /dev/vdb set 3 lvm on

```

- 8.2. Registre la nueva partición con el kernel.

```

[root@servera ~]# udevadm settle

```

- 8.3. Agregue la nueva partición como un PV.

```
[root@servera ~]# pvccreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

- 9. Con el espacio en disco creado recientemente, amplíe el sistema de archivos en `servera_volume` para que tenga un tamaño total de 700 MiB.

9.1. Extienda el VG `servera_group` con el nuevo PV `/dev/vdb3`.

```
[root@servera ~]# vgextend servera_group /dev/vdb3
Volume group "servera_group" successfully extended
```

9.2. Amplíe el LV `servera_volume` existente a 700 MiB.

```
[root@servera ~]# lvextend -L 700M /dev/servera_group/servera_volume
Size of logical volume servera_group/servera_volume changed from 400.00 MiB (100
extents) to 700.00 MiB (175 extents).
Logical volume servera_group/servera_volume successfully resized.
```

9.3. Amplíe el sistema de archivos XFS con espacio libre del LV.

```
[root@servera ~]# xfs_growfs /data
...output omitted...
data blocks changed from 102400 to 179200
```

- 10. Verifique que el tamaño del LV se haya ampliado y que el contenido aún esté presente en el volumen.

10.1. Verifique el tamaño del LV extendido con el comando `lvdisplay`.

```
[root@servera ~]# lvdisplay /dev/servera_group/servera_volume
--- Logical volume ---
LV Path                /dev/servera_group/servera_volume
LV Name                 servera_volume
VG Name                 servera_group
LV UUID                 mLQhsD-hyL0-KC2B-2nug-o2Nc-0znS-Q428fK
LV Write Access         read/write
LV Creation host, time  servera.lab.example.com, 2022-04-12 06:04:12 -0400
LV Status                available
# open                  1
LV Size                 700.00 MiB
Current LE              175
Segments                3
Allocation               inherit
Read ahead sectors      auto
- currently set to     8192
Block device            253:0
```

10.2. Verifique el tamaño del nuevo sistema de archivos. Verifique que los archivos copiados anteriormente aún estén presentes.

```
[root@servera ~]# df -h /data
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/servera_group-servera_volume 695M   26M  670M   4% /data
[root@servera ~]# ls /data | wc -l
32
```

- 11. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish lvm-manage
```

Esto concluye la sección.

Administración de almacenamiento en capas

Objetivos

Analizar los múltiples componentes de almacenamiento que conforman las capas de la pila (stack) de almacenamiento.

Pila (stack) de almacenamiento

El almacenamiento en RHEL está compuesto por varias capas de controladores, administradores y utilidades que son maduras, estables y están llenas de características modernas. La administración del almacenamiento requiere estar familiarizado con los componentes de la pila (stack) y reconocer que la configuración del almacenamiento afecta el proceso de arranque, el rendimiento de la aplicación y la capacidad de proporcionar las funciones de almacenamiento necesarias para casos de uso específicos de la aplicación.

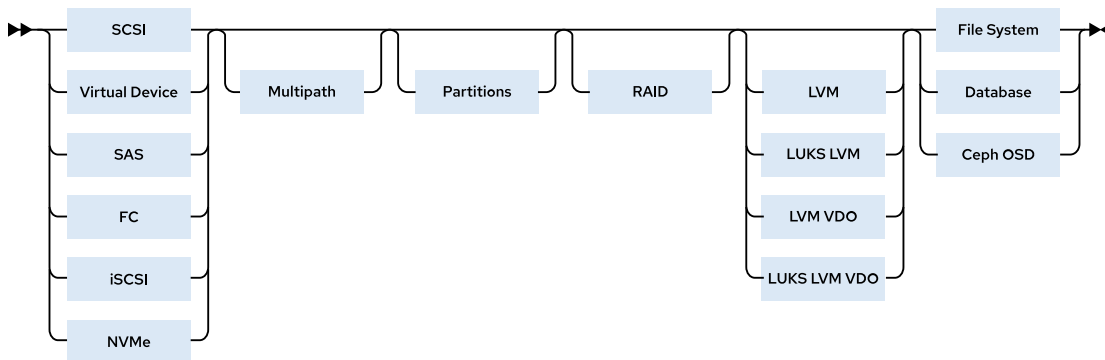


Figura 6.2: Pila (stack) de almacenamiento

Las secciones anteriores de los cursos Red Hat System Administration han presentado los sistemas de archivos XFS, el uso compartido de almacenamiento de red, la partición y el administrador de volúmenes lógicos. En esta sección, se presenta una descripción general de la pila (stack) de almacenamiento de RHEL de abajo hacia arriba y se presenta cada capa.

Esta sección también abarca Stratis, el daemon que unifica, configura y monitorea los componentes de la pila (stack) de almacenamiento RHEL subyacentes, y proporciona administración de almacenamiento local automatizada desde la CLI o desde la consola web de RHEL.

Dispositivo de bloque

Los dispositivos de bloque se encuentran en la parte inferior de la pila (stack) de almacenamiento y presentan un protocolo de dispositivo estable y uniforme que permite que prácticamente cualquier dispositivo de bloque se incluya de forma transparente en una configuración de almacenamiento RHEL. En la actualidad, se accede a la mayoría de los dispositivos de bloque a través del controlador de dispositivos SCSI de RHEL y aparecen como un dispositivo SCSI, incluidos los discos duros ATA heredados, los dispositivos de estado sólido y los adaptadores de bus de host (HBA) empresariales comunes. RHEL también soporta iSCSI, canal de fibra a través de Ethernet (FCoE), controlador de máquina virtual (`virtio`), almacenamiento conectado en serie (SAS), unidades de memoria exprés no volátil (NVMe) y otros.

Un objetivo iSCSI puede ser un dispositivo físico dedicado en una red o un dispositivo lógico configurado por software iSCSI en un servidor de almacenamiento en red. El objetivo es el extremo del portal en una comunicación de bus de protocolo SCSI, para acceder al almacenamiento como números de unidad lógica (LUN).

El protocolo de canal de fibra a través de Ethernet (FCoE) transmite marcos de canal de fibra a través de redes Ethernet. Por lo general, los centros de datos tienen cableado dedicado de red de área de almacenamiento (SAN) y LAN, cada uno configurado de manera única para su tráfico. Con FCoE, ambos tipos de tráfico se pueden combinar en una arquitectura de red Ethernet convergente más grande. Los beneficios de FCoE incluyen menores costos de hardware y energía.

Multitrayecto

Una ruta es una conexión entre un servidor y el almacenamiento subyacente. Multitrayecto del asignador de dispositivos (`dm-multipath`) es una herramienta de múltiples rutas nativa de RHEL para configurar rutas de E/S redundantes en un único dispositivo lógico de agregación de rutas. Un dispositivo lógico creado con el asignador de dispositivos (`dm`) aparece como un dispositivo de bloque único en `/dev/mapper/` para cada LUN conectado al sistema.

La redundancia de múltiples rutas de almacenamiento también se puede implementar mediante la unión de redes cuando el almacenamiento, como iSCSI y FCoE, usa cableado de red.

Particiones

Un dispositivo de bloque se puede dividir en particiones. Las particiones pueden consumir todo el tamaño del dispositivo de bloque o dividir el dispositivo de bloque para crear varias particiones. Estas particiones se pueden usar para crear un sistema de archivos, dispositivos LVM o se pueden usar directamente para estructuras de bases de datos u otro almacenamiento sin formato.

RAID

Una matriz redundante de discos económicos (RAID) es una tecnología de virtualización de almacenamiento que crea grandes volúmenes lógicos a partir de varios componentes de dispositivos de bloques físicos o virtuales. Las diferentes formas de volúmenes RAID ofrecen redundancia de datos, mejora del rendimiento o ambas cosas, mediante la implementación de diseños de duplicación o división.

LVM soporta los niveles RAID 0, 1, 4, 5, 6 y 10. Los volúmenes lógicos RAID creados y administrados por LVM aprovechan los controladores del kernel de dispositivos múltiples (`mdadm`). Cuando no se usa LVM, Device Mapper RAID (`dm-raid`) proporciona una interfaz de asignación de dispositivos a los controladores del kernel `mdadm`.

Administrador de volúmenes lógicos

Los volúmenes físicos, los grupos de volúmenes y los volúmenes lógicos del LVM se analizaron en una sección anterior. LVM puede tomar casi cualquier forma de dispositivos de bloque físicos o virtuales, y crear almacenamiento como nuevos volúmenes de almacenamiento lógico, ocultando efectivamente la configuración de almacenamiento físico de las aplicaciones y otros clientes de almacenamiento.

Puede apilar volúmenes LVM e implementar funciones avanzadas, como cifrado y compresión, para cada parte de la pila (stack). Existen reglas obligatorias y prácticas recomendadas que se deben seguir para la creación de capas prácticas para escenarios específicos, pero esta introducción se centra solo en presentar los componentes. Las recomendaciones específicas para casos de uso se encuentran en la guía del usuario *Configuración y administración de volúmenes lógicos*.

LVM soporta el *cifrado LUKS*, donde un dispositivo de bloque inferior o una partición se cifra y se presenta como un volumen seguro para crear un sistema de archivos en la parte superior. La ventaja práctica de LUKS sobre el cifrado basado en sistemas de archivos o basado en archivos es que un dispositivo cifrado con LUKS no permite la visibilidad pública ni el acceso a la estructura del sistema de archivos, de modo que un dispositivo físico permanece seguro incluso cuando se elimina de una computadora.

LVM ahora incorpora la *desduplicación y compresión de VDO* como una función configurable de volúmenes lógicos regulares. El cifrado LUKS y VDO se pueden usar junto con volúmenes lógicos, con el cifrado LUKS de LVM habilitado en el volumen de VDO de LVM.

Sistema de archivos u otro uso

La capa superior de la pila (stack) suele ser un sistema de archivos, pero se puede usar como espacio sin formato para bases de datos o requisitos de datos de aplicaciones personalizados. RHEL soporta varios tipos de sistemas de archivos, pero recomienda XFS para la mayoría de los casos de uso modernos. Se requiere XFS cuando la utilidad que implementa LVM es Red Hat Ceph Storage o la herramienta de almacenamiento Stratis.

Las aplicaciones del servidor de bases de datos consumen almacenamiento de diferentes maneras, según su arquitectura y tamaño. Algunas bases de datos más pequeñas almacenan sus estructuras en archivos regulares que están contenidos en un sistema de archivos. Debido a la sobrecarga adicional o las restricciones de acceso al sistema de archivos, esta arquitectura tiene límites de escalamiento. Las bases de datos más grandes que desean omitir el almacenamiento en caché del sistema de archivos y usar sus propios mecanismos de almacenamiento en caché prefieren crear sus estructuras de base de datos en almacenamiento sin formato. Los volúmenes lógicos son adecuados para su uso en bases de datos y otros casos de uso de almacenamiento sin formato.

Red Hat Ceph Storage también prefiere crear sus propias estructuras de metadatos de administración de almacenamiento en dispositivos sin formato que se usarán para crear dispositivos de almacenamiento de objetos (OSD) de Ceph. En las últimas versiones de Red Hat Ceph Storage, Ceph usa LVM para inicializar dispositivos de disco para usar como OSD. Para obtener más información, consulte el curso *Cloud Storage with Red Hat Ceph Storage* (CL260).

Administración de almacenamiento de Stratis

Stratis es una herramienta de administración de almacenamiento local desarrollada por Red Hat y la comunidad upstream de Fedora. Stratis facilita la configuración inicial del almacenamiento, realizar cambios en la configuración del almacenamiento y usar funciones avanzadas de almacenamiento.



Importante

Stratis está disponible actualmente como Vista previa de tecnología, pero se espera que cuente con soporte en una versión posterior de RHEL 9. Para obtener información sobre el alcance de soporte de Red Hat para funciones de vista previa de tecnología, consulte el documento Alcance del soporte de las prestaciones de tecnología [<https://access.redhat.com/support/offerings/techpreview>].

Red Hat recomienda a los clientes que envíen sus comentarios al implementar Stratis.

Stratis se ejecuta como un servicio que administra conjuntos (pools) de dispositivos de almacenamiento físico, y crea y administra de forma transparente los volúmenes para los sistemas de archivos recién creados.

Stratis compila los sistemas de archivos desde conjuntos (pools) de dispositivos de disco compartidos usando un concepto conocido como *aprovisionamiento ligero*. En lugar de asignar inmediatamente espacio físico de almacenamiento al sistema de archivos cuando lo crea, Stratis dinámicamente asigna ese espacio del conjunto (pool) a medida que el sistema de archivos almacena más datos. Por lo tanto, puede parecer que el sistema de archivos tiene un tamaño de 1 TiB, pero solo puede tener 100 GiB de almacenamiento real, que se le asignó realmente desde el conjunto (pool).

Puede crear múltiples conjuntos (pools) con diferentes dispositivos de almacenamiento. Desde cada conjunto (pool), puede crear uno o más sistemas de archivos. Actualmente, puede crear hasta 2^{24} sistemas de archivos por conjunto (pool).

Stratis compila los componentes que conforman un sistema de archivos administrado por Stratis a partir de componentes estándares de Linux. Internamente, Stratis usa la infraestructura del asignador de dispositivos que también usa LVM. Stratis formatea los sistemas de archivos administrados con XFS.

Figura 6.3 ilustra cómo Stratis ensambla los elementos de la solución de administración de almacenamiento. Stratis asigna dispositivos de almacenamiento de bloques, como discos duros o SSD, a conjuntos (pools), cada uno de los cuales contribuye a un almacenamiento físico en el conjunto. A continuación, crea sistemas de archivos a partir de los conjuntos (pools) y asigna el almacenamiento físico a cada sistema de archivos según sea necesario.

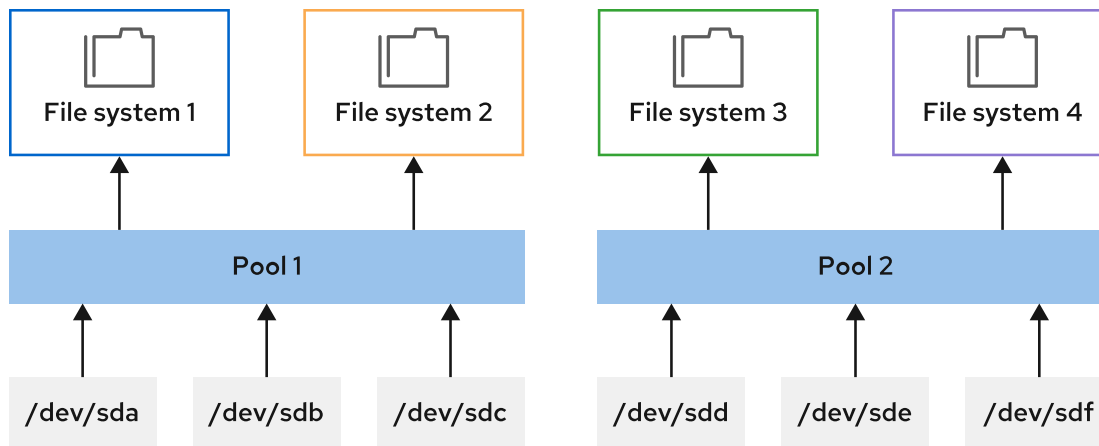


Figura 6.3: Arquitectura de Stratis

Métodos de administración de Stratis

Para administrar los sistemas de archivos con la solución de administración de almacenamiento Stratis, instale los paquetes `stratis-cli` y `stratisd`. El paquete `stratis-cli` proporciona el comando `stratis`, que envía solicitudes de reconfiguración al daemon del sistema `stratisd`. El paquete `stratisd` proporciona el servicio `stratisd`, que maneja los pedidos de reconfiguración y administra y monitorea los dispositivos de bloque, los conjuntos (pools) y los sistemas de archivos de Stratis.

La administración de Stratis está incluida en la consola web de RHEL.

**Advertencia**

Reconfigure los sistemas de archivos creados por Stratis solo con herramientas y comandos de Stratis.

Stratis usa metadatos almacenados para reconocer conjuntos (pools) administrados, volúmenes y sistemas de archivos. La configuración manual de los sistemas de archivos Stratis con comandos que no son de Stratis podría causar la sobrescritura de esos metadatos e impedir que Stratis reconozca los sistemas de archivos que ha creado.

Instalación y habilitación de Stratis

Para usar Stratis, asegúrese de que su sistema tenga el software requerido y de que el servicio `stratisd` se esté ejecutando. Instale los paquetes `stratis-cli` y `stratisd`, inicie y habilite el servicio `stratisd`.

```
[root@host ~]# dnf install stratis-cli stratisd
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
[root@host ~]# systemctl enable --now stratisd
```

Creación de conjuntos (pools) de Stratis

Cree conjuntos (pools) de uno o más dispositivos de bloque con el comando `stratis pool create`. A continuación, use el comando `stratis pool list` para ver la lista de conjuntos (pools) disponibles.

```
[root@host ~]# stratis pool create pool1 /dev/vdb
[root@host ~]# stratis pool list
```

Name	Total Physical	Properties	UUID
pool1	5 GiB / 37.63 MiB / 4.96 GiB	~Ca,~Cr	11f6f3c5-5...

**Advertencia**

El comando `stratis pool list` muestra el espacio de almacenamiento real que está en uso y el espacio del conjunto (pool) que aún está disponible. Actualmente, si un conjunto se llena, los datos adicionales escritos en los sistemas de archivos del conjunto (pool) se descartan discretamente.

Use el comando `stratis pool add-data` para agregar dispositivos de bloque adicionales a un conjunto (pool). A continuación, use el comando `stratis blockdev list` para verificar los dispositivos de bloque de un conjunto (pool).

```
[root@host ~]# stratis pool add-data pool1 /dev/vdc
[root@host ~]# stratis blockdev list pool1
```

Pool Name	Device Node	Physical Size	Tier
pool1	/dev/vdb	5 GiB	Data
pool1	/dev/vdc	5 GiB	Data

Administración de sistemas de archivos de Stratis

Use el comando `stratis filesystem create` para crear un sistema de archivos de un conjunto (pool). Los enlaces a los sistemas de archivos Stratis están en el directorio `/dev/stratis/pool1`. Use el comando `stratis filesystem list` para ver la lista de sistemas de archivos disponibles.

```
[root@host ~]# stratis filesystem create pool1 fs1
[root@host ~]# stratis filesystem list
```

Pool Name	Name	Used	Created	Device	UUID
pool1	fs1	546 MiB	Apr 08 2022 04:05	/dev/stratis/pool1/fs1	c7b5719...

Cree una instantánea del sistema de archivos Stratis con el comando `stratis filesystem snapshot`. Las capturas de instantáneas son independientes de los sistemas de archivos de origen. Stratis asigna dinámicamente el espacio de almacenamiento de instantáneas y usa 560 MB iniciales para almacenar el diario (journal) del sistema de archivos.

```
[root@host ~]# stratis filesystem snapshot pool1 fs1 snapshot1
```

Montaje de manera persistente de sistemas de archivos de Stratis

Para asegurarse de que los sistemas de archivos Stratis se monten de forma persistente, edite el archivo `/etc/fstab` y especifique los detalles del sistema de archivos. Use el comando `lsblk` para mostrar el UUID del sistema de archivos que debe usar en el archivo `/etc/fstab` para identificar el sistema de archivos. También puede usar el comando `stratis filesystem list` para obtener el UUID del sistema de archivos.

```
[root@host ~]# lsblk --output=UUID /dev/stratis/pool1/fs1
UUID
c7b57190-8fba-463e-8ec8-29c80703d45e
```

La siguiente es una entrada de ejemplo en el archivo `/etc/fstab` para montar de forma persistente un sistema de archivos Stratis. Esta entrada de ejemplo es una sola línea extensa en el archivo. La opción de montaje `x-systemd.requires=stratisd.service` retrasa el montaje del sistema de archivos hasta después de que el daemon `systemd` inicia el servicio `stratisd` durante el proceso de arranque.

```
UUID=c7b57190-8fba-463e-8ec8-29c80703d45e /dir1 xfs defaults,x-
systemd.requires=stratisd.service 0 0
```



Importante

Si no incluye la opción de montaje `x-systemd.requires=stratisd.service` en el archivo `/etc/fstab` para cada sistema de archivos Stratis, la máquina no se iniciará correctamente y se abortará en caso de `emergency.target` la próxima vez que la reinicie.

**Advertencia**

No use el comando `df` para consultar el espacio del sistema de archivos Stratis.

El comando `df` informa que cualquier sistema de archivos XFS administrado por Stratis tiene un tamaño de 1 TiB, independientemente de la asignación actual. Debido a que el sistema de archivos tiene un aprovisionamiento ligero, es posible que un conjunto (pool) no tenga suficiente almacenamiento físico para respaldar todo el sistema de archivos. Otros sistemas de archivos en el conjunto (pool) podrían usar todo el almacenamiento disponible.

Por lo tanto, es posible consumir todo el conjunto (pool) de almacenamiento, incluso cuando el comando `df` informa que el sistema de archivos tiene espacio disponible. Escribir en un sistema de archivos sin almacenamiento de conjunto (pool) disponible puede fallar.

En su lugar, use siempre el comando `stratis pool list` para monitorear con precisión el almacenamiento disponible de un conjunto (pool).

**Referencias**

Para obtener más información, consulte *Deduplicating And Compressing Logical Volumes On RHEL* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/deduplicating_and_compressing_logical_volumes_on_rhel/index

Para obtener más información, consulte *Red Hat Enterprise Linux 9 Managing File Systems Guide* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/managing_file_systems

Almacenamiento de Stratis

<https://stratis-storage.github.io/>

Lecciones que Stratis aprendió de ZFS, Btrfs y Linux Volume Manager

<https://opensource.com/article/18/4/stratis-lessons-learned>

► Ejercicio Guiado

Administración de almacenamiento en capas

En este ejercicio, usa Stratis para crear sistemas de archivos a partir de conjuntos (pools) de almacenamiento provistos por dispositivos de almacenamiento físicos.

Resultados

- Crear un sistema de archivos con aprovisionamiento ligero con la solución de administración de almacenamiento Stratis.
- Verificar que los volúmenes Stratis crezcan dinámicamente para soportar crecimiento de datos en tiempo real.
- Acceder a los datos desde la instantánea de un sistema de archivos con aprovisionamiento ligero.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start lvm-stratis
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Instale los paquetes `stratisd` y `stratis-cli`.

```
[root@servera ~]# dnf install stratisd stratis-cli
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 3. Active el servicio `stratisd`.

```
[root@servera ~]# systemctl enable --now stratisd
```

- 4. Asegúrese de que el conjunto (pool) de Stratis `stratispool1` existe con el dispositivo de bloque `/dev/vdb`.

4.1. Cree el conjunto (pool) de Stratis `stratispool1`.

```
[root@servera ~]# stratis pool create stratispool1 /dev/vdb
```

4.2. Verifique la disponibilidad del conjunto (pool) `stratispool1`. Observe el tamaño del conjunto (pool).

```
[root@servera ~]# stratis pool list
```

Name	Total Physical	Properties	UUID
stratispool1	5 GiB / 37.63 MiB / 4.96 GiB	~Ca,~Cr	3557c389-7...

- 5. Amplíe la capacidad del conjunto (pool) `stratispool1` con el dispositivo de bloque `/dev/vdc`.

5.1. Agregue el dispositivo de bloque `/dev/vdc` al conjunto (pool) `stratispool1`.

```
[root@servera ~]# stratis pool add-data stratispool1 /dev/vdc
```

5.2. Verifique el tamaño del conjunto (pool) `stratispool1`. El tamaño del conjunto (pool) `stratispool1` aumenta cuando agrega el dispositivo de bloque.

```
[root@servera ~]# stratis pool list
```

Name	Total Physical	Properties	UUID
stratispool1	10 GiB / 41.63 MiB / 9.96 GiB	~Ca,~Cr	3557c389-7...

5.3. Verifique los dispositivos de bloque que son actualmente miembros del conjunto (pool) `stratispool1`.

```
[root@servera ~]# stratis blockdev list stratispool1
```

Pool Name	Device Node	Physical Size	Tier
stratispool1	/dev/vdb	5 GiB	Data
stratispool1	/dev/vdc	5 GiB	Data

- 6. Agregue un sistema de archivos con aprovisionamiento ligero denominado `stratis-filesystem1` en el conjunto (pool) `stratispool1`. Monte el sistema de archivos en el directorio `/stratisvol`. Cree un archivo en el sistema de archivos `stratis-filesystem1` denominado `file1` que contenga el texto `Hello World!`. Modifique el archivo `/etc/fstab` para montar de forma persistente el sistema de archivos en el directorio `/stratisvol`. Reinicie su sistema y verifique que el sistema de archivos se monte de manera persistente en los reinicios.

6.1. Cree el sistema de archivos con aprovisionamiento ligero `stratis-filesystem1` en el conjunto (pool) `stratispool1`. Puede tardar hasta un minuto para que se complete el comando.

```
[root@servera ~]# stratis filesystem create stratispool1 stratis-filesystem1
```

- 6.2. Verifique la disponibilidad del sistema de archivos `stratis-filesystem1`. Observe el uso actual del sistema de archivos `stratis-filesystem1`. Este uso del sistema de archivos aumentará a pedido en los siguientes pasos.

```
[root@servera ~]# stratis filesystem list
```

Pool Name	Name	Used	Created	Device
		UUID		
stratispool1	stratis-filesystem1	546 MiB	Apr 08 2022 07:12	/dev/stratis/stratispool1/stratis-filesystem1
		48e8...		

- 6.3. Cree el directorio `/stratisvol`.

```
[root@servera ~]# mkdir /stratisvol
```

- 6.4. Monte el sistema de archivos `stratis-filesystem1` en el directorio `/stratisvol`.

```
[root@servera ~]# mount /dev/stratis/stratispool1/stratis-filesystem1 \
/stratisvol
```

- 6.5. Verifique que el volumen `stratis-filesystem1` esté montado en el directorio `/stratisvol`.

```
[root@servera ~]# mount
...output omitted...
/dev/mapper/stratis-1-3557...fbd3-thin-fs-48e8...9ebe on /stratisvol type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,sunit=2048,swidth=2048,
noquota)
```

- 6.6. Cree el archivo de texto `/stratisvol/file1`.

```
[root@servera ~]# echo "Hello World!" > /stratisvol/file1
```

- 6.7. Obtenga el UUID del sistema de archivos. Tenga en cuenta que el UUID podría ser diferente en su sistema.

```
[root@servera ~]# lsblk --output=UUID \
/dev/stratis/stratispool1/stratis-filesystem1
UUID
d18cb4fc-753c-473a-9ead-d6661533b475
```

- 6.8. Modifique el archivo `/etc/fstab` para montar de forma persistente el sistema de archivos en el directorio `/stratisvol`. Para ello, use el comando `vim /etc/fstab` y agregue la siguiente línea. Reemplace el UUID por el correcto para su sistema.

```
UUID=d18c... /stratisvol xfs defaults,x-systemd.requires=stratisd.service 0 0
```

- 6.9. Reinicie su sistema y verifique que el sistema de archivos se monte de manera persistente en los reinicios.

```
[root@servera ~]# systemctl reboot
...output omitted...
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]# mount
...output omitted...
/dev/mapper/stratis-1-3557...fbd3-thin-fs-d18c...b475 on /stratisvol type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,sunit=2048,swidth=2048,
noquota,x-systemd.requires=stratisd.service)
```

- 7. Verifique que el sistema de archivos con aprovisionamiento ligero `stratis-filesystem1` crece dinámicamente a medida que crecen los datos en el sistema de archivos.

7.1. Vea el uso actual del sistema de archivos `stratis-filesystem1`.

```
[root@servera ~]# stratis filesystem list
```

Pool Name	Name	Used	Created	Device
		UUID		
stratispool1	stratis-filesystem1	546 MiB	Apr 08 2022 07:12	/dev/stratis/
	stratispool1/stratis-filesystem1	48e8...		

- 7.2. Cree un archivo de 2 GiB en el sistema de archivos `stratis-filesystem1`. Puede tardar hasta un minuto para que se complete el comando.

```
[root@servera ~]# dd if=/dev/urandom of=/stratisvol/file2 bs=1M count=2048
```

- 7.3. Verifique el espacio usado en el sistema de archivos `stratis-filesystem1`.

La salida muestra que el espacio usado en el sistema de archivos `stratis-filesystem1` ha aumentado. El aumento del espacio usado confirma que el sistema de archivos de aprovisionamiento ligero se amplía dinámicamente según sea necesario.

```
[root@servera ~]# stratis filesystem list
```

Pool Name	Name	Used	Created	Device
		UUID		
stratispool1	stratis-filesystem1	2.60 GiB	Apr 08 2022 07:12	/dev/stratis/
	stratispool1/stratis-filesystem1	48e8...		

- 8. Cree una instantánea del sistema de archivos `stratis-filesystem1` denominada `stratis-filesystem1-snap`. La instantánea le proporcionará acceso a cualquier archivo que elimine del sistema de archivos `stratis-filesystem1`.

- 8.1. Cree una instantánea del sistema de archivos `stratis-filesystem1`. Puede tardar hasta un minuto para que se complete el comando.

```
[root@servera ~]# stratis filesystem snapshot stratispool1 \
stratis-filesystem1 stratis-filesystem1-snap
```

- 8.2. Verifique la disponibilidad de la instantánea.

```
[root@servera ~]# stratis filesystem list
```

Pool Name	Name	Used	Created	Device
		UUID		
stratispool1	stratis-filesystem1-snap	2.73 GiB	Apr 08 2022 07:22	/dev/
	stratis/stratispool1/stratis-filesystem1-snap	5774...		
stratispool1	stratis-filesystem1	2.73 GiB	Apr 08 2022 07:12	/dev/
	stratis/stratispool1/stratis-filesystem1	48e8...		

8.3. Quite el archivo `/stratisvol/file1`.

```
[root@servera ~]# rm /stratisvol/file1
rm: remove regular file '/stratisvol/file1'? y
```

8.4. Cree el directorio `/stratisvol-snap`.

```
[root@servera ~]# mkdir /stratisvol-snap
```

8.5. Monte la instantánea `stratis-filesystem1-snap` en el directorio `/stratisvol-snap`.

```
[root@servera ~]# mount /dev/stratis/stratispool1/stratis-filesystem1-snap \
/stratisvol-snap
```

8.6. Verifique que aún puede acceder al archivo que eliminó del sistema de archivos `stratis-filesystem1` en la instantánea.

```
[root@servera ~]# cat /stratisvol-snap/file1
Hello World!
```

► 9. Desmonte los volúmenes `/stratisvol` y `/stratisvol-snap`.

```
[root@servera ~]# umount /stratisvol-snap
[root@servera ~]# umount /stratisvol
```

► 10. Elimine el sistema de archivos con aprovisionamiento ligero `stratis-filesystem1` y su instantánea `stratis-filesystem1-snap` del sistema.

10.1. Destruya la instantánea `stratis-filesystem1-snap`.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1-snap
```

10.2. Destruya el sistema de archivos `stratis-filesystem1`.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1
```

10.3. Regrese al sistema `workstation` como el usuario `student`.


```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish lvm-stratis
```

Esto concluye la sección.

► Trabajo de laboratorio

Administración de la pila (stack) de almacenamiento

En este trabajo de laboratorio, modifica el tamaño de un volumen lógico existente, agrega recursos de LVM según sea necesario y, luego, agrega un nuevo volumen lógico con un sistema de archivos XFS montado de manera persistente en este.

Resultados

- Cambiar el tamaño del volumen lógico `serverb_01_lv` a 768 MiB.
- Cree el nuevo volumen lógico `serverb_02_lv` con 128 MiB con un sistema de archivos XFS.
- Monte el volumen de manera persistente en el directorio `/storage/data2`.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start lvm-review
```

Instrucciones

En la máquina `serverb`, el volumen lógico `serverb_01_lv` montado en el directorio `/storage/data1` se está quedando sin espacio en disco y debe ampliarse a 768 MiB. Debe asegurarse de que el LV `serverb_01_lv` permanece montado de forma persistente en el directorio `/storage/data1`.

El LV `serverb_01_lv` está presente en el grupo de volúmenes `serverb_01_vg`. Desafortunadamente, no tiene espacio suficiente para ampliar el volumen lógico existente. Existe una partición de 512 MiB en el disco `/dev/vdb`. Cree una nueva partición con el tamaño sucesivo de 512 MiB en el disco `/dev/vdb`.

Cree el LV `serverb_02_lv` con 128 MiB. Cree el sistema de archivos XFS en el volumen creado recientemente. Monte el volumen lógico recién creado en el directorio `/storage/data2`

1. Cree una partición de 512 MiB en el disco `/dev/vdb`. Inicialice esta partición como volumen físico y amplíe el grupo de volúmenes `serverb_01_vg` para usar esta partición.
2. Amplíe el volumen lógico `serverb_01_lv` a 768 MiB.
3. En el grupo de volúmenes existente, cree un nuevo volumen lógico `serverb_02_lv` con 128 MiB. Agregue un sistema de archivos XFS y móntelo de forma persistente en el directorio `/storage/data2`.
4. Verifique que el LV creado recientemente esté montado con el tamaño deseado.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade lvm-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish lvm-review
```

Esto concluye la sección.

► Solución

Administración de la pila (stack) de almacenamiento

En este trabajo de laboratorio, modifica el tamaño de un volumen lógico existente, agrega recursos de LVM según sea necesario y, luego, agrega un nuevo volumen lógico con un sistema de archivos XFS montado de manera persistente en este.

Resultados

- Cambiar el tamaño del volumen lógico `serverb_01_lv` a 768 MiB.
- Cree el nuevo volumen lógico `serverb_02_lv` con 128 MiB con un sistema de archivos XFS.
- Monte el volumen de manera persistente en el directorio `/storage/data2`.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start lvm-review
```

Instrucciones

En la máquina `serverb`, el volumen lógico `serverb_01_lv` montado en el directorio `/storage/data1` se está quedando sin espacio en disco y debe ampliarse a 768 MiB. Debe asegurarse de que el LV `serverb_01_lv` permanece montado de forma persistente en el directorio `/storage/data1`.

El LV `serverb_01_lv` está presente en el grupo de volúmenes `serverb_01_vg`. Desafortunadamente, no tiene espacio suficiente para ampliar el volumen lógico existente. Existe una partición de 512 MiB en el disco `/dev/vdb`. Cree una nueva partición con el tamaño sucesivo de 512 MiB en el disco `/dev/vdb`.

Cree el LV `serverb_02_lv` con 128 MiB. Cree el sistema de archivos XFS en el volumen creado recientemente. Monte el volumen lógico recién creado en el directorio `/storage/data2`

1. Cree una partición de 512 MiB en el disco `/dev/vdb`. Inicialice esta partición como volumen físico y amplíe el grupo de volúmenes `serverb_01_vg` para usar esta partición.
 - 1.1. Inicie sesión en la máquina `serverb` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 1.2. Cree la partición de 512 MiB y establezca el tipo de partición lvm.

```
[root@serverb ~]# parted /dev/vdb mkpart primary 514MiB 1026MiB
[root@serverb ~]# parted /dev/vdb set 2 lvm on
```

- 1.3. Registre la nueva partición con el kernel.

```
[root@serverb ~]# udevadm settle
```

- 1.4. Inicialice la partición como un PV.

```
[root@serverb ~]# pvcreate /dev/vdb2
Physical volume "/dev/vdb2" successfully created.
```

- 1.5. Extienda el VG serverb_01_vg con el nuevo PV /dev/vdb2.

```
[root@serverb ~]# vgextend serverb_01_vg /dev/vdb2
Volume group "serverb_01_vg" successfully extended
```

2. Amplíe el volumen lógico serverb_01_lv a 768 MiB.

- 2.1. Amplíe el LV serverb_01_lv a 768 MiB.

O también podría usar la opción `-L +512M` del comando `lvcreate` para cambiar el tamaño del LV.

```
[root@serverb ~]# lvextend -L 768M /dev/serverb_01_vg/serverb_01_lv
Size of logical volume serverb_01_vg/serverb_01_lv changed from 256.00 MiB (64
extents) to 768.00 MiB (192 extents).
Logical volume serverb_01_vg/serverb_01_lv successfully resized.
```

- 2.2. Amplíe el sistema de archivos XFS que consume el espacio restante del LV.

```
[root@serverb ~]# xfs_growfs /storage/data1
meta-data=/dev/mapper/serverb_01_vg-serverb_01_lv isize=512    agcount=4,
  agsize=16384 blks
...output omitted...
data blocks changed from 65536 to 196608
```



nota

El comando `xfs_growfs` introduce un paso adicional para ampliar el sistema de archivos. Una alternativa sería usar la opción `-r` del comando `lvextend`.

3. En el grupo de volúmenes existente, cree un nuevo volumen lógico `serverb_02_lv` con 128 MiB. Agregue un sistema de archivos XFS y móntelo de forma persistente en el directorio `/storage/data2`.

3.1. Cree el LV `serverb_02_lv` con 128 MiB desde el VG `serverb_01_vg`.

```
[root@serverb ~]# lvcreate -n serverb_02_lv -L 128M serverb_01_vg
Logical volume "serverb_02_lv" created.
```

3.2. Cree el sistema de archivos xfs en el LV `serverb_02_lv`.

```
[root@serverb ~]# mkfs -t xfs /dev/serverb_01_vg/serverb_02_lv
...output omitted...
```

3.3. Cree el directorio `/storage/data2` como el punto de montaje.

```
[root@serverb ~]# mkdir /storage/data2
```

3.4. Agregue la línea siguiente al final del archivo `/etc/fstab`:

```
/dev/serverb_01_vg/serverb_02_lv /storage/data2 xfs defaults 0 0
```

3.5. Actualice el daemon `systemd` con el nuevo archivo de configuración `/etc/fstab`.

```
[root@serverb ~]# systemctl daemon-reload
```

3.6. Monte el LV `serverb_02_lv`.

```
[root@serverb ~]# mount /storage/data2
```

4. Verifique que el LV creado recientemente esté montado con el tamaño deseado.

4.1. Use el comando `df` para verificar el tamaño del LV `serverb_01_lv`.

```
[root@serverb ~]# df -h /storage/data1
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/serverb_01_vg-serverb_01_lv  763M   19M  744M   3% /storage/data1
```

4.2. Verifique el tamaño del LV `serverb_02_lv`.

```
[root@serverb ~]# df -h /storage/data2
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/serverb_01_vg-serverb_02_lv  123M   7.6M  116M   7% /storage/data2
```

4.3. Verifique los detalles del LV `serverb_01_lv`.

```
[root@serverb ~]# lvsdisplay /dev/serverb_01_vg/serverb_01_lv
--- Logical volume ---
LV Path                /dev/serverb_01_vg/serverb_01_lv
LV Name                 serverb_01_lv
```

```

VG Name                serverb_01_vg
LV UUID                1pY3DZ-fs1F-mptC-fL32-e8tG-PFBT-bs7LSJ
LV Write Access        read/write
LV Creation host, time serverb.lab.example.com, 2022-05-05 14:40:51 -0400
LV Status              available
# open                 1
LV Size                768.00 MiB
Current LE             192
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device           253:0

```

4.4. Verifique los detalles del LV `serverb_02_lv`.

```

[root@serverb ~]# lvdisplay /dev/serverb_01_vg/serverb_02_lv
--- Logical volume ---
LV Path                /dev/serverb_01_vg/serverb_02_lv
LV Name                serverb_02_lv
VG Name                serverb_01_vg
LV UUID                0aJIb6-Ti2b-jLCk-imb6-rkLx-mUoX-acjkz9
LV Write Access        read/write
LV Creation host, time serverb.lab.example.com, 2022-05-05 14:45:46 -0400
LV Status              available
# open                 1
LV Size                128.00 MiB
Current LE             32
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    8192
Block device           253:1

```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade lvm-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish lvm-review
```

Esto concluye la sección.

Resumen

- Puede usar LVM para crear un almacenamiento flexible al asignar espacio en múltiples dispositivos de almacenamiento.
- Los volúmenes físicos, los grupos de volúmenes y los volúmenes lógicos se administran mediante los comandos `pvccreate`, `vgreduce` y `lvextend`.
- Los volúmenes lógicos se pueden formatear con un sistema de archivos o espacio de intercambio (swap), y se pueden montar de forma persistente.
- Se puede agregar almacenamiento adicional a los grupos de volúmenes y los volúmenes lógicos pueden ampliarse de forma dinámica.
- Comprender las capas y los componentes de la pila (stack) de almacenamiento para administrar el almacenamiento de manera eficiente.
- El optimizador de datos virtual (VDO) usa LVM para la compresión y deduplicación de datos.
- Puede usar Stratis para realizar una configuración de almacenamiento inicial o habilitar funciones de almacenamiento avanzadas.

capítulo 7

Acceso al almacenamiento conectado a la red

Meta

Acceder al almacenamiento conectado a la red con el protocolo NFS.

Objetivos

- Identificar la información de exportación de NFS, crear un directorio para usar como punto de montaje, montar una exportación de NFS con el comando `mount` o mediante la configuración del archivo `/etc/fstab` y desmonte una exportación de NFS con el comando `umount`.
- Describir los beneficios de usar el servicio de automontaje y las exportaciones de NFS de automontaje mediante el uso de asignaciones directas e indirectas.

Secciones

- Administración de almacenamiento conectado a la red con NFS (y ejercicio guiado)
- Montaje automático de almacenamiento conectado a la red (y ejercicio guiado)

Trabajo de laboratorio

Acceso al almacenamiento conectado a la red

Administración de almacenamiento conectado a la red con NFS

Objetivos

Identificar la información de exportación de NFS, crear un directorio para usar como punto de montaje, montar una exportación de NFS con el comando `mount` o mediante la configuración del archivo `/etc/fstab` y desmonte una exportación de NFS con el comando `umount`.

Acceso a directorios NFS exportados

El *sistema de archivos de red* (NFS) es un protocolo estándar de Internet que usan Linux, UNIX y sistemas operativos similares como su sistema de archivos de red nativo. NFS es un estándar abierto que soporta permisos nativos de Linux y atributos del sistema de archivos.

De forma predeterminada, Red Hat Enterprise Linux 9 usa la versión 4.2 de NFS. RHEL soporta totalmente los protocolos NFSv3 y NFSv4. NFSv3 puede usar un protocolo de transporte TCP o UDP, pero NFSv4 solo permite conexiones TCP.

Los servidores NFS *exportan* directorios. Los clientes NFS montan directorios exportados en un directorio de punto de montaje local existente. Los clientes NFS pueden montar directorios exportados de varias maneras:

- **Manualmente**, con el comando `mount`.
- **De forma persistente en el arranque** mediante la configuración de entradas en el archivo `/etc/fstab`.
- **A pedido** mediante la configuración de un método de automontaje.

Los métodos de automontaje, que incluyen el servicio `autofs` y la utilidad `systemd.automount`, se analizan en la sección **Automount Network-Attached Storage**. Debe instalar el paquete `nfs-utils` para obtener las herramientas de cliente para el montaje manual, o para el automontaje, para obtener directorios NFS exportados.

```
[root@host ~]# dnf install nfs-utils
```

RHEL también soporta el montaje de directorios *compartidos* desde sistemas Microsoft Windows mediante el uso de los mismos métodos que para el protocolo NFS, mediante el uso de los protocolos Server Message Block (SMB) o Common Internet File System (CIFS). Las opciones de montaje son específicas del protocolo y dependen de la configuración de Windows Server o Samba Server.

Consultar los directorios NFS exportados de un servidor

El protocolo NFS cambió significativamente entre NFSv3 y NFSv4. El método para consultar un servidor para ver las exportaciones disponibles es diferente para cada versión de protocolo.

NFSv3 usó el protocolo RPC, que requiere un servidor de archivos que soporta conexiones NFSv3 para ejecutar el servicio `rpcbind`. Un cliente NFSv3 se conecta al servicio `rpcbind` en el puerto 111 en el servidor para solicitar el servicio NFS. El servidor responde con el puerto actual para el servicio NFS. Use el comando `showmount` para consultar las exportaciones disponibles en un servidor NFSv3 basado en RPC.

```
[root@host ~]# showmount --exports server
```

```
Export list for server
/shares/test1
/shares/test2
```

El protocolo NFSv4 eliminó el uso del protocolo RPC heredado para las transacciones NFS. El uso del comando `showmount` en un servidor que soporta solo el tiempo de espera de NFSv4 sin recibir una respuesta, ya que el servicio `rpcbind` no se está ejecutando en el servidor. Sin embargo, consultar un servidor NFSv4 es más sencillo que consultar un servidor NFSv3.

NFSv4 introdujo un *árbol de exportación* que contiene todas las rutas para los directorios exportados del servidor. Para ver todos los directorios exportados, monte la raíz (/) del árbol de exportación del servidor. Montar la raíz (root) del árbol de exportación proporciona rutas navegables para todos los directorios exportados, como elementos secundarios del directorio raíz del árbol, pero no monta ("vincula") ninguno de los directorios exportados.

```
[root@host ~]# mkdir /mountpoint
[root@host ~]# mount server:/ /mountpoint
[root@host ~]# ls /mountpoint
```

Para montar una exportación de NFSv4 mientras navega por el árbol de exportación montado, cambie el directorio por una ruta de directorio exportado. De manera alternativa, use el comando `mount` con el nombre de ruta completo de un directorio exportado para montar un único directorio exportado. Los directorios exportados que usan la seguridad de Kerberos no permiten montar o acceder a un directorio mientras se navega por un árbol de exportación, aunque puede ver el nombre de la ruta de exportación. El montaje de recursos compartidos protegidos por Kerberos requiere una configuración de servidor adicional y el uso de credenciales de usuario de Kerberos, que se analizan en el curso de capacitación Red Hat Security: Identity Management and Active Directory Integration (RH362).

Montaje manual de directorios NFS exportados

Después de identificar la exportación de NFS que se montará, cree un punto de montaje local si aún no existe. El directorio `\mnt` está disponible para su uso como punto de montaje temporal, pero la práctica recomendada es no usar `\mnt` para el montaje a largo plazo o persistente.

```
[root@host ~]# mkdir /mountpoint
```

Al igual que con los sistemas de archivos de volúmenes locales, monte la exportación de NFS para acceder a su contenido. Los recursos compartidos de NFS pueden montarse de forma temporal o permanente, solo por un usuario con privilegios.

```
[root@host ~]# mount -t nfs -o rw,sync server:/export /mountpoint
```

La opción `-t nfs` especifica el tipo de sistema de archivos NFS. Sin embargo, cuando el comando `mount` detecta la sintaxis `server:/export`, el comando toma el tipo NFS de manera predeterminada. La opción `-o sync` especifica que todas las transacciones al sistema de archivos exportado se realizan de forma sincrónica, lo que se recomienda encarecidamente para todos los montajes de red de producción donde las transacciones deben completarse o, de lo contrario, se devuelven como fallidas.

El uso de un comando manual `mount` no es persistente. Cuando se reinicia el sistema, esa exportación de NFS aún no se montará. Los montajes manuales son útiles para proporcionar acceso temporal a un directorio exportado o para probar el montaje de una exportación de NFS antes de montarla de forma persistente.

Montaje de manera persistente de directorios NFS exportados

Para montar de forma persistente una exportación de NFS, edite el archivo `/etc/fstab` y agregue la entrada de montaje con una sintaxis similar al montaje manual.

```
[root@host ~]# vim /etc/fstab
...
server:/export /mountpoint nfs rw,soft 0 0
```

A continuación, puede montar la exportación de NFS usando solo el punto de montaje. El comando `mount` obtiene el servidor NFS y las opciones de montaje de la entrada coincidente en el archivo `/etc/fstab`.

```
[root@host ~]# mount /mountpoint
```

Desmontaje de directorios NFS exportados

Como usuario privilegiado, desmonte una exportación de NFS con el comando `umount`. Desmontar un recurso compartido no elimina su entrada en el archivo `/etc/fstab` si existe. Las entradas en el archivo `/etc/fstab` son persistentes y se vuelven a montar durante el arranque.

```
[root@host ~]# umount /mountpoint
```

Un directorio montado a veces puede fallar al desmontar y devuelve un error que el `device is busy`. El dispositivo está ocupado porque una aplicación mantiene un archivo abierto dentro del sistema de archivos o porque la shell de algún usuario tiene un directorio de trabajo en el directorio `root` del sistema de archivos montado o debajo de él.

Para resolver el error, verifique sus propias ventanas de shell activas y use el comando `cd` para dejar el sistema de archivos montado. Si los intentos posteriores de desmontar el sistema de archivos aún fallan, use el comando `lssof` (*enumerar archivos abiertos*) para consultar el punto de montaje. El comando `lssof` devuelve una lista de nombres de archivos abiertos y el proceso que mantiene el archivo abierto.

```
[root@host ~]# lssof /mountpoint
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
program 5534 user txt REG 252.4 910704 128 /home/user/program
```

Con esta información, cierre correctamente cualquier proceso que esté usando archivos en este sistema de archivos y vuelva a intentar el desmontaje. Solo en escenarios críticos, cuando una aplicación no se puede cerrar correctamente, finalice el proceso para cerrar el archivo. De manera alternativa, use la opción `umount -f` para forzar el desmontaje, que puede causar la pérdida de datos no escritos para todos los archivos abiertos.



Referencias

Páginas del manual: `mount(8)`, `umount(8)`, `showmount(8)`, `fstab(5)`, `mount.nfs(8)`, `nfsconf(8)` y `rpcbind(8)`

► Ejercicio Guiado

Administración de almacenamiento conectado a la red con NFS

Lista de verificación de rendimiento

En este ejercicio, modifica el archivo `/etc/fstab` para montar de forma persistente una exportación de NFS en el momento del arranque.

Resultados

- Probar un servidor NFS con el comando `mount`.
- Configurar exportaciones de NFS en el archivo de configuración `/etc/fstab` para guardar los cambios incluso después de reiniciar el sistema.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start netstorage-nfs
```

Instrucciones

Una empresa de transporte usa un servidor NFS central, `serverb`, para alojar diversos documentos y directorios exportados. Los usuarios de `servera`, que son todos miembros del grupo `admin`, necesitan acceso a la exportación NFS montada de forma persistente.

Características del entorno:

- La máquina `serverb` exporta el directorio `/shares/public`, que contiene algunos archivos de texto.
- Los miembros del grupo `admin` (`admin1`, `sysmanager1`) tienen acceso de lectura y escritura al directorio exportado `/shares/public`.
- El punto de montaje en `servera` debe ser el directorio `/public`.
- Todas las contraseñas de usuario están definidas en `redhat`.
- El paquete `nfs-utils` ya esté instalado.

- 1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

- 1.1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

► 2. Pruebe el servidor NFS en serverb con servera como cliente NFS.

2.1. Cree el punto de montaje `/public` en la máquina servera.

```
[root@servera ~]# mkdir /public
```

2.2. En servera, verifique que la exportación de NFS `/share/public` de serverb se monte correctamente en el directorio `/public`.

```
[root@servera ~]# mount -t nfs \
serverb.lab.example.com:/shares/public /public
```

2.3. Enumere el contenido de la exportación de NFS montado.

```
[root@servera ~]# ls -l /public
total 16
-rw-r--r--. 1 root admin 42 Apr  8 22:36 Delivered.txt
-rw-r--r--. 1 root admin 46 Apr  8 22:36 NOTES.txt
-rw-r--r--. 1 root admin 20 Apr  8 22:36 README.txt
-rw-r--r--. 1 root admin 27 Apr  8 22:36 Trackings.txt
```

2.4. Explore las opciones de comando `mount` para la exportación de NFS montado.

```
[root@servera ~]# mount | grep public
serverb.lab.example.com:/shares/public on /public type nfs4
(rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,
,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.25.250.10,
local_lock=none,addr=172.25.250.11)
```

2.5. Desmonte la exportación de NFS.

```
[root@servera ~]# umount /public
```

► 3. Configure servera para que la exportación `/share/public` se monte de forma persistente.

3.1. Edite el archivo `/etc/fstab`.

```
[root@servera ~]# vim /etc/fstab
```

Agregue la línea siguiente al final del archivo:

```
serverb.lab.example.com:/shares/public /public nfs rw,sync 0 0
```

3.2. Monte el directorio exportado.

```
[root@servera ~]# mount /public
```

3.3. Enumere el contenido del directorio exportado.

```
[root@servera ~]# ls -l /public
total 16
-rw-r--r--. 1 root   admin 42 Apr  8 22:36 Delivered.txt
-rw-r--r--. 1 root   admin 46 Apr  8 22:36 NOTES.txt
-rw-r--r--. 1 root   admin 20 Apr  8 22:36 README.txt
-rw-r--r--. 1 root   admin 27 Apr  8 22:36 Trackings.txt
```

3.4. Reinicie la máquina servera.

```
[root@servera ~]# systemctl reboot
```

- 4. Una vez que servera haya finalizado el reinicio, inicie sesión en servera con el usuario admin1 y pruebe la exportación de NFS montado de forma persistente.

4.1. Inicie sesión en servera con el usuario admin1.

```
[student@workstation ~]$ ssh admin1@servera
[admin1@servera ~]$
```

4.2. Pruebe la exportación de NFS que está montada en el directorio /public.

```
[admin1@servera ~]$ ls -l /public
total 16
-rw-r--r--. 1 root   admin 42 Apr  8 22:36 Delivered.txt
-rw-r--r--. 1 root   admin 46 Apr  8 22:36 NOTES.txt
-rw-r--r--. 1 root   admin 20 Apr  8 22:36 README.txt
-rw-r--r--. 1 root   admin 27 Apr  8 22:36 Trackings.txt
[admin1@servera ~]$ cat /public/NOTES.txt
###In this file you can log all your notes###
[admin1@servera ~]$ echo "This is a test" > /public/Test.txt
[admin1@servera ~]$ cat /public/Test.txt
This is a test
```

4.3. Regrese a la máquina workstation como el usuario student.

```
[admin1@servera ~]$ exit
logout
Connection to servera closed.
```

Finalizar

En la máquina workstation, cambie al directorio de inicio del usuario student y use el comando lab para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.


```
[student@workstation ~]$ lab finish netstorage-nfs
```

Esto concluye la sección.

Montaje automático de almacenamiento conectado a la red

Objetivos

Describir los beneficios de usar el servicio de automontaje y las exportaciones de NFS de automontaje mediante el uso de asignaciones directas e indirectas.

Montaje de exportaciones de NFS con el servicio de automontaje

El *automontaje* es un servicio (`autofs`) que monta automáticamente los sistemas de archivos y las exportaciones de NFS a pedido, y desmonta automáticamente los sistemas de archivos y las exportaciones de NFS cuando los recursos montados ya no están en uso.

La función de automontaje se creó para resolver el problema de que los usuarios sin privilegios no tienen permisos suficientes para usar el comando `mount`. Sin el uso del comando `mount`, los usuarios normales no pueden acceder a medios extraíbles como CD, DVD y unidades de disco extraíbles. Además, si un sistema de archivos local o remoto no se monta en el momento del arranque mediante la configuración `/etc/fstab`, un usuario normal no puede montar y acceder a esos sistemas de archivos desmontados.

Los archivos de configuración del servicio de automontaje se completan con información de montaje del sistema de archivos, de manera similar a las entradas `/etc/fstab`. Aunque los sistemas de archivos `/etc/fstab` se montan durante el arranque del sistema y permanecen montados hasta el apagado del sistema u otra intervención, los sistemas de archivos de automontaje no necesariamente se montan durante el arranque del sistema. En cambio, los sistemas de archivos controlados por el servicio de automontaje se montan a pedido, cuando un usuario o una aplicación intenta ingresar al punto de montaje del sistema de archivos para acceder a los archivos.

Beneficios del servicio de automontaje

El uso de recursos para los sistemas de archivos de automontaje es equivalente a los sistemas de archivos que se montan en el arranque, ya que un sistema de archivos usa recursos solo cuando un programa lee y escribe archivos abiertos. Los sistemas de archivos montados pero inactivos y los sistemas de archivos desmontados usan la misma cantidad de recursos: casi ninguno.

La ventaja del automontaje es que al desmontar el sistema de archivos cada vez que ya no está en uso, el sistema de archivos está protegido de daños inesperados mientras está abierto. Cuando se indica al sistema de archivos que se monte nuevamente, el servicio `autofs` usa la configuración de montaje más actual, a diferencia de un montaje `/etc/fstab`, que aún puede usar una configuración que se montó hace meses durante el último arranque del sistema. Además, si la configuración del servidor NFS incluye servidores y rutas redundantes, el servicio de automontaje puede seleccionar la conexión más rápida cada vez que se solicita un nuevo sistema de archivos.

Método de servicio `autofs` de automontaje

El servicio `autofs` soporta los mismos sistemas de archivos locales y remotos que en el archivo `/etc/fstab`, incluidos los protocolos de uso compartido de archivos NFS y SMB, y soporta las mismas opciones de montaje específicas del protocolo, incluidos los parámetros de seguridad. Los sistemas de archivos que se montan a través del programa de automontaje están disponibles de

manera predeterminada para todos los usuarios, pero se pueden restringir a través de las opciones de permisos de acceso.

Debido a que el servicio de automontaje es una configuración del lado del cliente que usa los comandos estándares `mount` y `umount` para administrar sistemas de archivos, los sistemas de archivos automontados en uso muestran un comportamiento idéntico a los sistemas de archivos que se montan con `/etc/fstab`. La diferencia es que un sistema de archivos de automontaje permanece desmontado hasta que se accede al punto de montaje, lo que hace que el sistema de archivos se monte inmediatamente y permanezca montado mientras el sistema de archivos está en uso. Cuando se cierran todos los archivos en el sistema de archivos, y todos los usuarios y procesos abandonan el directorio de punto de montaje, el servicio de automontaje desmonta el sistema de archivos después de un tiempo de espera mínimo.

Casos de uso de asignaciones directas e indirectas

El servicio de automontaje soporta la asignación directa e indirecta de puntos de montaje, para gestionar los dos tipos de montaje bajo demanda. Un montaje *directo* es cuando un sistema de archivos se monta en una ubicación de punto de montaje conocida que no cambia. Casi todos los montajes del sistema de archivos que configuró, antes de aprender sobre el automontaje, son ejemplos de montajes directos. Un punto de montaje *directo* existe como un directorio permanente, al igual que otros directorios normales.

Un montaje *indirecto* se produce cuando no se conoce la ubicación del punto de montaje hasta que se produce la demanda de montaje. Un ejemplo de montaje indirecto es la configuración para directorios de inicio montados de forma remota, donde el directorio de inicio de un usuario incluye su nombre de usuario en la ruta del directorio. El sistema de archivos remoto del usuario se monta en su directorio de inicio, solo después de que el programa de automontaje sepa qué usuario ha especificado montar su directorio de inicio y determina la ubicación del punto de montaje que se debe usar. Aunque los puntos de montaje *indirectos* parecen existir, el servicio `autofs` los crea cuando se produce la demanda de montaje y los elimina nuevamente cuando la demanda ha finalizado y se desmonta el sistema de archivos.

Configuración del servicio de automontaje

El proceso para configurar un automontaje tiene muchos pasos.

Primero debe instalar los paquetes `autofs` y `nfs-utils`.

```
[user@host ~]$ sudo dnf install autofs nfs-utils
```

Estos paquetes contienen todos los requisitos para usar el servicio de automontaje para las exportaciones de NFS.

Creación de una asignación maestra

A continuación, agregue un archivo de asignación maestra en `/etc/auto.master.d`. Este archivo identifica el directorio de base para puntos de montaje e identifica el archivo de asignación para crear los automontajes.

```
[user@host ~]$ sudo vim /etc/auto.master.d/demo.autofs
```

El nombre del archivo de asignación maestra es principalmente arbitrario (aunque suele tener un sentido) y debe tener una extensión de `.autofs` para que el subsistema lo reconozca. Puede colocar varias entradas en un solo archivo de asignación maestra; como alternativa, puede crear

varios archivos de asignación maestra, cada uno con sus propias entradas agrupadas de forma lógica.

Agregue la entrada de asignación maestra, en este caso, para montajes asignados indirectamente:

```
/shares /etc/auto.demo
```

Esta entrada usa el directorio `/shares` como la base para futuros automontajes indirectos. El archivo `/etc/auto.demo` contiene los detalles de montaje. Use un nombre de archivo absoluto. El archivo `auto.demo` debe crearse antes de comenzar el servicio de `autofs`.

Creación de un mapa indirecto

Ahora, cree los archivos de asignación. Cada archivo de asignación identifica el punto de montaje, las opciones de montaje y la ubicación de origen que se montará para un conjunto de automontajes.

```
[user@host ~]$ sudo vim /etc/auto.demo
```

La convención de nomenclatura de archivos de asignación es `/etc/auto.name`, dónde el *nombre* refleja el contenido de la asignación.

```
work -rw, sync serverb:/shares/work
```

El formato de una entrada es *punto de montaje*, *opciones de montaje* y *ubicación de origen*. En este ejemplo, se muestra una entrada de asignación indirecta básica. Las asignaciones directas y las asignaciones indirectas que usan comodines se tratan más adelante en esta sección.

Conocido como la *clave* en las páginas del manual, el *punto de montaje* se crea y elimina automáticamente con el servicio `autofs`. En este caso, el punto de montaje totalmente calificado es `/shares/work` (consulte el archivo de asignación maestra). Los directorios `/shares` y `/shares/work` se crean y se eliminan según sea necesario mediante el servicio `autofs`.

En este ejemplo, el punto de montaje local duplica la estructura del directorio del servidor. Sin embargo, esta duplicación no es necesaria; el punto de montaje local puede tener un nombre arbitrario. El servicio `autofs` no impone (enforces) una estructura de nombres específica en el cliente.

Las opciones de montaje comienzan con un carácter de guión (-) y se separan por comas sin espacios en blanco. Las opciones de montaje de un sistema de archivos disponibles para el montaje manual también están disponibles al realizar el automontaje. En este ejemplo, el servicio de automontaje monta la exportación con acceso de lectura y escritura (opción `rw`) y el servidor se sincroniza inmediatamente durante las operaciones de escritura (opción `sync`).

Algunas opciones útiles específicas del servicio de automontaje son `-fstype=y` y `-strict`. Use `fstype` para especificar el tipo de sistema de archivos, por ejemplo `nfs4` o `xfs`, y use `strict` para tratar errores como graves cuando monte sistemas de archivos.

La ubicación de origen para las exportaciones de NFS sigue el patrón `host:/pathname`; en este ejemplo, `serverb:/shares/work`. Para que este automontaje se realice correctamente, el servidor NFS, `serverb`, debe *exportar* el directorio con acceso de lectura y escritura, y el usuario que solicita el acceso debe tener permisos de archivo estándar de Linux en el directorio. Si `serverb` exporta el directorio con acceso de solo lectura, entonces el cliente obtendrá acceso de solo lectura a pesar de que solicitó acceso de lectura y escritura.

Comodines en una asignación indirecta

Cuando un servidor NFS exporta varios subdirectorios dentro de un directorio, el servicio de automontaje se puede configurar para acceder a cualquiera de esos subdirectorios usando una única entrada de asignación.

Para continuar con el ejemplo anterior, si `serverb:/shares` exporta dos o más subdirectorios y se puede acceder a estos con las mismas opciones de montaje, el contenido del archivo `/etc/auto.demo` podría verse del siguiente modo:

```
* -rw, sync serverb:/shares/&
```

El punto de montaje (o clave) es un carácter de asterisco (*) y el subdirectorio en la ubicación de origen es el carácter de ampersand (&). Todo lo demás en la entrada es igual.

Cuando un usuario intenta acceder a `/shares/work`, la clave * (que es `work` en este ejemplo) reemplaza el ampersand en la ubicación de origen y se monta `serverb:/exports/work`. Al igual que con el ejemplo indirecto, el servicio `autofs` crea y elimina automáticamente el directorio `work`.

Creación de una asignación indirecta

Una asignación indirecta se usa para asignar una exportación de NFS a un punto de montaje de ruta absoluta. Solo se necesita un archivo de asignación directa, que puede contener cualquier cantidad de asignaciones directas.

Para usar puntos de montaje asignados directamente, el archivo de asignación maestra puede tener la siguiente apariencia:

```
/- /etc/auto.direct
```

Todas las entradas de asignación directa usan `/-` como el directorio de base. En este caso, el archivo de asignación que contiene los detalles de montaje es `/etc/auto.direct`.

El contenido del archivo `/etc/auto.direct` puede verse de la siguiente forma:

```
/mnt/docs -rw, sync serverb:/shares/docs
```

El punto de montaje (o clave) es siempre una ruta absoluta. El resto del archivo de asignación usa la misma estructura.

En este ejemplo, el directorio `/mnt` existe y no lo administra el servicio `autofs`. El servicio `autofs` crea y elimina automáticamente el directorio `/mnt/docs` completo.

Inicio del servicio de automontaje

Por último, use el comando `systemctl` para iniciar y habilitar el servicio `autofs`.

```
[user@host ~]$ sudo systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
```

Método systemd.automount alternativo

El daemon `systemd` puede crear automáticamente archivos de unidad para entradas en `/etc/fstab` que incluyen la opción `x-systemd.automount`. Use el comando `systemctl daemon-reload` después de modificar las opciones de montaje de una entrada para generar un nuevo archivo de unidad y, luego, use el comando `systemctl start unit.automount` para habilitar esa configuración de automontaje.

El nombre de la unidad se basa en su ubicación de montaje. Por ejemplo, si el punto de montaje es `/remote/finance`, el archivo de la unidad se denomina `remote-finance.automount`. El daemon `systemd` monta el sistema de archivos cuando se accede inicialmente al directorio `/remote/finance`.

Este método puede ser más sencillo que instalar y configurar el servicio `autofs`. Sin embargo, una unidad `systemd.automount` puede soportar solo puntos de montaje de ruta absoluta, similar a las asignaciones directas `autofs`.



Referencias

Páginas del manual: `autofs(5)`, `automount(8)`, `auto.master(5)`, `mount.nfs(8)` y `systemd.automount(5)`

► Ejercicio Guiado

Montaje automático de almacenamiento conectado a la red

Lista de verificación de rendimiento

En este ejercicio, crea puntos de montaje de asignación directa y de asignación indirecta administrados por automontaje que montan sistemas de archivos NFS.

Resultados

- Instalar los paquetes necesarios para el servicio de automontaje.
- Configurar asignaciones directas e indirectas de automontaje con recursos de un servidor NFSv4 preconfigurado.
- Describir la diferencia entre las asignaciones directas e indirectas de automontaje.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este script de inicio determina si `servera` y `serverb` son accesibles en la red. El script le avisa si no están disponibles. El script de inicio configura `serverb` como servidor NFSv4, configura los permisos y exporta los directorios. El script también crea los usuarios y grupos necesarios en `servera` y `serverb`.

```
[student@workstation ~]$ lab start netstorage-autofs
```

Instrucciones

Un proveedor de servicios de Internet usa un servidor central, `serverb`, para alojar directorios compartidos con documentos importantes que deben estar disponibles a pedido. Cuando los usuarios inician sesión en `servera` necesitan acceso a los directorios compartidos de automontaje.

Características del entorno:

- La máquina `serverb` exporta el directorio `/shares/indirect`, que a su vez contiene los subdirectorios `west`, `central` y `east`.
- La máquina `serverb` también exporta el directorio `/shares/direct/external`.
- El grupo `operators` está compuesto por los usuarios `operator1` y `operator2`. Tienen acceso de lectura y escritura a los directorios exportados `/shares/indirect/west`, `/shares/indirect/central` y `/shares/indirect/east`.
- El grupo `contractors` está compuesto por los usuarios `contractor1` y `contractor2`. Tienen acceso de lectura y escritura al directorio exportado `/shares/direct/external`.
- Los puntos de montaje esperados para `servera` son `/external` y `/internal`.
- El directorio exportado `/shares/direct/external` se monta automáticamente en `servera` con una asignación *directa* en `/external`.
- El directorio exportado `/shares/indirect/west` debe montarse automáticamente en `servera` con una asignación *indirecta* en `/internal/west`.

- El directorio exportado `/shares/indirect/central` debe montarse automáticamente en `servera` con una asignación *indirecta* en `/internal/central`.
- El directorio exportado `/shares/indirect/east` debe montarse automáticamente en `servera` con una asignación *indirecta* en `/internal/east`.
- Todas las contraseñas de usuario están definidas en `redhat`.
- El paquete `nfs-utils` ya esté instalado.

► **1.** Inicie sesión en `servera` e instale los paquetes requeridos.

1.1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

1.2. Instale el paquete `autofs`.

```
[root@servera ~]# dnf install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

► **2.** Configure una asignación indirecta de automontaje en `servera` con los recursos exportados de `serverb`. Cree la asignación directa con los archivos denominados `/etc/auto.master.d/direct.autofs` para la asignación maestra y `/etc/auto.direct` para el archivo de asignación. Use el directorio `/external` como el punto de montaje principal en `servera`.

2.1. Pruebe el servidor NFS y exporte antes de configurar el servicio de automontaje.

```
[root@servera ~]# mount -t nfs \
serverb.lab.example.com:/shares/direct/external /mnt
[root@servera ~]# ls -l /mnt
total 4
-rw-r--r--. 1 root contractors 22 Apr  7 23:15 README.txt
[root@servera ~]# umount /mnt
```

2.2. Cree un archivo de asignación maestra llamado `/etc/auto.master.d/direct.autofs`, inserte el siguiente contenido y guarde los cambios.

```
/- /etc/auto.direct
```

2.3. Cree un archivo de asignación directa llamado `/etc/auto.direct`, inserte el siguiente contenido y guarde los cambios.

```
/external -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/direct/external
```

► **3.** Configure una asignación indirecta de montaje automático en `servera` con los recursos exportados de `serverb`. Cree la asignación indirecta con los archivos denominados

`/etc/auto.master.d/indirect.autofs` para la asignación maestra y `/etc/auto.indirect` para el archivo de asignación. Use el directorio `/internal` como el punto de montaje principal en `servera`.

3.1. Pruebe el servidor NFS y exporte antes de configurar el servicio de automontaje.

```
[root@servera ~]# mount -t nfs \
serverb.lab.example.com:/shares/indirect /mnt
[root@servera ~]# ls -l /mnt
total 0
drwxrws---. 2 root operators 24 Apr  7 23:34 central
drwxrws---. 2 root operators 24 Apr  7 23:34 east
drwxrws---. 2 root operators 24 Apr  7 23:34 west
[root@servera ~]# umount /mnt
```

3.2. Cree un archivo de asignación maestra llamado `/etc/auto.master.d/indirect.autofs`, inserte el siguiente contenido y guarde los cambios.

```
/internal /etc/auto.indirect
```

3.3. Cree un archivo de asignación indirecta llamado `/etc/auto.indirect`, inserte el siguiente contenido y guarde los cambios.

```
* -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/indirect/&
```

- ▶ 4. Inicie el servicio `autofs` en `servera` y habilítelo para que se inicie automáticamente durante el proceso de arranque. Reinicie `servera` para determinar si el servicio `autofs` se inicia automáticamente.

4.1. Inicie y habilite el servicio `autofs` en `servera`.

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
```

4.2. Reinicie la máquina `servera`.

```
[root@servera ~]# systemctl reboot
```

- ▶ 5. Pruebe la asignación directa de automontaje con el usuario `contractor1`. Cuando haya terminado, salga de la sesión de usuario `contractor1` en `servera`.

5.1. Después de que la máquina `servera` haya terminado de arrancar, inicie sesión en `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

5.2. Cambie al usuario `contractor1`.

```
[student@servera ~]$ su - contractor1
Password: redhat
```

5.3. Enumere el punto de montaje `/external`.

```
[contractor1@servera ~]$ ls -l /external
total 4
-rw-r--r--. 1 root contractors 22 Apr  7 23:34 README.txt
```

5.4. Revise el contenido y pruebe el acceso del punto de montaje `/external`.

```
[contractor1@servera ~]$ cat /external/README.txt
###External Folder###
[contractor1@servera ~]$ echo testing-direct > /external/testing.txt
[contractor1@servera ~]$ cat /external/testing.txt
testing-direct
```

5.5. Salga de la sesión de usuario `contractor1`.

```
[contractor1@servera ~]$ exit
logout
[student@servera ~]$
```

- ▶ 6. Pruebe la asignación indirecta de automontaje con el usuario `operator1`. Cuando finalice, cierre sesión de `servera`.

6.1. Cambie al usuario `operator1`.

```
[student@servera ~]$ su - operator1
Password: redhat
```

6.2. Enumere el punto de montaje `/internal`.

```
[operator1@servera ~]$ ls -l /internal
total 0
```



nota

Con una asignación indirecta de automontaje, debe acceder a cada subdirectorio exportado para que se monten. En una asignación directa del servicio de automontaje, después de abrir el punto de montaje asignado, podrá ver y acceder inmediatamente a los subdirectorios y al contenido en el directorio exportado.

6.3. Pruebe el acceso a directorio exportado del servicio de montaje automático `/internal/west`.

```
[operator1@servera ~]$ ls -l /internal/west/
total 4
-rw-r--r--. 1 root operators 18 Apr  7 23:34 README.txt
```

```
[operator1@servera ~]$ cat /internal/west/README.txt
###West Folder###
[operator1@servera ~]$ echo testing-1 > /internal/west/testing-1.txt
[operator1@servera ~]$ cat /internal/west/testing-1.txt
testing-1
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws---. 2 root operators 24 Apr  7 23:34 west
```

- 6.4. Pruebe el acceso a directorio exportado del servicio de montaje automático / internal/central.

```
[operator1@servera ~]$ ls -l /internal/central
total 4
-rw-r--r--. 1 root operators 21 Apr  7 23:34 README.txt
[operator1@servera ~]$ cat /internal/central/README.txt
###Central Folder###
[operator1@servera ~]$ echo testing-2 > /internal/central/testing-2.txt
[operator1@servera ~]$ cat /internal/central/testing-2.txt
testing-2
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws---. 2 root operators 24 Apr  7 23:34 central
drwxrws---. 2 root operators 24 Apr  7 23:34 west
```

- 6.5. Pruebe el acceso a directorio exportado del servicio de montaje automático / internal/east.

```
[operator1@servera ~]$ ls -l /internal/east
total 4
-rw-r--r--. 1 root operators 18 Apr  7 23:34 README.txt
[operator1@servera ~]$ cat /internal/east/README.txt
###East Folder###
[operator1@servera ~]$ echo testing-3 > /internal/east/testing-3.txt
[operator1@servera ~]$ cat /internal/east/testing-3.txt
testing-3
[operator1@servera ~]$ ls -l /internal
total 0
drwxrws---. 2 root operators 24 Apr  7 23:34 central
drwxrws---. 2 root operators 24 Apr  7 23:34 east
drwxrws---. 2 root operators 24 Apr  7 23:34 west
```

- 6.6. Pruebe el acceso a directorio exportado del servicio de montaje automático / external.

```
[operator1@servera ~]$ ls -l /external
ls: cannot open directory '/external': Permission denied
```

- 6.7. Regrese a la máquina workstation como el usuario student.

```
[operator1@servera ~]$ exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netstorage-autofs
```

Esto concluye la sección.

► Trabajo de laboratorio

Acceso al almacenamiento conectado a la red

Lista de verificación de rendimiento

En este trabajo de laboratorio, configurará el servicio de automontaje con una asignación indirecta, usando exportaciones de un servidor NFSv4.

Resultados

- Instalar los paquetes requeridos para configurar el servicio de automontaje.
- Configurar la asignación indirecta del servicio de automontaje con recursos de un servidor NFSv4 preconfigurado.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este script de inicio determina si los sistemas `servera` y `serverb` son accesibles en la red. El script de inicio configura `serverb` como servidor NFSv4, configura los permisos y exporta los directorios. El script también crea los usuarios y grupos necesarios en los sistemas `servera` y `serverb`.

```
[student@workstation ~]$ lab start netstorage-review
```

Instrucciones

Una empresa de soporte de TI usa un servidor central, `serverb`, para alojar algunos directorios exportados en `/shares` para sus grupos y usuarios. Los usuarios deben ser capaces de iniciar sesión y tener sus directorios exportados montados a pedido y listos para usar, en el directorio `/remote` en `servera`.

Características del entorno:

- La máquina `serverb` comparte el directorio `/shares`, que a su vez contiene los subdirectorios `management`, `production` y `operation`.
- El grupo `managers` está compuesto por los usuarios `manager1` y `manager2`. Tienen acceso de lectura y escritura al directorio exportado `/shares/management`.
- El grupo `production` está compuesto por los usuarios `dbuser1` y `sysadmin1`. Tienen acceso de lectura y escritura al directorio exportado `/shares/production`.
- El grupo `operators` está compuesto por los usuarios `contractor1` y `consultant1`. Tienen acceso de lectura y escritura al directorio exportado `/shares/operation`.
- El punto de montaje principal para `servera` es el directorio `/remote`.
- Use el archivo `/etc/auto.master.d/shares.autofs` como el archivo de asignación maestra y el archivo `/etc/auto.shares` como el archivo de asignación indirecta.

- El directorio exportado `/shares/management` debe montarse automáticamente en `/remote/management` en `servera`.
 - El directorio exportado `/shares/production` debe montarse automáticamente en `/remote/production` en `servera`.
 - El directorio exportado `/shares/operation` debe montarse automáticamente en `/remote/operation` en `servera`.
 - Todas las contraseñas de usuario están definidas en `redhat`.
1. Inicie sesión en `servera` e instale los paquetes requeridos.
 2. Configure una asignación indirecta de montaje automático en `servera` con los recursos exportados de `serverb`. Cree una asignación indirecta con los archivos denominados `/etc/auto.master.d/shares.autofs` para la asignación maestra y `/etc/auto.shares` para el archivo de asignación. Use el directorio `/remote` como el punto de montaje principal en `servera`. Reinicie `servera` para determinar si el servicio `autofs` se inicia automáticamente.
 3. Pruebe la configuración de `autofs` con los distintos usuarios. Cuando finalice, cierre sesión de `servera`.

Evaluación

En la máquina `workstation`, use el comando `lab` para confirmar que ha realizado correctamente este ejercicio.

```
[student@workstation ~]$ lab grade netstorage-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netstorage-review
```

Esto concluye la sección.

► Solución

Acceso al almacenamiento conectado a la red

Lista de verificación de rendimiento

En este trabajo de laboratorio, configurará el servicio de automontaje con una asignación indirecta, usando exportaciones de un servidor NFSv4.

Resultados

- Instalar los paquetes requeridos para configurar el servicio de automontaje.
- Configurar la asignación indirecta del servicio de automontaje con recursos de un servidor NFSv4 preconfigurado.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este script de inicio determina si los sistemas `servera` y `serverb` son accesibles en la red. El script de inicio configura `serverb` como servidor NFSv4, configura los permisos y exporta los directorios. El script también crea los usuarios y grupos necesarios en los sistemas `servera` y `serverb`.

```
[student@workstation ~]$ lab start netstorage-review
```

Instrucciones

Una empresa de soporte de TI usa un servidor central, `serverb`, para alojar algunos directorios exportados en `/shares` para sus grupos y usuarios. Los usuarios deben ser capaces de iniciar sesión y tener sus directorios exportados montados a pedido y listos para usar, en el directorio `/remote` en `servera`.

Características del entorno:

- La máquina `serverb` comparte el directorio `/shares`, que a su vez contiene los subdirectorios `management`, `production` y `operation`.
- El grupo `managers` está compuesto por los usuarios `manager1` y `manager2`. Tienen acceso de lectura y escritura al directorio exportado `/shares/management`.
- El grupo `production` está compuesto por los usuarios `dbuser1` y `sysadmin1`. Tienen acceso de lectura y escritura al directorio exportado `/shares/production`.
- El grupo `operators` está compuesto por los usuarios `contractor1` y `consultant1`. Tienen acceso de lectura y escritura al directorio exportado `/shares/operation`.
- El punto de montaje principal para `servera` es el directorio `/remote`.
- Use el archivo `/etc/auto.master.d/shares.autofs` como el archivo de asignación maestra y el archivo `/etc/auto.shares` como el archivo de asignación indirecta.

- El directorio exportado `/shares/management` debe montarse automáticamente en `/remote/management` en `servera`.
 - El directorio exportado `/shares/production` debe montarse automáticamente en `/remote/production` en `servera`.
 - El directorio exportado `/shares/operation` debe montarse automáticamente en `/remote/operation` en `servera`.
 - Todas las contraseñas de usuario están definidas en `redhat`.
1. Inicie sesión en `servera` e instale los paquetes requeridos.
 - 1.1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 1.2. Instale el paquete `autofs`.

```
[root@servera ~]# dnf install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

2. Configure una asignación indirecta de montaje automático en `servera` con los recursos exportados de `serverb`. Cree una asignación indirecta con los archivos denominados `/etc/auto.master.d/shares.autofs` para la asignación maestra y `/etc/auto.shares` para el archivo de asignación. Use el directorio `/remote` como el punto de montaje principal en `servera`. Reinicie `servera` para determinar si el servicio `autofs` se inicia automáticamente.

- 2.1. Pruebe el servidor NFS antes de configurar el servicio de automontaje.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares /mnt
[root@servera ~]# ls -l /mnt
total 0
drwxrwx---. 2 root managers 25 Apr 4 01:13 management
drwxrwx---. 2 root operators 25 Apr 4 01:13 operation
drwxrwx---. 2 root production 25 Apr 4 01:13 production
[root@servera ~]# umount /mnt
```

- 2.2. Cree un archivo de asignación maestra llamado `/etc/auto.master.d/shares.autofs`, inserte el siguiente contenido y guarde los cambios.

```
/remote /etc/auto.shares
```

- 2.3. Cree un archivo de asignación indirecta llamado `/etc/auto.shares`, inserte el siguiente contenido y guarde los cambios.


```
* -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/&
```

2.4. Inicie y habilite el servicio autofs en servera.

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/lib/systemd/system/autofs.service.
```

2.5. Reinicie la máquina servera.

```
[root@servera ~]# systemctl reboot
```

3. Pruebe la configuración de autofs con los distintos usuarios. Cuando finalice, cierre sesión de servera.

3.1. Después de que la máquina servera haya terminado de arrancar, inicie sesión en servera con el usuario student.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

3.2. Cambie al usuario manager1 y pruebe el acceso.

```
[student@servera ~]$ su - manager1
Password: redhat
[manager1@servera ~]$ ls -l /remote/management/
total 4
-rw-r--r--. 1 root managers 46 Apr  4 01:13 Welcome.txt
[manager1@servera ~]$ cat /remote/management/Welcome.txt
###Welcome to Management Folder on SERVERB###
[manager1@servera ~]$ echo TEST1 > /remote/management/Test.txt
[manager1@servera ~]$ cat /remote/management/Test.txt
TEST1
[manager1@servera ~]$ ls -l /remote/operation/
ls: cannot open directory '/remote/operation/': Permission denied
[manager1@servera ~]$ ls -l /remote/production/
ls: cannot open directory '/remote/production/': Permission denied
[manager1@servera ~]$ exit
logout
[student@servera ~]$
```

3.3. Cambie al usuario dbuser1 y pruebe el acceso.

```
[student@servera ~]$ su - dbuser1
Password: redhat
[dbuser1@servera ~]$ ls -l /remote/production/
total 4
-rw-r--r--. 1 root production 46 Apr  4 01:13 Welcome.txt
[dbuser1@servera ~]$ cat /remote/production/Welcome.txt
###Welcome to Production Folder on SERVERB###
```

```
[dbuser1@servera ~]$ echo TEST2 > /remote/production/Test.txt
[dbuser1@servera ~]$ cat /remote/production/Test.txt
TEST2
[dbuser1@servera ~]$ ls -l /remote/operation/
ls: cannot open directory '/remote/operation/': Permission denied
[dbuser1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[dbuser1@servera ~]$ exit
logout
[student@servera ~]$
```

3.4. Cambie al usuario `contractor1` y pruebe el acceso.

```
[student@servera ~]$ su - contractor1
Password: redhat
[contractor1@servera ~]$ ls -l /remote/operation/
total 4
-rw-r--r--. 1 root operators 45 Apr  4 01:13 Welcome.txt
[contractor1@servera ~]$ cat /remote/operation/Welcome.txt
###Welcome to Operation Folder on SERVERB###
[contractor1@servera ~]$ echo TEST3 > /remote/operation/Test.txt
[contractor1@servera ~]$ cat /remote/operation/Test.txt
TEST3
[contractor1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[contractor1@servera ~]$ ls -l /remote/production/
ls: cannot open directory '/remote/production/': Permission denied
[contractor1@servera ~]$ exit
logout
[student@servera ~]$
```

3.5. Explore las opciones de `mount` para la exportación de NFS de automontaje.

```
[student@servera ~]$ mount | grep nfs
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
serverb.lab.example.com:/shares/management on /remote/management type nfs4
(rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,
sync,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.25.250.10,
local_lock=none,addr=172.25.250.11)
serverb.lab.example.com:/shares/operation on /remote/operation type nfs4
(rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,
sync,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.25.250.10,
local_lock=none,addr=172.25.250.11)
serverb.lab.example.com:/shares/production on /remote/production type nfs4
(rw,relatime,vers=4.2,rsz=262144,wsz=262144,namlen=255,
sync,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.25.250.10,
local_lock=none,addr=172.25.250.11)
```

3.6. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
[student@workstation ~]$
```

Evaluación

En la máquina `workstation`, use el comando `lab` para confirmar que ha realizado correctamente este ejercicio.

```
[student@workstation ~]$ lab grade netstorage-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netstorage-review
```

Esto concluye la sección.

Resumen

- Montar y desmontar un recurso compartido de NFS desde la línea de comandos.
- Configurar un recurso compartido de NFS para montarse automáticamente en el inicio.
- Configurar el servicio de automontaje con asignaciones directas e indirectas, así como describir sus diferencias.

capítulo 8

Control del proceso de arranque

Meta

Administrar el proceso de arranque para controlar los servicios ofrecidos y para solucionar y reparar problemas.

Objetivos

- Describir el proceso de arranque de Red Hat Enterprise Linux, configurar el objetivo predeterminado que se usa en el arranque e iniciar un sistema con un objetivo no predeterminado.
- Iniciar sesión en un sistema y cambiar la contraseña de root cuando la actual se haya perdido.
- Reparar manualmente la configuración del sistema de archivos o problemas de daños que detengan el proceso de arranque.

Secciones

- Selección del objetivo de arranque (y ejercicio guiado)
- Restablecimiento de la contraseña de root (y ejercicio guiado)
- Reparación de problemas de sistemas de archivos en el arranque (y ejercicio guiado)

Trabajo de laboratorio

Control del proceso de arranque

Selección del objetivo de arranque

Objetivos

Describir el proceso de arranque de Red Hat Enterprise Linux, configurar el objetivo predeterminado que se usa en el arranque e iniciar un sistema con un objetivo no predeterminado.

Descripción del proceso de arranque de Red Hat Enterprise Linux 9

Los sistemas de computación modernos son combinaciones complejas de hardware y software. Desde un estado de apagado no definido hasta un sistema de ejecución con un prompt de inicio de sesión, se requieren muchas piezas de hardware y software que funcionen en conjunto. En la siguiente lista, se proporciona una descripción general de alto nivel de las tareas de arranque de un sistema físico x86_64 de Red Hat Enterprise Linux 9. La lista de máquinas virtuales x86_64 es prácticamente la misma, pero el hipervisor maneja algunos de los pasos específicos del hardware en el software.

- La máquina se enciende. El firmware del sistema (UEFI moderno o BIOS anterior) ejecuta una *prueba automática de encendido* (*Power On Self Test, POST*) y comienza a inicializar el hardware.

Se ajusta por medio de las pantallas de configuración de BIOS o UEFI del sistema, a las cuales se llega típicamente al presionar una determinada combinación de teclas (p. ej., F2) al principio del proceso de arranque.

- El firmware del sistema busca un dispositivo con capacidad de arranque, ya sea configurado en el firmware de arranque UEFI o al buscar un *registro de arranque maestro* (*Master Boot Record, MBR*) en todos los discos, en el orden configurado en el BIOS o UEFI.

Se ajusta por medio de las pantallas de configuración de BIOS o UEFI del sistema, a las cuales se llega típicamente al presionar una determinada combinación de teclas (p. ej., F2) al principio del proceso de arranque.

- El firmware del sistema lee un cargador de arranque desde el disco, luego pasa el control del sistema al cargador de arranque. En un sistema de Red Hat Enterprise Linux 9, el cargador de arranque es *GRand Unified Bootloader versión 2* (*GRUB2*).

Se configura con el comando `grub2-install`, el cual instala GRUB2 como el cargador de arranque en el disco para sistemas BIOS. No use el comando `grub2-install` directamente para instalar el cargador de arranque de UEFI. RHEL 9 proporciona un archivo `/boot/efi/EFI/redhat/grubx64.efi` precompilado, que contiene las firmas de autenticación requeridas para un sistema de arranque seguro. La ejecución de `grub2-install` directamente en un sistema UEFI genera un nuevo archivo `grubx64.efi` sin las firmas requeridas. Puede restaurar el archivo `grubx64.efi` correcto del paquete `grub2-efi`.

- GRUB2 carga su configuración desde el archivo `/boot/grub2/grub.cfg` para BIOS y desde el archivo `/boot/efi/EFI/redhat/grub.cfg` para UEFI y muestra un menú donde puede seleccionar qué kernel arrancar.

Se configura por medio del directorio `/etc/grub.d/` y el archivo `/etc/default/grub`; el comando `grub2-mkconfig` genera los archivos `/boot/grub2/grub.cfg` o `/boot/efi/EFI/redhat/grub.cfg` para BIOS o UEFI respectivamente.

- Después de seleccionar un kernel, o después de que el tiempo de espera finaliza, el cargador de arranque carga el kernel e `initramfs` desde el disco y los coloca en la memoria. Un `initramfs` es una colección de archivos que contiene módulos del kernel para todo el hardware necesario en el arranque, scripts de inicialización y más. En Red Hat Enterprise Linux 9, `initramfs` contiene un sistema totalmente usable por sí solo.

Se configura por medio del directorio `/etc/dracut.conf.d/`, el comando `dracut` y el comando `lsinitrd` para inspeccionar el archivo `initramfs`.

- El cargador de arranque pasa el control al kernel, y detalla todas las opciones especificadas en la línea de comandos del kernel en el cargador de arranque, y la ubicación del `initramfs` en la memoria.

Se configura por medio del directorio `/etc/grub.d/`, el archivo `/etc/default/grub` y el comando `grub2-mkconfig` para generar el archivo `/boot/grub2/grub.cfg`.

- El kernel inicializa todo el hardware para el que puede encontrar un controlador en el `initramfs` y, luego, ejecuta `/sbin/init` desde `initramfs` como PID 1. En Red Hat Enterprise Linux 9, `/sbin/init` es un enlace a `systemd`.

Se configura por medio del parámetro de la línea de comandos `init=` del kernel.

- La instancia `systemd` del `initramfs` ejecuta todas las unidades para el objetivo `initrd.target`. Esto incluye el montaje del sistema de archivos root en el disco en el directorio `/sysroot`.

Se configura con el archivo `/etc/fstab`.

- El kernel cambia (articula) el sistema de archivos root desde `initramfs` al sistema de archivos root en el directorio `/sysroot`. A continuación, `systemd` vuelve a ejecutarse usando la copia de `systemd` instalada en el disco.
- `systemd` busca un objetivo predeterminado, ya sea especificado desde la línea de comandos del kernel o configurado en el sistema, luego inicia (y detiene) unidades para cumplir con la configuración para ese objetivo, y resuelve dependencias entre unidades automáticamente. Básicamente, un objetivo `systemd` es un conjunto de unidades que el sistema debe activar para alcanzar un estado deseado. Por lo general, estos objetivos inician una pantalla de inicio de sesión basado en texto o inicio de sesión gráfico.

Se configura con el archivo `/etc/systemd/system/default.target` y el directorio `/etc/systemd/system/`.

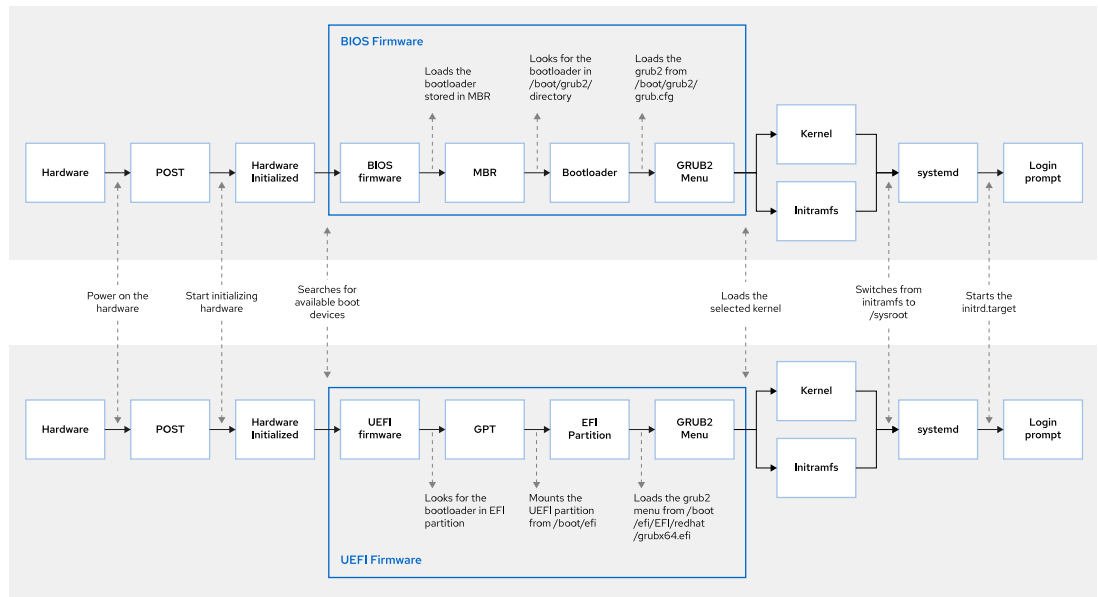


Figura 8.1: Proceso de arranque para sistemas basados en BIOS y UEFI

Reinicio y apagado

Para apagar o reiniciar un sistema en ejecución desde la línea de comandos, puede usar el comando `systemctl`.

El comando `systemctl poweroff` detiene todos los servicios en ejecución, desmonta todos los sistemas de archivos (o vuelve a montarlos como solo lectura cuando no se puedan desmontar) y, luego, apaga el sistema.

El comando `systemctl reboot` detiene todos los servicios en ejecución, desmonta todos los sistemas de archivos y, luego, reinicia el sistema.

También puede usar el atajo de estos comandos, `poweroff` y `reboot`, que son enlaces simbólicos a sus equivalentes de `systemctl`.



nota Los comandos `systemctl halt` y `halt` también están disponibles para detener el sistema, pero a diferencia del comando `poweroff`, estos comandos no apagan el sistema, sino que lo llevan hasta un punto donde es seguro apagarlo manualmente.

Selección de un objetivo de Systemd

Un objetivo de `systemd` es un conjunto de unidades de `systemd` que el sistema debe iniciar para alcanzar un estado deseado. En la siguiente tabla, se detallan los objetivos más importantes.

Objetivos que se usan normalmente

Objetivo	Propósito
<code>graphical.target</code>	El sistema soporta varios usuarios, e inicios de sesión gráficos y basados en texto.

Objetivo	Propósito
<code>multi-user.target</code>	El sistema soporta varios usuarios y solo inicios de sesión basados en texto.
<code>rescue.target</code>	Prompt <code>su login</code> , inicialización del sistema básica finalizada.
<code>emergency.target</code>	Prompt <code>su login</code> , cambio de <code>initramfs</code> completo y root de sistema montado en <code>/</code> con acceso de solo lectura.

Un objetivo puede ser parte de otro objetivo. Por ejemplo, `graphical.target` incluye `multi-user.target`, que a su vez depende de `basic.target` y otros. Puede ver estas dependencias con el siguiente comando.

```
[user@host ~]$ systemctl list-dependencies graphical.target | grep target
graphical.target
* └─multi-user.target
*   └─basic.target
*     └─paths.target
*     └─slices.target
*     └─sockets.target
*     └─sysinit.target
*       └─cryptsetup.target
*       └─integritysetup.target
*       └─local-fs.target
...output omitted...
```

Para enumerar los objetivos disponibles, use el siguiente comando.

```
[user@host ~]$ systemctl list-units --type=target --all
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
-----
basic.target                        loaded active active Basic System
...output omitted...
cloud-config.target                loaded active active Cloud-config availability
cloud-init.target                  loaded active active Cloud-init target
cryptsetup-pre.target              loaded inactive dead Local Encrypted Volumes
(Pre)
cryptsetup.target                  loaded active active Local Encrypted Volumes
...output omitted...
```

Selección de un objetivo en el tiempo de ejecución

En un sistema en ejecución, los administradores pueden cambiar a un objetivo diferente con el comando `systemctl isolate`.

```
[root@host ~]# systemctl isolate multi-user.target
```

Si aísla un objetivo, detiene todos los servicios no requeridos por ese objetivo (y sus dependencias) e inicia todos los servicios requeridos que aún no se hayan iniciado.

No todos los objetivos se pueden aislar. Solo puede aislar objetivos que tengan `AllowIsolate=yes` establecido en sus archivos de unidad. Por ejemplo, puede aislar el objetivo gráfico (graphical target), pero no el objetivo `cryptsetup`.

```
[user@host ~]$ systemctl cat graphical.target
# /usr/lib/systemd/system/graphical.target
...output omitted...
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes
[user@host ~]$ systemctl cat cryptsetup.target
# /usr/lib/systemd/system/cryptsetup.target
...output omitted...
[Unit]
Description=Local Encrypted Volumes
Documentation=man:systemd.special(7)
```

Configuración de un objetivo predeterminado

Cuando el sistema se inicia, `systemd` activa el objetivo `default.target`. Normalmente, el objetivo predeterminado en `/etc/systemd/system/` es un enlace simbólico a los objetivos `graphical.target` o `multi-user.target`. En lugar de editar este enlace simbólico manualmente, el comando `systemctl` viene con dos subcomandos para administrar este enlace: `get-default` y `set-default`.

```
[root@host ~]# systemctl get-default
multi-user.target
[root@host ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/graphical.target.
[root@host ~]# systemctl get-default
graphical.target
```

Selección de un objetivo diferente en el momento del arranque

Para seleccionar un objetivo diferente en el momento del arranque, agregue la opción `systemd.unit=target.target` a la línea de comandos del kernel desde el cargador de arranque.

Por ejemplo, para iniciar el sistema en una shell de recuperación donde pueda cambiar la configuración del sistema sin casi ningún servicio en ejecución, agregue la siguiente opción a la línea de comandos del kernel desde el cargador de arranque.

```
systemd.unit=rescue.target
```

Este cambio de configuración solo afecta a un único arranque, lo que hace que sea una herramienta útil para la solución de problemas en el proceso de arranque.

Para usar este método de selección de un objetivo diferente, use el siguiente procedimiento:

1. Arranque o reinicie el sistema.
2. Interrumpa la cuenta regresiva del menú del cargador de arranque presionando cualquier tecla (excepto **Enter** que iniciaría un arranque normal).
3. Mueva el cursor hasta la entrada del kernel que desea iniciar.
4. Presione **e** para editar la entrada actual.
5. Mueva el cursor hasta la línea que comienza con **linux**. Esta es la línea de comandos del kernel.
6. Anexe `systemd.unit=target.target`. Por ejemplo, `systemd.unit=emergency.target`.
7. Presione **Ctrl+x** para realizar el arranque con estos cambios.



Referencias

`info grub2` (*GNU GRUB manual*)

Páginas del manual: `bootup(7)`, `dracut.bootup(7)`, `lsinitrd(1)`, `systemd.target(5)`, `systemd.special(7)`, `sulogin(8)` y `systemctl(1)`

Para obtener más información, consulte el capítulo *Managing services with systemd* de la guía *Configuring basic system settings* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/configuring_basic_system_settings/index#managing-services-with-systemd

► Ejercicio Guiado

Selección del objetivo de arranque

En este ejercicio, determinará el objetivo predeterminado en el que se inicia un sistema y arrancará ese sistema en otros objetivos.

Resultados

- Actualizar el objetivo predeterminado del sistema y usar un objetivo temporal desde el cargador de arranque.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start boot-selecting
```

Instrucciones

1. En la máquina `workstation`, abra un terminal y confirme que el objetivo predeterminado sea `graphical.target`.

```
[student@workstation ~]$ systemctl get-default  
graphical.target
```

2. En la máquina `workstation`, cambie al objetivo `multi-user` manualmente sin reiniciar. Use el comando `sudo` y, si se le solicita, use `student` como la contraseña.

```
[student@workstation ~]$ sudo systemctl isolate multi-user.target  
[sudo] password for student: student
```

3. Acceda a una consola basada en texto. Use la secuencia de teclas `Ctrl+Alt+F1` por medio de la entrada del menú o el botón relevantes. Inicie sesión con el usuario `root` con la contraseña `redhat`.



nota

Recordatorio: si está usando el terminal a través de una página web, puede hacer clic en el ícono Show Keyboard (Mostrar teclado) debajo de la barra de URL de su navegador web y luego a la derecha de la dirección IP de la máquina.

```
workstation login: root
Password: redhat
[root@workstation ~]#
```

- 4. Configure la máquina `workstation` para que inicie automáticamente en el objetivo `multi-user` y, luego, reinicie la máquina `workstation` para verificarlo. Cuando haya finalizado, cambie el objetivo `systemd` predeterminado al objetivo `graphical` nuevamente.

4.1. Establezca el objetivo predeterminado.

```
[root@workstation ~]# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target.
```

4.2. Reinicie la máquina `workstation`. Después del reinicio, el sistema presenta una consola basada en texto, no una pantalla de inicio de sesión gráfico.

```
[root@workstation ~]# systemctl reboot
```

4.3. Inicie sesión con el usuario `root`.

```
workstation login: root
Password: redhat
Last login: Thu Mar 28 14:50:53 on tty1
[root@workstation ~]#
```

4.4. Establezca el objetivo `systemd` predeterminado nuevamente en el objetivo `graphical`.

```
[root@workstation ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/graphical.target.
```

4.5. Esto concluye la primera parte del ejercicio donde practica la configuración del objetivo `systemd` predeterminado.

- 5. En esta segunda parte del ejercicio, practicará el modo de recuperación para recuperar el sistema.

Reinicie `workstation` nuevamente para acceder al cargador de arranque. Desde el menú del cargador de arranque, arranque en el objetivo `rescue` (recuperación).

5.1. Active el reinicio.

```
[root@workstation ~]# systemctl reboot
```

- 5.2. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla (excepto **Enter** que iniciaría un arranque normal) para interrumpir la cuenta regresiva.
- 5.3. Use las teclas de dirección para destacar la entrada del cargador de arranque predeterminada.
- 5.4. Presione **e** para editar la entrada actual.
- 5.5. Con las teclas de dirección, navegue hacia la línea que comienza con **linux**.
- 5.6. Presione **End** (Fin) para mover el cursor hasta el final de la línea.
- 5.7. Agregue **systemd.unit=rescue.target** en el final de la línea.
- 5.8. Presione **Ctrl+x** para realizar el arranque con la configuración modificada.
- 5.9. Inicie sesión en el modo de recuperación. Es posible que tenga que presionar **Enter** (Intro) para obtener un prompt limpio.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@workstation ~]#
```

- ▶ 6. Confirme que, en el modo de recuperación, el sistema de archivos root se encuentre en el modo de lectura y escritura.

```
[root@workstation ~]# mount
...output omitted...
/dev/vda4 on / type xfs
(rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
...output omitted...
```

- ▶ 7. Presione **Ctrl+d** para continuar con el proceso de arranque.
El sistema presenta un inicio de sesión gráfico. Inicie sesión con el usuario **student**.

Finalizar

En la máquina **workstation**, cambie al directorio de inicio del usuario **student** y use el comando **lab** para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish boot-selecting
```

Esto concluye la sección.

Restablecimiento de la contraseña de root

Objetivos

Iniciar sesión en un sistema y cambiar la contraseña de root cuando la actual se haya perdido.

Restablecimiento de la contraseña de root desde el cargador de arranque.

Una tarea que todos los administradores de sistemas deben ser capaces de realizar es restablecer una contraseña `root` perdida. Si el administrador aún tiene la sesión iniciada, ya sea como usuario no privilegiado, pero con acceso `sudo` completo, o como `root`, esta tarea es sencilla. Esta tarea es un poco más complicada cuando el administrador no ha iniciado sesión.

Existen muchos métodos para establecer una nueva contraseña `root`. Un administrador de sistemas podría, por ejemplo, arrancar el sistema usando un Live CD, montar el sistema de archivos `root` desde allí y editar `/etc/shadow`. En esta sección, exploramos un método que no requiere el uso de medios externos.



nota

En Red Hat Enterprise Linux 6 y en las versiones anteriores, los administradores pueden arrancar el sistema en el nivel de ejecución 1 para obtener un prompt de `root`. Los análogos más cercanos al nivel de ejecución 1 en una máquina Red Hat Enterprise Linux 8 o posterior son los objetivos de recuperación y emergencia y ambos requieren la contraseña `root` para iniciar sesión.

Si su sistema se implementó desde una imagen de nube de Red Hat, no tendrá un kernel de rescate en su menú de arranque; sin embargo, su kernel predeterminado tendrá un comportamiento similar que le permitirá ingresar al modo de mantenimiento sin la contraseña de root.

En Red Hat Enterprise Linux 9, es posible tener scripts que se ejecuten desde la pausa de `initramfs` en ciertos puntos, proporcionar una shell `root` y, luego, continuar cuando esa shell se cierre. Esto se suele realizar principalmente para depuraciones, pero también se puede usar para restablecer una contraseña `root` perdida.

A partir de Red Hat Enterprise Linux 9, si instala su sistema desde un DVD, el kernel predeterminado solicita la contraseña `root` cuando intenta ingresar al modo de mantenimiento. Por lo tanto, para restablecer una contraseña `root` perdida, debe usar el kernel de rescate. No obstante, si implementa su sistema desde una imagen de nube de Red Hat, no tiene un kernel de rescate en su menú de arranque, pero su kernel predeterminado presenta un comportamiento similar que le permite ingresar al modo de mantenimiento sin la contraseña `root`.

Para acceder a la shell `root`, siga estos pasos:

1. Reinicie el sistema.
2. Interrumpa la cuenta regresiva del cargador de arranque presionando cualquier tecla (excepto `Enter`).

3. Mueva el cursor a la entrada del kernel de rescate para arrancar (la que tiene la palabra *rescue* en su nombre).
4. Presione **e** para editar la entrada seleccionada.
5. Mueva el cursor hasta la línea de comandos del kernel (la línea que empieza con `linux`).
6. Anexe `rd.break`. Con esa opción, se produce un quiebre en el sistema antes de que el control se entregue de `initramfs` al sistema real.
7. Presione **Ctrl+x** para realizar el arranque con los cambios.
8. Presione **Enter** para realizar el mantenimiento cuando se le solicite.

En este punto, se presenta una shell `root`, con el sistema de archivos `root` real en el disco montado para solo lectura en `/sysroot`. Debido a que la solución de problemas suele exigir modificaciones en el sistema de archivos `root`, debe volver a montar el sistema de archivos `root` para que su acceso sea de lectura y escritura. En el siguiente paso, se muestra cómo la opción `remount, rw` del comando `mount` vuelve a montar el sistema de archivos con la opción nueva (`rw`) configurada.

**nota**

Es posible que las imágenes compiladas previamente coloquen múltiples argumentos `console=` en el kernel para respaldar una amplia variedad de situaciones de implementación. Estos argumentos `console=` indican los dispositivos que se deben usar para la salida por consola. La advertencia con `rd.break` es que si bien el sistema envía los mensajes del kernel a todas las consolas, el prompt usará en última instancia la última consola. Si no recibe el prompt, es posible que desee reordenar temporalmente los argumentos `console=` cuando edite la línea de comandos del kernel desde el cargador de arranque.

**Importante**

El sistema aún no ha habilitado SELinux, por lo que cualquier archivo que cree no tiene contexto SELinux. Algunas herramientas, como el comando `passwd`, primero crean un archivo temporal, y, luego, lo reemplazan con el archivo que intentan editar; y se crea así de manera efectiva un nuevo archivo sin un contexto SELinux. Por este motivo, cuando usa el comando `passwd` con `rd.break`, el archivo `/etc/shadow` no recibe un contexto SELinux.

Para restablecer la contraseña `root` desde este punto, realice el siguiente procedimiento:

1. Vuelva a montar `/sysroot` con acceso de lectura y escritura.

```
sh-5.1# mount -o remount,rw /sysroot
```

2. Cambie a jail de `chroot`, donde `/sysroot` se trata como la `root` de un árbol de sistemas de archivos.

```
sh-5.1# chroot /sysroot
```

3. Establezca una nueva contraseña `root`.


```
sh-5.1# passwd root
```

4. Asegúrese de que todos los archivos no etiquetados, incluido `/etc/shadow` en este punto, obtengan una nueva etiqueta durante el arranque.

```
sh-5.1# touch /.autorelabel
```

5. Ingrese `exit` dos veces. El primer comando saldrá del jail de `chroot` y el segundo comando saldrá de la shell de depuración de `initramfs`.

En este punto, el sistema continúa con el arranque, realiza un nuevo etiquetado de SELinux completo y, luego, realiza el arranque nuevamente.

Inspección de registros

Puede ser útil mirar los registros de arranques anteriores que fallaron. Si los diarios (journals) del sistema son persistentes durante los reinicios, puede usar la herramienta `journalctl` para inspeccionar esos registros.

Recuerde que, de manera predeterminada, los diarios (journals) del sistema se almacenan en el directorio `/run/log/journal`, lo que significa que se borran cuando se reinicia el sistema. Para almacenar diarios (journals) en el directorio `/var/log/journal`, que no se borra en los reinicios, configure el parámetro `Storage` en `persistent` en `/etc/systemd/journald.conf`.

```
[root@host ~]# vim /etc/systemd/journald.conf
...output omitted...
[Journal]
Storage=persistent
...output omitted...
[root@host ~]# systemctl restart systemd-journald.service
```

Para inspeccionar los registros de un arranque anterior, use la opción `-b` del comando `journalctl`. Sin ningún argumento, la opción `-b` del comando `journalctl` solo muestra mensajes desde el último arranque. Si el argumento es un número negativo, se muestran los registros de los arranques anteriores.

```
[root@host ~]# journalctl -b -1 -p err
```

Este comando muestra todos los mensajes calificados como error o peor del arranque anterior.

Reparación de problemas de arranque de Systemd

Para solucionar problemas de inicio del servicio en el momento del arranque, Red Hat Enterprise Linux 8 y posteriores cuentan con las siguientes herramientas.

Habilitación de la shell de depuración temprana

Al habilitar el servicio `debug-shell` con `systemctl enable debug-shell.service`, el sistema genera una shell `root` en `TTY9` (`Ctrl+Alt+F9`) al principio de la secuencia de arranque. Esta shell inicia sesión automáticamente como `root`, de modo que los administradores pueden depurar el sistema mientras el sistema operativo aún está arrancando.

**Advertencia**

No olvide deshabilitar el servicio `debug-shell.service` cuando haya finalizado la depuración, ya que deja una shell `root` no autenticada abierta a cualquier usuario con acceso a la consola local.

O bien puede seguir estos pasos para activar la shell de depuración durante el arranque con el menú de GRUB2:

1. Reinicie el sistema.
2. Interrumpa la cuenta regresiva del cargador de arranque presionando cualquier tecla (excepto `Enter`).
3. Mueva el cursor hasta la entrada del kernel que debe iniciarse.
4. Presione `e` para editar la entrada seleccionada.
5. Mueva el cursor hasta la línea de comandos del kernel (la línea que empieza con `linux`).
6. Anexe `systemd.debug-shell`. Con este parámetro, el sistema arranca en la shell de depuración.
7. Presione `Ctrl+x` para realizar el arranque con los cambios.

Uso de los objetivos de emergencia y recuperación

Al agregar `systemd.unit=rescue.target` o `systemd.unit=emergency.target` delante de la línea de comandos del kernel desde el cargador de arranque, el sistema indica una shell de emergencia o recuperación en lugar de iniciarse normalmente. Estas dos shells requieren la contraseña `root`.

El objetivo de emergencia mantiene el sistema de archivos `root` montado con solo lectura; mientras que el objetivo de recuperación espera que `sysinit.target` se complete, para que una mayor parte del sistema se inicie (p. ej., servicio de inicio de sesión o sistemas de archivos). El usuario `root` en este punto no puede hacer cambios a `/etc/fstab` hasta que la unidad se vuelva a montar en un estado de lectura y escritura con el comando `mount -o remount, rw /`.

Los administradores pueden usar estas shells para corregir problemas que impiden que el sistema arranque normalmente; por ejemplo, un bucle de dependencia entre servicios o una entrada incorrecta en `/etc/fstab`. Cuando se sale de estas shells, continúa el proceso de arranque regular.

Identificación de trabajos atascados

Durante el inicio, `systemd` inicia una cantidad de trabajos. Si alguno de estos trabajos no se puede completar, impide que se ejecuten otros trabajos. Para inspeccionar la lista de trabajos actual, los administradores pueden usar el comando `systemctl list-jobs`. Todos los trabajos detallados como en ejecución deben completarse para que los trabajos detallados como en espera puedan continuar.



Referencias

Páginas del manual: `dracut.cmdline(7)`, `systemd-journald(8)`, `journald.conf(5)`, `journalctl(1)` y `systemctl(1)`

► Ejercicio Guiado

Restablecimiento de la contraseña de root

En este ejercicio, restablece la contraseña `root` en un sistema.

Resultados

- Restablecer una contraseña de usuario de `root` perdida.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando ejecuta un script de inicio que determina si la máquina `servera` es accesible en la red. También restablece la contraseña `root` a una cadena aleatoria y establece un tiempo de espera mayor para el menú de GRUB2.

```
[student@workstation ~]$ lab start boot-resetting
```

Instrucciones

- 1. Vuelva a arrancar `servera` e interrumpa la cuenta regresiva en el menú del cargador de arranque.
 - 1.1. Localice el icono de la consola de `servera`, según corresponda para el entorno del aula y, luego, abra la consola.
Envíe `Ctrl+Alt+Del` a su sistema usando la entrada del menú o el botón relevantes.
 - 1.2. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla (excepto `Enter`) para interrumpir la cuenta regresiva.
- 2. Edite la entrada del cargador de arranque del kernel de rescate (en la memoria) para anular el proceso de arranque después de que todos los sistemas de archivos hayan sido montados por el kernel, pero antes de que el control se entregue a `systemd`.
 - 2.1. Use las teclas del cursor para resaltar la entrada del kernel de rescate (la que tiene la palabra `rescue` en su nombre).
 - 2.2. Presione `e` para editar la entrada actual.
 - 2.3. Use las teclas de dirección para navegar hacia la línea que comienza con `linux`.
 - 2.4. Presione `End` (Fin) para mover el cursor hasta el final de la línea.
 - 2.5. Agregue `rd.break` en el final de la línea.
 - 2.6. Presione `Ctrl+x` para realizar el arranque con la configuración modificada.

- ▶ 3. Presione **Enter** para realizar el mantenimiento. En el prompt `sh-5.1#`, vuelva a montar el sistema de archivos `/sysroot` con acceso de lectura y escritura y, luego, use el comando `chroot` para ingresar a un jail `chroot` en `/sysroot`.

```
sh-5.1# mount -o remount,rw /sysroot
...output omitted...
sh-5.1# chroot /sysroot
```

- ▶ 4. Cambie la contraseña `root` a `redhat`.

```
sh-5.1# passwd root
Changing password for user root.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

- ▶ 5. Configure el sistema para que realice automáticamente un etiquetado nuevo de SELinux completo después del arranque. Esto es necesario porque el comando `passwd` recrea el archivo `/etc/shadow` sin un contexto SELinux.

```
sh-5.1# touch /.autorelabel
```

- ▶ 6. Escriba `exit` dos veces para continuar reiniciando su sistema de forma habitual. El sistema ejecuta una operación de reetiquetado de SELinux y, luego, se reinicia automáticamente. Cuando el sistema esté funcionando, verifique su trabajo al iniciar sesión como `root` en la consola.

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish boot-resetting
```

Esto concluye la sección.

Reparación de problemas del sistema de archivos en el arranque

Objetivos

Reparar manualmente la configuración del sistema de archivos o problemas de daños que detengan el proceso de arranque.

Problemas del sistema de archivos

Durante el proceso de arranque, el servicio `systemd` monta los sistemas de archivos persistentes definidos en el archivo `/etc/fstab`.

Los errores en el archivo `/etc/fstab` o los sistemas de archivos dañados pueden impedir que un sistema complete el proceso de arranque. En algunos escenarios de falla, el sistema sale del proceso de arranque y abre una shell de emergencia que requiere la contraseña de usuario `root`.

La siguiente lista describe algunos problemas comunes de montaje del sistema de archivos al analizar `/etc/fstab` durante el proceso de arranque:

Sistema de archivos dañado

El servicio `systemd` intenta reparar el sistema de archivos. Si el problema no se puede reparar automáticamente, el sistema abre una shell de emergencia.

Dispositivo o UUID no existente

El tiempo de espera del servicio `systemd` se agota esperando que el dispositivo esté disponible. Si el dispositivo no responde, el sistema abre una shell de emergencia.

Punto de montaje no existente o incorrecto

El sistema abre una shell de emergencia.

Reparación de problemas del sistema de archivos

Para obtener acceso a un sistema que no puede completar el arranque debido a problemas del sistema de archivos, la arquitectura `systemd` proporciona un objetivo de arranque `emergency`, que abre una shell de emergencia que requiere la contraseña `root` para acceder.

El siguiente ejemplo demuestra la salida del proceso de arranque cuando el sistema encuentra un problema del sistema de archivos y cambia al objetivo `emergency`:

```
...output omitted...
[*      ] A start job is running for /dev/vda2 (27s / 1min 30s)
[ TIME  ] Timed out waiting for device /dev/vda2.
[DEPEND] Dependency failed for /mnt/mountfolder
[DEPEND] Dependency failed for Local File Systems.
[DEPEND] Dependency failed for Mark need to relabel after reboot.
...output omitted...
[ OK   ] Started Emergency Shell.
[ OK   ] Reached target Emergency Mode.
...output omitted...
Give root password for maintenance
(or press Control-D to continue):
```

El daemon `systemd` no pudo montar el dispositivo `/dev/vda2` y se agotó el tiempo de espera. Debido a que el dispositivo no está disponible, el sistema abre una shell de emergencia para el acceso de mantenimiento.

Para reparar problemas del sistema de archivos cuando su sistema abre una shell de emergencia, primero localice el sistema de archivos errante, determine y repare la falla, luego vuelva a cargar la configuración `systemd` para volver a intentar el montaje automático.

Use el comando `mount` para determinar qué sistemas de archivos están montados actualmente por el daemon `systemd`.

```
[root@host ~]# mount
...output omitted...
/dev/vda1 on / type xfs (ro,relatime,seclabel,attr2,inode64,noquota)
...output omitted...
```

Si el sistema de archivos `root` se monta con la opción `ro` (solo lectura), usted no podrá editar el archivo `/etc/fstab`. Vuelva a montar temporalmente el sistema de archivos `root` con la opción `rw` (lectura y escritura), si es necesario, antes de abrir el archivo `/etc/fstab`. La opción de volver a montar permite que un sistema de archivos en uso modifique sus parámetros de montaje sin tener que desmontar el sistema de archivos.

```
[root@host ~]# mount -o remount,rw /
```

Intente montar todos los sistemas de archivos enumerados en el archivo `/etc/fstab` con la opción `mount --all`. Esta opción monta procesos en cada entrada del sistema de archivos, pero omite los que ya están montados. El comando muestra los errores que ocurren al montar un sistema de archivos.

```
[root@host ~]# mount --all
mount: /mnt/mountfolder: mount point does not exist.
```

En este escenario, el directorio de montaje `/mnt/mountfolder` no existe; cree el directorio `/mnt/mountfolder` antes de volver a intentar el montaje. Pueden aparecer otros mensajes de error, incluidos errores tipográficos en las entradas o nombres de dispositivos o UUID incorrectos.

Cuando haya corregido todos los problemas en el archivo `/etc/fstab`, informe al daemon `systemd` para que registre el nuevo archivo `/etc/fstab` con el comando `systemctl daemon-reload`, luego intente montar todas las entradas.

```
[root@host ~]# systemctl daemon-reload
[root@host ~]# mount --all
```



nota

El servicio `systemd` procesa el archivo `/etc/fstab` transformando cada entrada en una configuración de unidad `systemd` tipo `.mount` y, luego, inicie la unidad como un servicio. `daemon-reload` solicita a que `systemd` vuelva a compilar y a cargar todas las configuraciones de unidad.

Si el intento `mount --all` se ejecuta correctamente sin más errores, la prueba final es verificar que el montaje del sistema de archivos se realiza correctamente durante el arranque del sistema. Reinicie el sistema y espere a que el arranque finalice normalmente.

```
[root@host ~]# systemctl reboot
```

Para realizar pruebas rápidas en el archivo `/etc/fstab`, use la opción de entrada de montaje `nofail`. Usar la opción `nofail` en una entrada `/etc/fstab` permite que el sistema arranque, incluso si el montaje de ese sistema de archivos no se ejecuta de forma correcta. Esta opción no debe usarse con sistemas de archivos de producción que siempre deben montarse. Con la opción `nofail`, una aplicación se podría iniciar sin los datos del sistema de archivos y generar posibles consecuencias graves.



Referencias

Páginas del manual: `systemd-fsck(8)`, `systemd-fstab-generator(8)` y `systemd.mount(5)`

► Ejercicio Guiado

Reparación de problemas del sistema de archivos en el arranque

En este ejercicio, recuperará un sistema de una configuración incorrecta en el archivo `/etc/fstab` que hace que falle el proceso de arranque.

Resultados

- Diagnosticar problemas del archivo `/etc/fstab` y usar el modo de emergencia para recuperar el sistema.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start boot-repairing
```

Instrucciones

- 1. Acceda a la consola de la máquina `servera` y observe que el proceso de arranque no se ejecuta correctamente desde el principio.
 - 1.1. Localice el icono de la consola de `servera`, según corresponda para el entorno del aula. Abra la consola.
Advierta que, al parecer, un trabajo de inicio no se ejecuta por completo. Tómese un momento para especular sobre la posible causa de este comportamiento.
 - 1.2. Reinicie la máquina `servera` al enviar un `Ctrl+Alt+Del` a su sistema usando la entrada del menú o el botón relevantes. En el caso de este problema de arranque en particular, esta secuencia de teclas podría no detener inmediatamente el trabajo en ejecución y es posible que tenga que esperar a que finalice el tiempo de espera antes de que el sistema se reinicie.
Si espera a que la tarea finalice el tiempo de espera sin enviar `Ctrl+Alt+Del`, el sistema eventualmente genera por su cuenta una shell de emergencia.
 - 1.3. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla (excepto `Enter`) para interrumpir la cuenta regresiva.
- 2. Si observa el error que tuvo en el arranque anterior, parece que al menos ciertas partes del sistema aún están funcionando. Use `redhat` como contraseña de usuario `root` para intentar un arranque de emergencia.
 - 2.1. Use las teclas de dirección para destacar la entrada del cargador de arranque predeterminada.

- 2.2. Presione la tecla **e** para editar la entrada actual.
- 2.3. Use las teclas de dirección para navegar hacia la línea que comienza con la palabra **linux**.
- 2.4. Presione **End** (Fin) para mover el cursor hasta el final de la línea.
- 2.5. Agregue la cadena **systemd.unit=emergency.target** en el final de la línea.
- 2.6. Presione **Ctrl+x** para realizar el arranque con la configuración modificada.

► 3. Inicie sesión en el modo de emergencia.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@servera ~]#
```

► 4. Determine qué sistemas de archivos está montando el daemon **systemd** actualmente. Observe que el daemon **systemd** monta el sistema de archivos **root** en modo de solo lectura.

```
[root@servera ~]# mount
...output omitted...
/dev/vda4 on / type xfs
(ro,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
...output omitted...
```

► 5. Vuelva a montar el sistema de archivos **root** en modo de lectura y escritura.

```
[root@servera ~]# mount -o remount,rw /
```

► 6. Intente montar todos los demás sistemas de archivos. La opción **--all (-a)** monta todos los sistemas de archivos detallados en el archivo **/etc/fstab** que todavía no están montados.

```
[root@servera ~]# mount -a
mount: /RemoveMe: mount point does not exist.
```

► 7. Edite el archivo **/etc/fstab** para corregir el problema.

- 7.1. Elimine o comente la línea incorrecta con el comando **vim /etc/fstab**.

```
[root@servera ~]# cat /etc/fstab
...output omitted...
# /dev/sdz1 /RemoveMe xfs defaults 0 0
```

- 7.2. Vuelva a cargar el daemon **systemd** para que el sistema registre la nueva configuración del archivo **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

- ▶ 8. Intente montar todas las entradas para verificar que el archivo `/etc/fstab` ahora sea correcto.

```
[root@servera ~]# mount -a
```

- ▶ 9. Reinicie el sistema y espere a que el arranque finalice. Ahora el sistema debería arrancar normalmente.

```
[root@servera ~]# systemctl reboot
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish boot-repairing
```

Esto concluye la sección.

► Trabajo de laboratorio

Control del proceso de arranque

En este trabajo de laboratorio, restablece la contraseña de `root` en un sistema, corrige una configuración incorrecta y establece el objetivo de arranque predeterminado.

Resultados

- Restablecer una contraseña perdida para el usuario `root`.
- Diagnosticar y corregir problemas de arranque.
- Establecer el objetivo `systemd` predeterminado.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start boot-review
```

Instrucciones

1. En la máquina `serverb`, restablezca la contraseña a `redhat` para el usuario `root`.
Localice el icono de la consola de la máquina `serverb`, según corresponda para el entorno del aula y, luego, abra la consola.
2. En el menú del cargador de arranque, seleccione la entrada predeterminada del cargador de arranque del kernel. El sistema no puede arrancar porque un trabajo de inicio no se completa correctamente. Corrija el problema desde la consola de la máquina `serverb`.
3. Cambie el objetivo de `systemd` predeterminado en la máquina `serverb` para que el sistema inicie automáticamente una interfaz gráfica en el arranque.
No hay una interfaz gráfica instalada en la máquina `serverb`. Solo establezca el objetivo predeterminado para este ejercicio y no instale los paquetes.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade boot-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish boot-review
```

Esto concluye la sección.

► Solución

Control del proceso de arranque

En este trabajo de laboratorio, restablece la contraseña de `root` en un sistema, corrige una configuración incorrecta y establece el objetivo de arranque predeterminado.

Resultados

- Restablecer una contraseña perdida para el usuario `root`.
- Diagnosticar y corregir problemas de arranque.
- Establecer el objetivo `systemd` predeterminado.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start boot-review
```

Instrucciones

1. En la máquina `serverb`, restablezca la contraseña a `redhat` para el usuario `root`.
Localice el icono de la consola de la máquina `serverb`, según corresponda para el entorno del aula y, luego, abra la consola.
 - 1.1. Envíe `Ctrl+Alt+Del` a su sistema usando la entrada del menú o el botón relevantes.
 - 1.2. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla (excepto `Enter`) para interrumpir la cuenta regresiva.
 - 1.3. Use las teclas del cursor para resaltar la entrada del cargador de arranque del kernel de rescate (la que tiene la palabra `rescue` en su nombre).
 - 1.4. Presione `e` para editar la entrada actual.
 - 1.5. Use las teclas de dirección para navegar por la línea que comienza con el texto `linux`.
 - 1.6. Presione `Ctrl+e` (Fin) para mover el cursor hasta el final de la línea.
 - 1.7. Agregue el texto `rd.break` en el final de la línea.
 - 1.8. Presione `Ctrl+x` para realizar el arranque con la configuración modificada.
 - 1.9. Presione `Enter` para ingresar al modo de mantenimiento.
 - 1.10. En el prompt `sh-5.1`, vuelva a montar el sistema de archivos `/sysroot` en modo escritura, luego use el comando `chroot` para el directorio `/sysroot`.

```
sh-5.1# mount -o remount,rw /sysroot
...output omitted...
sh-5.1# chroot /sysroot
```

- 1.11. Establezca **redhat** como contraseña para el usuario **root**.

```
sh-5.1# passwd root
Changing password for user root.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

- 1.12. Configure el sistema para que realice automáticamente un etiquetado nuevo de SELinux completo después del arranque.

```
sh-5.1# touch /.autorelabel
```

- 1.13. Salga del entorno **chroot** y del prompt **sh-5.1**. Después de volver a etiquetar el sistema de archivos, el sistema solicita ingresar al modo de mantenimiento, pero si espera, completa el reinicio y muestra el menú del cargador de arranque.
2. En el menú del cargador de arranque, seleccione la entrada predeterminada del cargador de arranque del kernel. El sistema no puede arrancar porque un trabajo de inicio no se completa correctamente. Corrija el problema desde la consola de la máquina **serverb**.
 - 2.1. Arranque el sistema en modo de emergencia. Reinicie la máquina **serverb** al enviar un **Ctrl+Alt+Del** a su sistema usando la entrada del menú o el botón relevantes.
 - 2.2. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla (excepto **Enter**) para interrumpir la cuenta regresiva.
 - 2.3. Use las teclas de dirección para destacar la entrada predeterminada del cargador de arranque.
 - 2.4. Presione **e** para editar la entrada actual.
 - 2.5. Use las teclas de dirección para navegar por la línea que comienza con el texto **linux**.
 - 2.6. Presione **Ctrl+e** (Fin) para mover el cursor hasta el final de la línea.
 - 2.7. Agregue el texto **systemd.unit=emergency.target** en el final de la línea.
 - 2.8. Presione **Ctrl+x** para realizar el arranque con la configuración modificada.
 - 2.9. Inicie sesión en el modo de emergencia.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@serverb ~]#
```

- 2.10. Vuelva a montar el sistema de archivos / en modo escritura.

```
[root@serverb ~]# mount -o remount,rw /
...output omitted...
```

2.11. Monte todos los sistemas de archivos.

```
[root@serverb ~]# mount -a
mount: /olddata: can't find UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc.
```

2.12. Edite el archivo `/etc/fstab` para eliminar o comentar la línea incorrecta que monta el punto de montaje `/olddata`.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc /olddata xfs defaults 0 0
```

2.13. Actualice el daemon `systemd` para que el sistema registre los cambios en la configuración del archivo `/etc/fstab`.

```
[root@serverb ~]# systemctl daemon-reload
```

2.14. Intente montar todas las entradas para verificar que la configuración del archivo `/etc/fstab` sea correcto.

```
[root@serverb ~]# mount -a
```

2.15. Reinicie el sistema y espere a que el arranque finalice. Ahora el sistema debería arrancar normalmente.

```
[root@serverb ~]# systemctl reboot
```

3. Cambie el objetivo de `systemd` predeterminado en la máquina `serverb` para que el sistema inicie automáticamente una interfaz gráfica en el arranque.

No hay una interfaz gráfica instalada en la máquina `serverb`. Solo establezca el objetivo predeterminado para este ejercicio y no instale los paquetes.

3.1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3.2. Establezca `graphical.target` como el objetivo predeterminado.


```
[root@serverb ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target → /usr/lib/systemd/system/graphical.target.
```

3.3. Verifique que esté configurado el valor predeterminado correcto.

```
[root@serverb ~]# systemctl get-default
graphical.target
```

3.4. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade boot-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish boot-review
```

Esto concluye la sección.

Resumen

- Los comandos `systemctl reboot` y `systemctl poweroff` reinician y apagan el sistema, respectivamente.
- El comando `systemctl isolate target-name.target` cambia a un nuevo objetivo en tiempo de ejecución.
- Los comandos `systemctl get-default` y `systemctl set-default` se pueden usar para consultar y establecer el objetivo predeterminado.
- Puede usar la opción `rd.break` en la línea de comandos del kernel para interrumpir el proceso de arranque antes de que se entregue el control desde el `initramfs`. El sistema de archivos `root` se monta como solo lectura en `/sysroot`.
- El objetivo de emergencia se puede usar para diagnosticar y corregir problemas de sistemas de archivos.

capítulo 9

Administración de la seguridad de redes

Meta

Controlar las conexiones de red a los servicios con el firewall del sistema y las reglas de SELinux.

Objetivos

- Aceptar o rechazar las conexiones de red a los servicios del sistema con reglas de firewall.
- Verificar que los puertos de red tengan el tipo de SELinux correcto para que los servicios hagan referencia a ellos.

Secciones

- Administración de firewalls del servidor (y ejercicio guiado)
- Control del etiquetado de puertos de SELinux (y ejercicio guiado)

Trabajo de laboratorio

Administración de la seguridad de redes

Administración de firewalls del servidor

Objetivos

Aceptar o rechazar las conexiones de red a los servicios del sistema con reglas de `firewalld`.

Conceptos de arquitectura del firewall

El kernel de Linux proporciona el marco (framework) `netfilter` para las operaciones de tráfico de red, como el filtrado de paquetes, la traducción de direcciones de red y la traducción de puertos. El marco (framework) `netfilter` incluye *enlaces* para que los módulos del kernel interactúen con los paquetes de red a medida que atraviesan la pila (stack) de red de un sistema. Fundamentalmente, los enlaces `netfilter` son rutinas del kernel que interceptan eventos (por ejemplo, un paquete que ingresa a una interfaz) y ejecutan otras rutinas relacionadas (por ejemplo, reglas de firewall).

El marco (framework) de `nftables`

El marco (framework) de clasificación de paquetes `nftables` se basa en el marco `netfilter` para aplicar reglas de firewall al tráfico de red. En Red Hat Enterprise Linux 9, `nftables` es el núcleo del firewall del sistema y reemplaza al marco (framework) obsoleto `iptables`.

El marco (framework) `nftables` ofrece numerosas ventajas sobre `iptables`, incluida la facilidad de uso y conjuntos de reglas más eficientes. Por ejemplo, el marco (framework) `iptables` requería una regla para cada protocolo, pero las reglas `nftables` se pueden aplicar al tráfico IPv4 e IPv6 simultáneamente. El marco (framework) `iptables` requería el uso de diferentes herramientas, como `iptables`, `ip6tables`, `arptables` y `ebtables`, para cada protocolo, pero el marco `nftables` usa la utilidad de espacio de usuario `nft` para administrar todos los protocolos a través de una única interfaz.



nota

Convierta los archivos de configuración heredados `iptables` en sus equivalentes `nftables` mediante las utilidades `iptables-translate` y `ip6tables-translate`.

El servicio `firewalld`

El servicio `firewalld` es un administrador de firewall dinámico y es el frontend recomendado para el marco (framework) `nftables`. La distribución de Red Hat Enterprise Linux 9 incluye el paquete `firewalld`.

El servicio `firewalld` simplifica la administración de firewall al clasificar todo el tráfico de la red en *zonas*. La zona asignada de un paquete de red depende de criterios como la dirección IP de origen del paquete o la interfaz de red entrante. Cada zona tiene su propia lista de puertos y servicios que están abiertos o cerrados.

**nota**

En el caso de equipos portátiles u otras máquinas que cambian regularmente las redes, el servicio **NetworkManager** puede configurar automáticamente la zona de firewall para una conexión. Esto es útil en el cambio entre el hogar, el trabajo y las redes inalámbricas públicas. Un usuario podría desear llegar al servicio **sshd** de su sistema cuando se conecta a las redes de su hogar o corporativas, pero no cuando se conecta a una red inalámbrica pública en la tienda de café local.

El servicio **firewalld** verifica la dirección de origen para cada paquete que ingresa al sistema. Si esa dirección de origen está asignada a una zona específica, rigen las reglas de esa zona. Si la dirección de origen no está asignada a una zona, **firewalld** asocia el paquete con la zona para la interfaz de red entrante y rigen las reglas para esa zona. Si la interfaz de red no está asociada con una zona, **firewalld** envía el paquete con la zona predeterminada.

La zona predeterminada no es una zona separada, sino una designación asignada a una zona existente. Inicialmente, el servicio **firewalld** designa la zona **public** como la zona predeterminada y asigna la interfaz de bucle invertido **lo** a la zona **trusted**.

La mayoría de las zonas permiten el tráfico a través del firewall que relaciona una lista de puertos y protocolos particulares (como **631/udp**) o una configuración de servicios predefinidos (como **ssh**). Normalmente, si el tráfico no relaciona un puerto y protocolo o servicio permitidos, entonces se rechaza. La zona **trusted**, que permite todo el tráfico de forma predeterminada, es una excepción.

Zonas predefinidas

El servicio **firewalld** usa zonas predefinidas, que puede personalizar. De forma predeterminada, todas las zonas permiten todo el tráfico entrante que sea parte de una sesión iniciada existente por el sistema y también todo el tráfico saliente. En la siguiente tabla, se detalla la configuración de zona inicial.

Configuración predeterminada de zonas firewalld

Nombre de la zona	Configuración predeterminada
trusted	Permite todo el tráfico entrante.
home	Rechaza el tráfico entrante, a menos que esté relacionado con tráfico saliente o que relacione los servicios predefinidos ssh , mdns , ipp-client , samba-client o dhcpv6-client .
internal	Rechaza el tráfico entrante, a menos que esté relacionado con tráfico saliente o que relacione los servicios predefinidos ssh , mdns , ipp-client , samba-client o dhcpv6-client (lo mismo que la zona home para empezar).
work	Rechaza el tráfico entrante, a menos que esté relacionado con tráfico saliente o que relacione los servicios predefinidos ssh , ipp-client o dhcpv6-client .

Nombre de la zona	Configuración predeterminada
public	Rechaza el tráfico entrante, a menos que esté relacionado con tráfico saliente o que relacione los servicios predefinidos <code>ssh</code> o <code>dhcpv6-client</code> . La zona predeterminada para interfaces de red recientemente agregadas.
external	Rechaza el tráfico entrante a menos que esté relacionado con tráfico saliente o que relacione el servicio predefinido <code>ssh</code> . El tráfico IPv4 saliente reenviado a través de esta zona es <i>enmascarado</i> para que luzca como si se hubiera originado desde la dirección IPv4 de la interfaz de red saliente.
dmz	Rechaza el tráfico entrante a menos que esté relacionado con tráfico saliente o que relacione el servicio predefinido <code>ssh</code> .
block	Rechaza todo el tráfico entrante, a menos que esté relacionado con tráfico saliente.
drop	Deja caer todo el tráfico entrante, a menos que esté relacionado con tráfico saliente (ni siquiera responde con errores ICMP).

Para conocer una lista de las zonas predefinidas y sus usos previstos, consulte la página del manual `firewalld.zones(5)`.

Servicios predefinidos

El servicio `firewalld` incluye una serie de configuraciones predefinidas para servicios comunes, para simplificar la configuración de reglas de firewall. Por ejemplo, en lugar de investigar los puertos relevantes para un servidor NFS, use la configuración predefinida `nfs` para crear reglas para los puertos y protocolos correctos. En la siguiente tabla, se enumeran las configuraciones de servicio predefinidas que el servicio `firewalld` usa en su configuración predeterminada.

Servicios firewalld predefinidos seleccionados

Nombre del servicio	Configuración
<code>ssh</code>	Servidor SSH local. Tráfico a 22/tcp.
<code>dhcpv6-client</code>	Cliente DHCPv6 local. Tráfico a 546/udp en la red <code>fe80::/64</code> IPv6.
<code>ipp-client</code>	Impresión IPP local. Tráfico a 631/udp.
<code>samba-client</code>	Archivo Windows local y cliente de intercambio de impresión. Tráfico a 137/udp y 138/udp.
<code>mdns</code>	Resolución del nombre del enlace local DNS (mDNS) multidifusión (multicast). Tráfico a 5353/udp a las direcciones de multidifusión (multicast) 224.0.0.251 (IPv4) o <code>ff02::fb</code> (IPv6).

El paquete `firewalld` incluye muchas configuraciones de servicio predefinidas. Puede enumerar los servicios con el comando `firewall-cmd --get-services`.

```
[root@host ~]# firewall-cmd --get-services
RH-Satellite-6 RH-Satellite-6-capsule amanda-client amanda-k5-client amqp amqps
apcupsd audit bacula bacula-client bb bgp bitcoin bitcoin-rpc bitcoin-testnet
bitcoin-testnet-rpc bittorrent-lsd ceph ceph-mon cfengine cockpit collectd
...output omitted...
```

Si las configuraciones de servicio predefinidas no son adecuadas para su escenario, puede especificar manualmente los puertos y protocolos requeridos. Puede usar la interfaz gráfica de la consola web para revisar los servicios predefinidos y para definir manualmente puertos y protocolos adicionales.

Configuración del daemon firewalld

Entre otras, estas son dos formas comunes que usan los administradores de sistemas para interactuar con el servicio `firewalld`:

- La interfaz gráfica de la consola web
- La herramienta de línea de comandos `firewall-cmd`

Configuración de los servicios de firewall con la consola web

Para administrar los servicios de firewall con la consola web, debe iniciar sesión y aumentar los privilegios. Puede aumentar los privilegios haciendo clic en los botones **Limited access** (Acceso limitado) o **Turn on administrative access** (Activar acceso administrativo). A continuación, ingrese su contraseña cuando se le solicite. El modo administrativo eleva los privilegios en función de la configuración de sudo del usuario. Como recordatorio de seguridad, recuerde que debe volver al modo de acceso limitado una vez que realice en su sistema la tarea que requiere privilegios administrativos.

Haga clic en la opción **Networking (Redes)** en el menú de navegación izquierdo para mostrar la sección **Firewall (Firewall)** en la página principal de redes. Haga clic en las zonas del botón **Edit rules and zones** (Editar reglas y zonas) para navegar a la página **Firewall (Firewall)**.

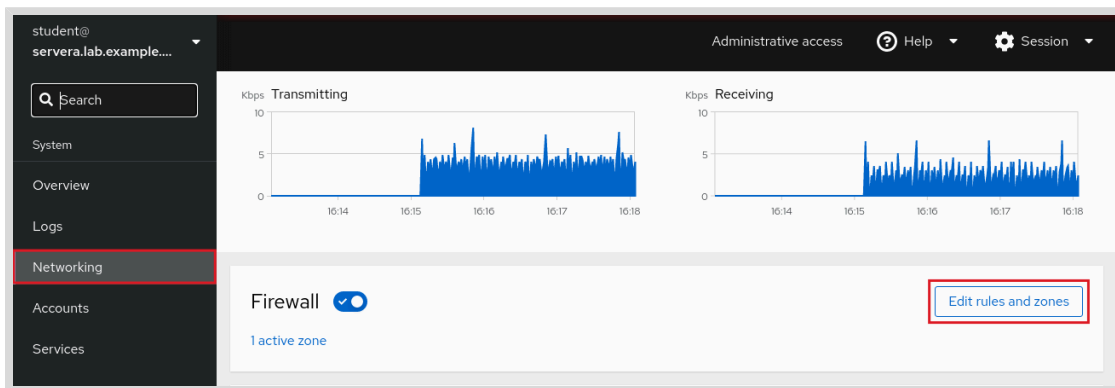


Figura 9.1: Páginas de redes de la consola web

La página **Firewall (Firewall)** muestra las zonas activas y sus servicios permitidos. Haga clic en el botón de la flecha (>) a la izquierda del nombre del servicio para ver los detalles del servicio. Para agregar un servicio a una zona, haga clic en el botón **Add services** (Agregar servicios) en la esquina superior derecha de la zona correspondiente.

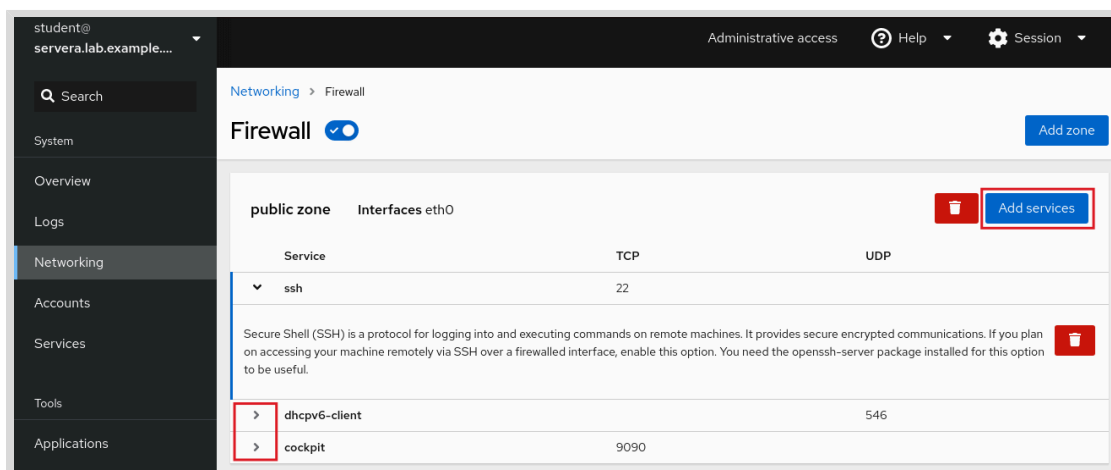


Figura 9.2: Página de firewall de la consola web

La página **Add Services** (Agregar servicios) muestra los servicios predefinidos disponibles.

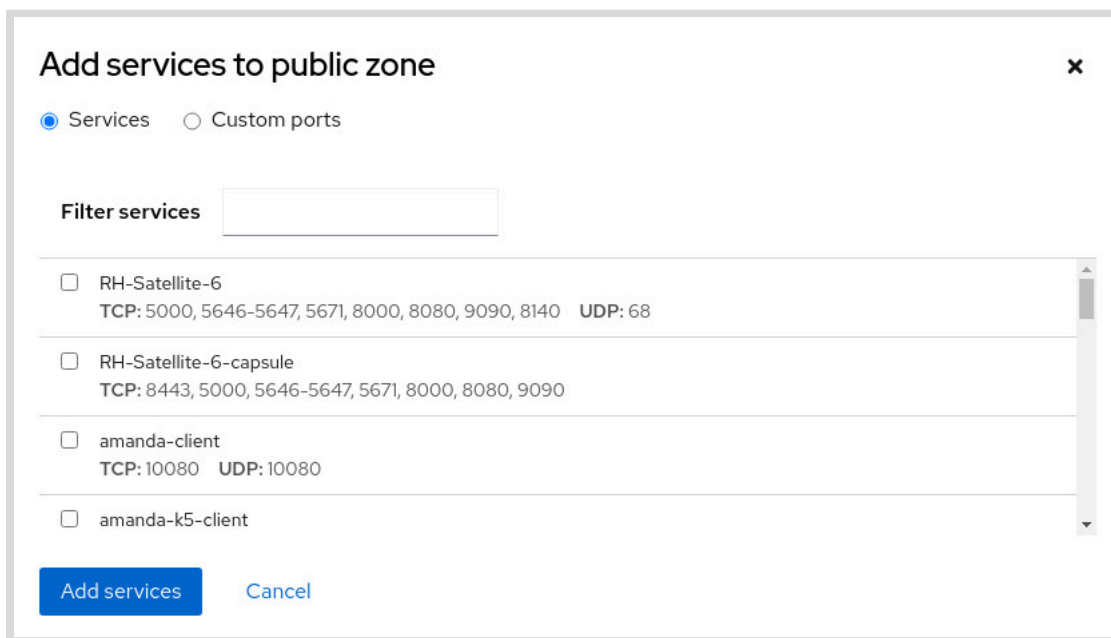


Figura 9.3: Menú para agregar servicios en la consola web

Para seleccionar un servicio, desplácese por la lista o seleccione una opción en el cuadro de texto **Filter services** (Filtrar servicios). En el siguiente ejemplo, la cadena `http` filtra las opciones a servicios relacionados con la web. Seleccione la casilla de verificación a la izquierda del servicio para que el firewall los permita. Haga clic en el botón **Add services** (Agregar servicios) para completar el proceso.

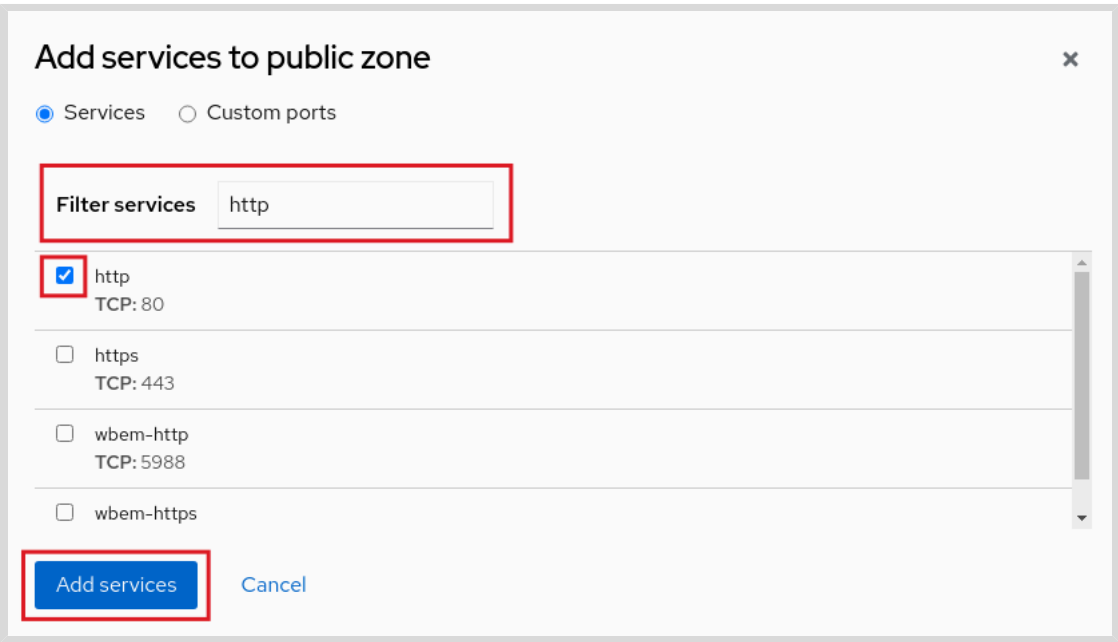


Figura 9.4: Opciones de menú para agregar servicios en la consola web

La interfaz vuelve a la página de Firewall (Firewall), donde puede revisar la lista actualizada de servicios permitidos.

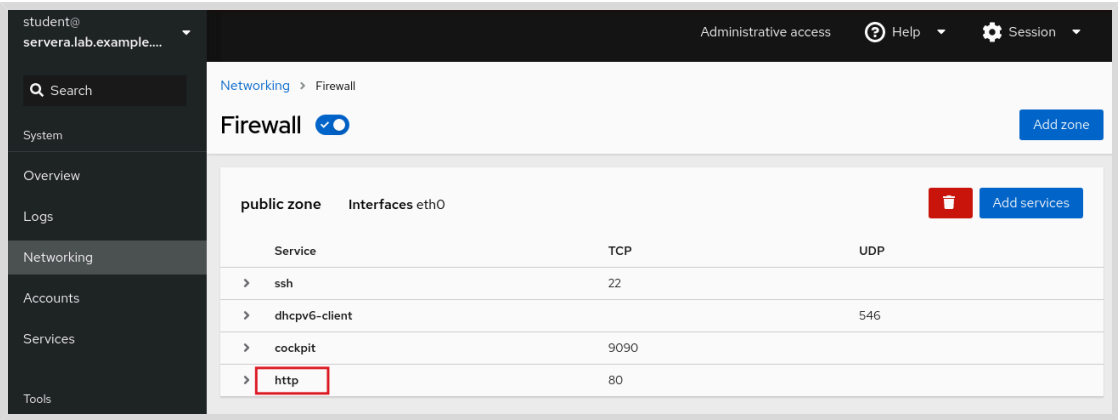


Figura 9.5: Descripción general del firewall de la consola web

Configuración de firewall desde la línea de comandos

El comando `firewall-cmd` interactúa con el daemon `firewalld`. Se instala como parte del paquete `firewalld` y está disponible para los administradores que prefieren trabajar en la línea de comandos a los fines de trabajar en sistemas sin un entorno gráfico o ejecutar el scripting de la configuración de firewall.

En la siguiente tabla, se detallan comandos de `firewall-cmd` usados frecuentemente, junto con una explicación. Observe que, a menos que se especifique lo contrario, casi todos los comandos funcionan en la configuración de *tiempo de ejecución*, a menos que se especifique la opción `--permanent`. Si se especifica la opción `--permanent`, debe activar la configuración ejecutando también el comando `firewall-cmd --reload`, que lee la configuración permanente actual y la aplica como la nueva configuración de tiempo de ejecución. Muchos de los comandos detallados toman la opción `--zone=ZONE` para determinar qué zona afectan. Cuando se requiera una máscara de red, use la notación CIDR (como 192.168.1/24).

Comandos firewall-cmd	Explicación
--get-default-zone	Consultar la zona predeterminada actual.
--set-default-zone=ZONE	Configurar la zona predeterminada. Esto cambia tanto la configuración del tiempo de ejecución como la permanente.
--get-zones	Mostrar todas las zonas disponibles.
--get-active-zones	Mostrar todas las zonas que están actualmente en uso (tienen una interfaz u origen conectados a esta), junto con la información de su interfaz y origen.
--add-source=CIDR [--zone=ZONE]	Enrutar todo el tráfico que proviene de la dirección IP o red/máscara de red a la zona especificada. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--remove-source=CIDR [--zone=ZONE]	Eliminar la regla que enruta todo el tráfico que proviene de la red de la dirección IP o red. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--add-interface=INTERFACE [--zone=ZONE]	Enrutar todo el tráfico que proviene de INTERFACE a la zona especificada. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--change-interface=INTERFACE [--zone=ZONE]	Asociar la interfaz con ZONE en lugar de su zona actual. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--list-all [--zone=ZONE]	Detallar todas las interfaces, fuentes, servicios y puertos configurados para ZONE. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--list-all-zones	Recuperar toda la información para todas las zonas (interfaces, fuentes, puertos, servicios).
--add-service=SERVICE [--zone=ZONE]	Permitir el tráfico a SERVICE. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.
--add-port=PORT/PROTOCOL [--zone=ZONE]	Permitir el tráfico a los puertos PORT/PROTOCOL. Si no se proporciona ninguna opción --zone=, se usa la zona predeterminada.

Comandos firewall-cmd	Explicación
<code>--remove-service=SERVICE [--zone=ZONE]</code>	Elimina <i>SERVICE</i> de la lista permitida para la zona. Si no se proporciona ninguna opción <code>--zone=</code> , se usa la zona predeterminada.
<code>--remove-port=PORT/PROTOCOL [--zone=ZONE]</code>	Eliminar los puertos <i>PORT/PROTOCOL</i> de la lista permitida para la zona. Si no se proporciona ninguna opción <code>--zone=</code> , se usa la zona predeterminada.
<code>--reload</code>	Dejar caer la configuración del tiempo de ejecución y aplicar la configuración persistente.

El siguiente ejemplo configura la zona predeterminada para *dmz*, asigna todo el tráfico proveniente de la red *192.168.0.0/24* a la zona *internal* y abren los puertos de red para el servicio *mysql* en la zona *internal*.

```
[root@host ~]# firewall-cmd --set-default-zone=dmz
[root@host ~]# firewall-cmd --permanent --zone=internal \
--add-source=192.168.0.0/24
[root@host ~]# firewall-cmd --permanent --zone=internal --add-service=mysql
[root@host ~]# firewall-cmd --reload
```



nota

En situaciones donde la sintaxis básica no es suficiente, puede agregar *reglas enriquecidas* para escribir reglas complejas. Si aun así la sintaxis de las reglas enriquecidas no es suficiente, también puede usar reglas de Configuración directa (sintaxis de *nft* sin formato mezclada con las reglas de *firewalld*). Esta configuración avanzada no está incluida en el alcance de este capítulo.



Referencias

Páginas del manual: *firewall-cmd*(1), *firewalld*(1), *firewalld.zone*(5), *firewalld.zones*(5) y *nft*(8)

► Ejercicio Guiado

Administración de firewalls del servidor

En este ejercicio, controla el acceso a los servicios del sistema mediante el ajuste de las reglas del firewall del sistema con `firewalld`.

Resultados

- Configurar reglas de firewall para controlar el acceso a los servicios.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start netsecurity-firewalls
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Instale los paquetes `httpd` y `mod_ssl`. Estos paquetes proporcionan el servidor web Apache y las extensiones necesarias para que el servidor web sirva contenido mediante SSL.

```
[root@servera ~]# dnf install httpd mod_ssl
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 3. Cree el archivo `/var/www/html/index.html`. Agregue la siguiente línea de texto: `I am servera..`

```
[root@servera ~]# echo 'I am servera.' > /var/www/html/index.html
```

- 4. Inicie y habilite el servicio `httpd`.

```
[root@servera ~]# systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
```

- 5. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

- 6. Desde `workstation`, intente acceder al servidor web en `servera` por medio del puerto no cifrado 80/TCP y del puerto encapsulado SSL 443/TCP. Ambos intentos deberían fallar.

6.1. El comando `curl` fallará.

```
[student@workstation ~]$ curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```

6.2. El comando `curl` con la opción `-k` para conexiones inseguras también fallará.

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 443: No route to host
```

- 7. Verifique que el servicio `firewalld` en `servera` esté habilitado y en ejecución.

```
[student@workstation ~]$ ssh student@servera 'systemctl status firewalld'
• firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
  Active: active (running) since Wed 2022-04-13 11:22:50 EDT; 7min ago
    Docs: man:firewalld(1)
  Main PID: 768 (firewalld)
    Tasks: 2 (limit: 10798)
  Memory: 39.9M
    CPU: 584ms
  CGroup: /system.slice/firewalld.service
          └─768 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid

Apr 13 11:22:49 servera.lab.example.com systemd[1]: Starting firewalld - dynamic firewall daemon...
Apr 13 11:22:50 servera.lab.example.com systemd[1]: Started firewalld - dynamic firewall daemon.
```

- 8. Desde `workstation`, abra Firefox e inicie sesión en la consola web que se ejecuta en `servera` para agregar el servicio `https` a la zona de firewall `public`.

- 8.1. Abra Firefox y navegue hasta `https://servera.lab.example.com:9090` para acceder a la consola web. Haga clic en **Advanced** (Avanzado) y **Accept the Risk and Continue** (Aceptar el riesgo y continuar) para aceptar el certificado autofirmado.
- 8.2. Inicie sesión con el usuario `student` y proporcione `student` como contraseña.
- 8.3. Haga clic en **Turn on administrative access** (Aceptar el riesgo y continuar) e ingrese la contraseña `student` nuevamente.
- 8.4. Haga clic en **Networking** (Redes) en la barra de navegación izquierda.
- 8.5. Haga clic en **Edit rules and zones** (Editar reglas y zonas) en la sección **Firewall** (Firewall) de la página **Networking** (Redes).
- 8.6. Haga clic en **Add services** (Agregar servicios) ubicado en la esquina superior derecha de la sección **public zone** (zona pública).
- 8.7. En la interfaz **Add services** (Agregar servicios), desplácese hacia abajo o use **Filter services** (Filtrar servicios) para ubicar y seleccionar la casilla de verificación junto al servicio `https`.
- 8.8. Haga clic en **Add services** (Agregar servicios) para aplicar el cambio.
- ▶ 9. Regrese a un terminal en `workstation` y verifique su trabajo intentando acceder al contenido del servidor web de `servera`.
 - 9.1. El comando `curl` para el puerto estándar 80 debe fallar.

```
[student@workstation ~]$ curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```

- 9.2. El comando `curl` con la opción `-k` en el puerto 443 debería funcionar correctamente.

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
I am servera.
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netsecurity-firewalls
```

Esto concluye la sección.

Control del etiquetado de puertos de SELinux

Objetivos

Verificar que los puertos de red tengan el tipo de SELinux correcto para que los servicios hagan referencia a ellos.

Etiquetado de puertos de SELinux

Además del contexto de archivo y el etiquetado de tipo de proceso, SELinux etiqueta los puertos de red con un contexto de SELinux. SELinux controla el acceso a la red al etiquetar los puertos de red e incluir reglas en la política de destino de un servicio. Por ejemplo, la política de destino SSH incluye el puerto 22/TCP con una etiqueta de contexto de puerto `ssh_port_t`. En la política HTTP, los puertos predeterminados 80/TCP y 443/TCP usan una etiqueta de contexto de puerto `http_port_t`.

Cuando un proceso objetivo intenta abrir un puerto para escuchar, SELinux verifica que la política objetivo incluya entradas, que permiten que ese tipo de proceso se vincule con ese tipo de contexto de puerto. SELinux puede bloquear un servicio no autorizado para que no use puertos de otra manera usados por otros servicios de red legítimos.

Administración del etiquetado de puertos de SELinux

Si un servicio intenta escuchar en un puerto no estándar, SELinux bloquea el intento, a menos que opte por incluir el puerto en la política de destino del servicio al etiquetar el puerto con el contexto de puerto correcto.

Por lo general, la política `targeted` ya ha etiquetado todos los puertos esperados con el tipo correcto. Por ejemplo, debido a que el puerto 8008/TCP se usa a menudo para aplicaciones web, ese puerto ya está etiquetado con `http_port_t`, el tipo de puerto predeterminado para usar por un servidor web. Los puertos individuales se pueden etiquetar con un solo contexto de puerto.

Listado de etiquetas de puertos

Use el comando `grep` para filtrar el número de puerto.

```
[root@host ~]# grep gopher /etc/services
gopher          70/tcp          # Internet Gopher
gopher          70/udp
```

Use el comando `semanage` para enumerar las asignaciones de etiquetas de puerto actuales.

```
[root@host ~]# semanage port -l
...output omitted...
http_cache_port_t      tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t      udp      3130
http_port_t            tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
...output omitted...
```

Use el comando `grep` para filtrar la etiqueta del puerto de SELinux con el nombre del servicio.

```
[root@host ~]# semanage port -l | grep ftp
ftp_data_port_t      tcp      20
ftp_port_t           tcp      21, 989, 990
ftp_port_t           udp      989, 990
tftp_port_t          udp      69
```

Una etiqueta de puerto puede aparecer en la lista varias veces para cada protocolo de red soportado.

Use el comando `grep` para filtrar la etiqueta del puerto de SELinux con el número de puerto.

```
[root@host ~]# semanage port -l | grep -w 70
gopher_port_t        tcp      70
gopher_port_t        udp      70
```

Administración de enlaces de puertos

Use el comando `semanage` para asignar etiquetas de puertos nuevas, quitar etiquetas de puertos y modificar las existentes.



Importante

Prácticamente todos los servicios incluidos en la distribución de RHEL proporcionan un módulo de políticas de SELinux, que incluye los contextos de puerto predeterminados de ese servicio. No puede cambiar las etiquetas de puerto predeterminadas con el comando `semanage`. En su lugar, debe modificar y volver a cargar el módulo de políticas del servicio de destino. En este curso no se analiza la escritura ni la generación de módulos de políticas.

Puede etiquetar un nuevo puerto con una etiqueta de contexto de puerto existente (tipo). La opción `-a` del comando `semanage port` agrega una nueva etiqueta de puerto, la opción `-t` denota el tipo y la opción `-p` denota el protocolo.

```
[root@host ~]# semanage port -a -t port_label -p tcp|udp PORTNUMBER
```

Por ejemplo, para permitir que el servicio `gopher` escuche en el puerto 71/TCP:

```
[root@host~]# semanage port -a -t gopher_port_t -p tcp 71
```

Para ver los cambios locales realizados en la política predeterminada, use la opción `-C` del comando `semanage port`.

```
[root@host~]# semanage port -l -C
SELinux Port Type      Proto  Port Number
gopher_port_t          tcp    71
```

Las políticas objetivo incluyen muchos tipos de puertos.

Las páginas del manual de SELinux específicas del servicio se nombran con el nombre del servicio más `_selinux`. Estas páginas de manual incluyen información específica del servicio sobre tipos de SELinux, booleanos y tipos de puerto, pero no se instalan de manera predeterminada. Para ver una lista de todas las páginas del manual de SELinux disponibles, instale el paquete y, luego, ejecute una búsqueda de palabras clave `man -k para _selinux`

```
[root@host ~]# dnf -y install selinux-policy-doc
[root@host ~]# man -k _selinux
```

Use el comando `semanage` para eliminar una etiqueta de puerto, con la opción `-d`. Por ejemplo, para quitar el enlace del puerto 71/TCP al tipo `gopher_port_t`:

```
[root@host ~]# semanage port -d -t gopher_port_t -p tcp 71
```

Para cambiar un enlace de puerto, cuando cambien los requisitos, use la opción `-m`. Esto es más eficiente que quitar un enlace antiguo y agregar uno nuevo.

Por ejemplo, para modificar el puerto 71/TCP desde `gopher_port_t` hasta `http_port_t`, use el siguiente comando:

```
[root@server ~]# semanage port -m -t http_port_t -p tcp 71
```

Visualice la modificación con el comando `semanage`.

```
[root@server ~]# semanage port -l -C
SELinux Port Type          Proto  Port Number

http_port_t                tcp    71
[root@server ~]# semanage port -l | grep http
http_cache_port_t          tcp    8080, 8118, 8123, 10001-10010
http_cache_port_t          udp    3130
http_port_t                tcp    71, 80, 81, 443, 488, 8008, 8009, 8443,
9000
pegasus_http_port_t        tcp    5988
pegasus_https_port_t       tcp    5989
```



Referencias

Páginas del manual: `semanage(8)`, `semanage-port(8)` y `*_selinux(8)`

► Ejercicio Guiado

Control del etiquetado de puertos de SELinux

En este trabajo de laboratorio, configurará su sistema para permitir el acceso HTTP en un puerto no estándar.

Resultados

- Configurar un servidor web que se ejecuta en `servera` de manera correcta y ofrece contenido por medio de un puerto no estándar.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando determina si se puede acceder a la máquina `servera` en la red, instala el servicio `httpd` y configura el firewall en `servera` para permitir conexiones `http`.

```
[student@workstation ~]$ lab start netsecurity-ports
```

Instrucciones

Su organización está implementando una nueva aplicación web personalizada. La aplicación web se ejecuta en un puerto no estándar; en este caso `82/TCP`.

Uno de sus administradores junior ya ha configurado la aplicación en su host `servera`. Sin embargo, no se puede acceder al contenido del servidor web.

- 1. Inicie sesión en `servera` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 2. Intente corregir el problema del contenido web al reiniciar el servicio `httpd`.

- 2.1. Reinicie el `httpd.service`. Se espera que este comando falle.

```
[root@servera ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 2.2. Visualice el estado del servicio `httpd`. Advierta el error `permission denied`.

```
[root@servera ~]# systemctl status -l httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Mon 2019-04-08 14:23:29 CEST; 3min 33s ago
     Docs: man:httpd.service(8)
   Process: 28078 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
  Main PID: 28078 (code=exited, status=1/FAILURE)
    Status: "Reading configuration..."

Apr 08 14:23:29 servera.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission denied: AH00072: make_sock: could not bind to address [::]:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: no listening sockets available, shutting down
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: AH00015: Unable to open logs
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Apr 08 14:23:29 servera.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

2.3. Compruebe si SELinux está impidiendo que httpd se enlace con el puerto 82/TCP.

```
[root@servera ~]# sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----

SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 82.

***** Plugin bind_ports (99.5 confidence) suggests *****

If you want to allow /usr/sbin/httpd to bind to network port 82
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 82 where PORT_TYPE is one of the following:
http_cache_port_t, http_port_t, jboss_management_port_t, jboss_messaging_port_t,
ntop_port_t, puppet_port_t.
...output omitted...
Raw Audit Messages
type=AVC msg=audit(1554726569.188:852): avc: denied { name_bind } for
pid=28393 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
...output omitted...
```

- ▶ 3. Configure SELinux para que permita que `httpd` se enlace con el puerto 82/TCP y luego reinicie el servicio `httpd.service`.

3.1. Encuentre un tipo de puerto apropiado para el puerto 82/TCP.

El tipo `http_port_t` incluye los puertos HTTP predeterminados, 80/TCP y 443/TCP. Este es el tipo de puerto correcto para el servidor web.

```
[root@servera ~]# semanage port -l | grep http
http_cache_port_t      tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t      udp      3130
http_port_t            tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t    tcp      5988
pegasus_https_port_t   tcp      5989
```

3.2. Asigne el puerto 82/TCP al tipo `http_port_t`.

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
```

3.3. Reinicie el servicio `httpd.service`. Este comando debe arrojar resultados satisfactorios.

```
[root@servera ~]# systemctl restart httpd.service
```

- ▶ 4. Verifique si ahora puede acceder al servidor web que se ejecuta en el puerto 82/TCP.

```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
```

- ▶ 5. En una ventana de terminal diferente, verifique si puede acceder al nuevo servicio web desde `workstation`.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
curl: (7) Failed to connect to servera.example.com:82; No route to host
```

Ese error significa que todavía no puede conectarse al servicio web desde `workstation`.

- ▶ 6. En `servera`, abra un puerto 82/TCP en el firewall.

6.1. Abra el puerto 82/TCP en la configuración permanente para la zona predeterminada en el firewall, en `servera`.

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
```

6.2. Active sus cambios del firewall en `servera`.

```
[root@servera ~]# firewall-cmd --reload
success
```

- ▶ 7. Acceda al servicio web desde `workstation`.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
Hello
```

- 8. Regrese al sistema `workstation` como el usuario `student`.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netsecurity-ports
```

Esto concluye la sección.

► Trabajo de laboratorio

Administración de la seguridad de redes

En este trabajo de laboratorio, configurará un firewall y los ajustes de SELinux para permitir el acceso a múltiples servidores web que se ejecutan en el mismo host.

Resultados

- Configurar el firewall y los ajustes de SELinux en un host del servidor web.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start netsecurity-review
```

Instrucciones

Su empresa ha decidido ejecutar una nueva aplicación web. Esta aplicación escucha en puertos 80/TCP y 1001/TCP. También debe poner a disposición el puerto 22/TCP para el acceso ssh. Todos los cambios que hace deben persistir en un reinicio.



Importante

El entorno de aprendizaje en línea de Red Hat necesita el puerto 5900/TCP para permanecer disponible para usar la interfaz gráfica. Este puerto también es conocido en el servicio `vnc-server`. Si accidentalmente se bloquea a usted mismo fuera de la máquina `serverb`, puede intentar recuperar el acceso al usar el comando `ssh` para su máquina `serverb` desde su máquina `workstation` o restablecer su máquina `serverb`. Si elige restablecer su máquina `serverb`, debe ejecutar los scripts de configuración para este trabajo de laboratorio nuevamente. La configuración de sus máquinas ya incluye una zona personalizada denominada `R0L` que abre estos puertos.

1. Desde la máquina `workstation`, pruebe acceder al servidor web predeterminado en `http://serverb.lab.example.com` y al host virtual en `http://serverb.lab.example.com:1001`.
2. Inicie sesión en la máquina `serverb` para determinar qué está impidiendo el acceso a los servidores web.
3. Configure SELinux para que permita que el servicio `httpd` escuche en el puerto 1001/TCP.
4. Desde `workstation`, pruebe acceder nuevamente al servidor web predeterminado en `http://serverb.lab.example.com` y al host virtual en `http://serverb.lab.example.com:1001`.

5. Inicie sesión en la máquina `serverb` para determinar si los puertos correctos están asignados al firewall.
6. Agregue el puerto `1001/TCP` a la configuración permanente para la zona de red `public` (pública). Confirme su configuración.
7. Desde `workstation`, confirme que el servidor web predeterminado en `serverb.lab.example.com` vuelva a `SERVER B` y que el host virtual en `serverb.lab.example.com:1001` vuelva a `VHOST 1`.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade netsecurity-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netsecurity-review
```

Esto concluye la sección.

► Solución

Administración de la seguridad de redes

En este trabajo de laboratorio, configurará un firewall y los ajustes de SELinux para permitir el acceso a múltiples servidores web que se ejecutan en el mismo host.

Resultados

- Configurar el firewall y los ajustes de SELinux en un host del servidor web.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start netsecurity-review
```

Instrucciones

Su empresa ha decidido ejecutar una nueva aplicación web. Esta aplicación escucha en puertos 80/TCP y 1001/TCP. También debe poner a disposición el puerto 22/TCP para el acceso ssh. Todos los cambios que hace deben persistir en un reinicio.



Importante

El entorno de aprendizaje en línea de Red Hat necesita el puerto 5900/TCP para permanecer disponible para usar la interfaz gráfica. Este puerto también es conocido en el servicio `vnc-server`. Si accidentalmente se bloquea a usted mismo fuera de la máquina `serverb`, puede intentar recuperar el acceso al usar el comando `ssh` para su máquina `serverb` desde su máquina `workstation` o restablecer su máquina `serverb`. Si elige restablecer su máquina `serverb`, debe ejecutar los scripts de configuración para este trabajo de laboratorio nuevamente. La configuración de sus máquinas ya incluye una zona personalizada denominada `ROL` que abre estos puertos.

1. Desde la máquina `workstation`, pruebe acceder al servidor web predeterminado en `http://serverb.lab.example.com` y al host virtual en `http://serverb.lab.example.com:1001`.
 - 1.1. Pruebe acceder al servidor web en `http://serverb.lab.example.com`. La prueba falla actualmente. El servidor web debería volver a `SERVER B`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
curl: (7) Failed to connect to serverb.lab.example.com port 80: Connection refused
```


- 1.2. Pruebe acceder al host virtual en `http://serverb.lab.example.com:1001`. La prueba falla actualmente. Por último, el host virtual debería volver a VHOST 1.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No route to host
```

2. Inicie sesión en la máquina `serverb` para determinar qué está impidiendo el acceso a los servidores web.

- 2.1. Inicie sesión en la máquina `serverb` con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 2.2. Determine si el servicio `httpd` está activo.

```
[student@serverb ~]$ systemctl is-active httpd
inactive
```

- 2.3. Habilite e inicie el servicio `httpd`. El servicio `httpd` no puede iniciarse.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
[sudo] password for student: student
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service" for details.
```

- 2.4. Investigue los motivos que el servicio `httpd` no se pudo iniciar.

```
[student@serverb ~]$ systemctl status httpd.service
x httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: failed (Result: exit-code) since Wed 2022-04-13 06:55:01 EDT; 2min 52s ago
     Docs: man:httpd.service(8)
   Process: 1640 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
    Main PID: 1640 (code=exited, status=1/FAILURE)
      Status: "Reading configuration..."
        CPU: 31ms

Apr 13 06:55:01 serverb.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 13 06:55:01 serverb.lab.example.com httpd[1640]: (13)Permission denied: AH00072: make_sock: could not bind to address [::]:1001
Apr 13 06:55:01 serverb.lab.example.com httpd[1640]: (13)Permission denied: AH00072: make_sock: could not bind to address 0.0.0.0:1001
```

```
Apr 13 06:55:01 serverb.lab.example.com httpd[1640]: no listening sockets
available, shutting down
Apr 13 06:55:01 serverb.lab.example.com httpd[1640]: AH00015: Unable to open logs
Apr 13 06:55:01 serverb.lab.example.com systemd[1]: httpd.service: Main process
exited, code=exited, status=1/FAILURE
Apr 13 06:55:01 serverb.lab.example.com systemd[1]: httpd.service: Failed with
result 'exit-code'.
Apr 13 06:55:01 serverb.lab.example.com systemd[1]: Failed to start The Apache
HTTP Server.
```

- 2.5. Compruebe si SELinux está impidiendo que el servicio `httpd` haga referencia al puerto 1001/TCP.

```
[student@serverb ~]$ sudo sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----

SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port
1001.

**** Plugin bind_ports (99.5 confidence) suggests ****

If you want to allow /usr/sbin/httpd to bind to network port 1001
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 1001
   where PORT_TYPE is one of the following: http_cache_port_t, http_port_t,
   jboss_management_port_t, jboss_messaging_port_t, ntop_port_t, puppet_port_t.

**** Plugin catchall (1.49 confidence) suggests ****
...output omitted...
```

3. Configure SELinux para que permita que el servicio `httpd` escuche en el puerto 1001/TCP.

- 3.1. Use el comando `semanage` para encontrar el tipo de puerto correcto.

```
[student@serverb ~]$ sudo semanage port -l | grep 'http'
http_cache_port_t      tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t      udp      3130
http_port_t            tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t    tcp      5988
pegasus_https_port_t   tcp      5989
```

- 3.2. Vincule el puerto 1001/TCP al tipo `http_port_t`.

```
[student@serverb ~]$ sudo semanage port -a -t http_port_t -p tcp 1001
```

- 3.3. Confirme si ese puerto 1001/TCP está enlazado con el tipo de puerto `http_port_t`.

```
[student@serverb ~]$ sudo semanage port -l | grep '^http_port_t'
http_port_t            tcp      1001, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

- 3.4. Habilite e inicie el servicio `httpd`.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
```

- 3.5. Verifique si el servicio `httpd` está en ejecución.

```
[student@serverb ~]$ systemctl is-active httpd
active
[student@serverb ~]$ systemctl is-enabled httpd
enabled
```

- 3.6. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

4. Desde `workstation`, pruebe acceder nuevamente al servidor web predeterminado en `http://serverb.lab.example.com` y al host virtual en `http://serverb.lab.example.com:1001`.

- 4.1. Pruebe acceder al servidor web en `http://serverb.lab.example.com`. El servidor web debería volver a `SERVER B`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
SERVER B
```

- 4.2. Pruebe acceder al host virtual en `http://serverb.lab.example.com:1001`. La prueba sigue fallando.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No route to host
```

5. Inicie sesión en la máquina `serverb` para determinar si los puertos correctos están asignados al firewall.

- 5.1. Inicie sesión en la máquina `serverb` como el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 5.2. Verifique que la zona de firewall predeterminada esté configurada en la zona `public`.

```
[student@serverb ~]$ firewall-cmd --get-default-zone
public
```

- 5.3. Si el paso anterior no establece la zona `public` como la zona predeterminada, corrija la con el siguiente comando:

```
[student@serverb ~]$ sudo firewall-cmd --set-default-zone public
```

5.4. Determine los puertos abiertos listados en la zona de red `public`.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
[sudo] password for student: student
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

6. Agregue el puerto 1001/TCP a la configuración permanente para la zona de red `public` (pública). Confirme su configuración.

6.1. Agregue el puerto 1001/TCP a la zona de red `public` (pública).

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public \
--add-port=1001/tcp
success
```

6.2. Vuelva a cargar la configuración del firewall.

```
[student@serverb ~]$ sudo firewall-cmd --reload
success
```

6.3. Verifique su configuración.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 1001/tcp
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
```

```
source-ports:
icmp-blocks:
rich rules:
```

6.4. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

7. Desde `workstation`, confirme que el servidor web predeterminado en `serverb.lab.example.com` vuelva a `SERVER B` y que el host virtual en `serverb.lab.example.com:1001` vuelva a `VHOST 1`.

7.1. Pruebe acceder al servidor web en `http://serverb.lab.example.com`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
SERVER B
```

7.2. Pruebe acceder al host virtual en `http://serverb.lab.example.com:1001`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001
VHOST 1
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade netsecurity-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish netsecurity-review
```

Esto concluye la sección.

Resumen

- El marco (framework) `netfilter` permite a los módulos del kernel inspeccionar cada paquete que atraviesa el sistema, incluidos todos los paquetes de red entrantes, salientes o reenviados.
- El servicio `firewalld` simplifica la administración al clasificar todo el tráfico de la red en zonas. Cada zona tiene su propia lista de puertos y servicios. La zona `public` se establece como la zona predeterminada.
- El servicio `firewalld` incluye un número de servicios predefinidos. Puede enumerarlos con el comando `firewall-cmd --get-services`.
- La política de SELinux controla estrictamente el tráfico de red al etiquetar los puertos de red. Por ejemplo, el puerto 22/TCP tiene asociada la etiqueta `ssh_port_t`. Cuando un proceso desea escuchar en un puerto, SELinux verifica si la etiqueta asociada puede hacer referencia a la etiqueta de dicho puerto.
- Use el comando `semanage` para agregar, eliminar y modificar etiquetas.

Instalación de Red Hat Enterprise Linux

Meta

Instalar Red Hat Enterprise Linux en servidores y máquinas virtuales.

Objetivos

- Instalar Red Hat Enterprise Linux en un servidor.
- Explicar los conceptos y la arquitectura de Kickstart, crear un archivo Kickstart con el sitio web Kickstart Generator, modificar un archivo Kickstart existente con un editor de texto y verificar su sintaxis con `ksvalidator`, publicar un archivo Kickstart en el instalador e instalar Kickstart en la red.
- Instalar una máquina virtual en su servidor Red Hat Enterprise Linux con la consola web.

Secciones

- Instalación de Red Hat Enterprise Linux (y ejercicio guiado)
- Automatización de la instalación con Kickstart (y ejercicio guiado)
- Instalación y configuración de máquinas virtuales (y cuestionario)

Trabajo de laboratorio

Instalación de Red Hat Enterprise Linux

Instalación de Red Hat Enterprise Linux

Objetivos

Instalar Red Hat Enterprise Linux en un servidor.

Comprensión de los medios de instalación

Red Hat proporciona varias formas distintas de instalación que puede descargar desde el sitio web del Portal de Clientes con su suscripción activa.

- Un archivo de imagen binario en formato ISO 9660 que contiene *Anaconda*, el programa de instalación de Red Hat Enterprise Linux y los repositorios de paquetes de BaseOS y AppStream. Estos repositorios contienen los paquetes necesarios para completar la instalación sin repositorios adicionales.
- Un archivo de imagen "ISO de arranque" que contiene Anaconda y requiere una red configurada para acceder a los repositorios de paquetes disponibles mediante HTTP, FTP o NFS.
- Una imagen QCOW2 que contiene un disco del sistema compilado previamente, listo para implementarse como máquina virtual en la nube o en entornos virtuales empresariales. QCOW2 es el formato de imagen estándar usado por Red Hat con la virtualización basada en KVM.
- Código fuente (instrucciones en lenguaje de programación legibles por humanos) para Red Hat Enterprise Linux. No hay documentación para los DVD de origen. Esta imagen ayuda a compilar o desarrollar su software en función de la versión de Red Hat.

Red Hat Enterprise Linux soporta las siguientes arquitecturas:

- Arquitecturas AMD, Intel y ARM de 64 bits.
- IBM Power Systems (Little Endian, servidores LC y servidores AC).
- IBM Z de 64 bits.

Después de la descarga, cree medios de instalación de arranque según las instrucciones proporcionadas en la sección de referencia.

Creación de imágenes con el generador de imágenes

La herramienta Red Hat Image Builder ayuda a crear imágenes personalizadas de Red Hat Enterprise Linux. Para casos de uso especializados, Image Builder permite a los administradores compilar imágenes de sistema personalizadas para su implementación en plataformas en la nube o entornos virtuales.

Use el comando `composer -cli` o la interfaz de la consola web de Red Hat para acceder a Image Builder.

Instalación manual de Red Hat Enterprise Linux

Por medio del DVD binario o archivo ISO de arranque, los administradores instalan un nuevo sistema RHEL en un servidor bare-metal o en una máquina virtual. El programa Anaconda soporta dos métodos de instalación: manual y automatizada.

- La instalación manual interactúa con el usuario para consultar cómo Anaconda instala y configura el sistema.

- La instalación automatizada usa un archivo *Kickstart* que le dice a Anaconda cómo instalar el sistema.

Instalación de RHEL con la interfaz gráfica

Anaconda se inicia como una aplicación gráfica cuando arranca el sistema desde el DVD binario o el archivo ISO de arranque.

En la pantalla **WELCOME TO RED HAT ENTERPRISE LINUX 9** (BIENVENIDO A RED HAT ENTERPRISE LINUX 9), seleccione el idioma que desea usar durante la instalación y haga clic en el botón **Continue** (Continuar). Este botón también define el idioma predeterminado del sistema después de la instalación. Los usuarios individuales pueden elegir el idioma de preferencia después de la instalación.

Anaconda muestra la ventana **INSTALLATION SUMMARY** (RESUMEN DE LA INSTALACIÓN), la interfaz central para personalizar los parámetros antes de comenzar la instalación.

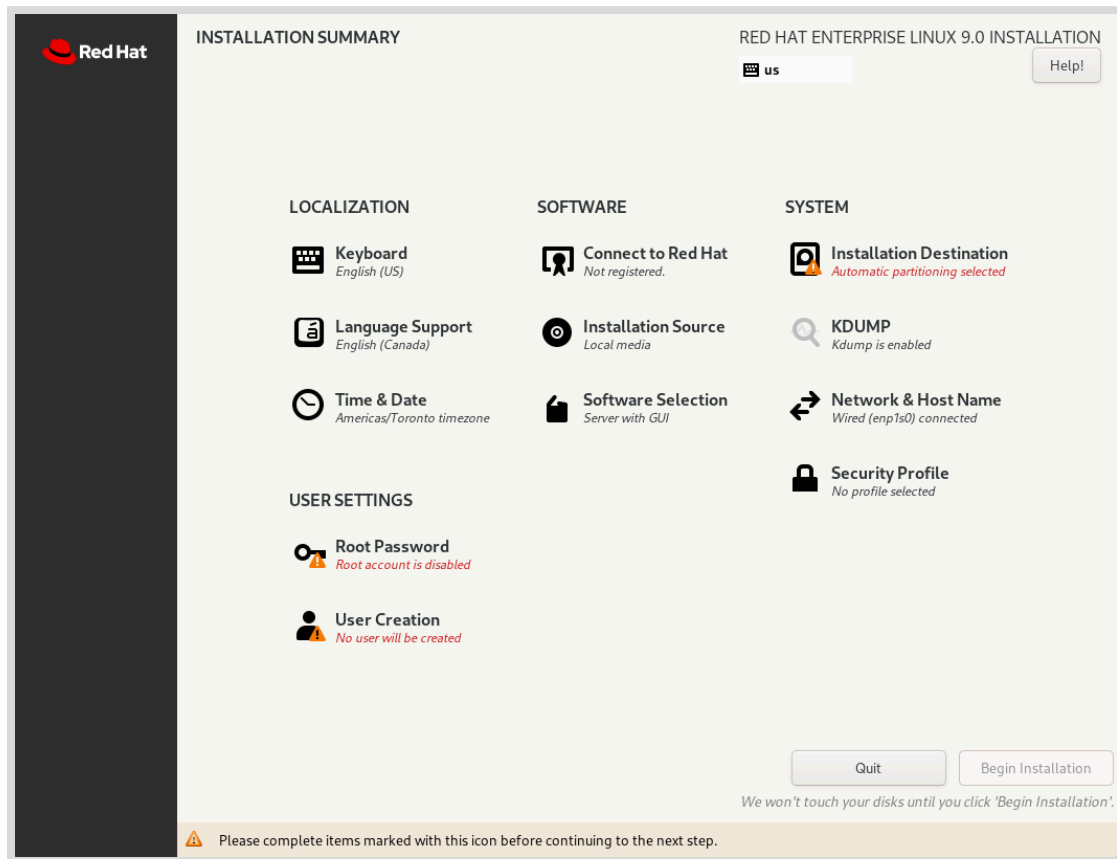


Figura 10.1: Ventana Installation Summary (Resumen de la instalación)

Desde esta ventana, seleccione los iconos en cualquier orden para configurar los parámetros de instalación. Seleccione un ítem para ver o editar. Haga clic en **Done** (Listo) en cualquier ítem para volver a esta pantalla central.

Anaconda marca los ítems obligatorios con un símbolo de advertencia con forma de triángulo y un mensaje. La barra de estado naranja en la parte inferior de la pantalla advierte que complete la información obligatoria antes de que comience la instalación.

Complete los siguientes ítems según sea necesario:

- **Keyboard** (Teclado): agregue configuraciones de teclado adicionales.

- **Language Support** (Compatibilidad de idioma): seleccione los idiomas adicionales que desee instalar.
- **Time & Date** (Hora y fecha): para seleccionar la ciudad del sistema, haga clic en el mapa interactivo o use las listas. Especifique la zona horaria local incluso al usar el *Protocolo de hora de red* (NTP).
- **Connect to Red Hat** (Conectarse a Red Hat): registre el sistema con su cuenta de Red Hat y seleccione el *propósito del sistema*. La función del propósito del sistema permite al proceso de registro asociar automáticamente la suscripción más adecuada al sistema. Primero debe conectarse a la red con el icono **Network & Host Name** (Nombre del host y de la red) para registrar el sistema.
- **Installation Source** (Fuente de instalación): ingrese la ubicación del paquete fuente que Anaconda necesita para la instalación. El campo de fuente de instalación ya hace referencia directa al DVD cuando usa el DVD binario.
- **Software Selection** (Selección de software): seleccione el entorno base que desea instalar y cualquier complemento adicional. El entorno **Minimal Install** (Instalación mínima) instala solamente los paquetes esenciales para ejecutar Red Hat Enterprise Linux.
- **Installation Destination** (Destino de instalación): seleccione y particione los discos donde se instalará Red Hat Enterprise Linux. Para realizar la tarea, el administrador debe entender los esquemas de partición y los criterios de selección del sistema de archivos. El botón de opción predeterminado para la partición automática asigna los dispositivos de almacenamiento seleccionados usando todo el espacio disponible.
- **KDUMP**: la función de volcado de error del kernel, *kdump*, recopila información acerca del estado de la memoria del sistema cuando el kernel se bloquea. Los ingenieros de Red Hat analizan un archivo kdump para identificar la causa del error. Use este ítem de Anaconda para habilitar o deshabilitar kdump.
- **Network & Host Name** (Nombre del host y de la red): las conexiones de red detectadas se enumeran a la izquierda. Seleccione una conexión para visualizar sus detalles. De manera predeterminada, Anaconda ahora activa la red automáticamente. Haga clic en el botón **Configure** (Configurar) para configurar la conexión de red seleccionada.
- **Security Profile** (Perfil de seguridad): al activar un perfil de seguridad, Anaconda aplica restricciones y recomendaciones durante la instalación en función del perfil seleccionado.
- **Root Password** (Contraseña root): el programa de instalación solicita que se configure una contraseña de `root`. La etapa final del proceso de instalación no continuará hasta que defina una contraseña `root`.
- **User Creation** (Creación de usuario): cree una cuenta opcional que no corresponda al usuario `root`. Se recomienda crear una cuenta local de uso general. También puede crear cuentas después de que se complete la instalación.



nota

Al configurar la contraseña de usuario `root`, Red Hat Enterprise Linux 9 ahora habilita una opción para bloquear el acceso del usuario `root` al sistema. También habilita el acceso SSH basado en contraseña para el usuario `root`.

Después de completar la configuración de la instalación y resolver todas las advertencias, haga clic en el botón **Begin Installation** (Comenzar la instalación). Si hace clic en el botón **Quit** (Salir), la instalación se interrumpe sin aplicar los cambios al sistema.

Haga clic en el botón **Reboot** (Reiniciar) cuando la instalación finalice. Anaconda muestra la pantalla **Initial Setup** (Configuración inicial) al instalar un escritorio gráfico. Acepte la información de la licencia y, opcionalmente, registre el sistema en el administrador de suscripciones. Puede omitir el registro del sistema y realizarlo más tarde.

Solución de problemas durante la instalación

Durante la instalación de Red Hat Enterprise Linux 9, Anaconda proporciona dos consolas virtuales. La primera tiene cinco ventanas provistas por el multiplexor del terminal de software `tmux`. Puede acceder a la consola con `Ctrl+Alt+F1`. La segunda consola virtual, que se muestra de forma predeterminada, muestra la interfaz gráfica de Anaconda. Puede acceder con `Ctrl+Alt+F6`.

El terminal `tmux` proporciona un prompt de shell en la segunda ventana en la primera consola virtual. Puede usarlo para ingresar comandos para inspeccionar y solucionar problemas del sistema mientras la instalación continúa. Las otras ventanas muestran mensajes de diagnóstico, registros e información adicional.

En la siguiente tabla, se detallan las combinaciones de teclas para acceder a las consolas virtuales y a las ventanas de terminales `tmux`. En el terminal `tmux`, los atajos de teclado se realizan en dos pasos: presionar y soltar `Ctrl+B`; luego, presionar la tecla numérica de la ventana a la cual desea acceder. En el terminal `tmux`, también puede usar `Alt+Tab` para rotar entre las diferentes ventanas.

Secuencia de teclas	contenidos
<code>Ctrl+Alt+F1</code>	Acceder al multiplexor del terminal de <code>tmux</code> .
<code>Ctrl+B 1</code>	En el terminal <code>tmux</code> , acceder a la página de información principal del proceso de instalación.
<code>Ctrl+B 2</code>	En el terminal <code>tmux</code> , proporcione una shell de root. Anaconda almacena los archivos de registro de instalación en el directorio <code>/tmp</code> .
<code>Ctrl+B 3</code>	En el terminal <code>tmux</code> , muestre el contenido del archivo <code>/tmp/anaconda.log</code> .
<code>Ctrl+B 4</code>	En el terminal <code>tmux</code> , muestre el contenido del archivo <code>/tmp/storage.log</code> .
<code>Ctrl+B 5</code>	En el terminal <code>tmux</code> , muestre el contenido del archivo <code>/tmp/program.log</code> .
<code>Ctrl+Alt+F6</code>	Acceder a la interfaz gráfica de Anaconda.



nota

A los fines de que exista compatibilidad con versiones anteriores de Red Hat Enterprise Linux, las consolas virtuales de `Ctrl+Alt+F2` a `Ctrl+Alt+F5` también presentan shells root durante la instalación.



Referencias

Para obtener más información, consulte *Understanding the various RHEL .iso file* en <https://access.redhat.com/solutions/104063>

Para obtener más información, consulte *Creating a Bootable Installation Medium for RHEL* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_a_standard_rhel_installation/index#assembly_creating-a-bootable-installation-medium_installing-RHEL

Para obtener más información, consulte *Composing A Customized RHEL System Image* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/composing_a_customized_rhel_system_image/index#composer-description_composing-a-customized-rhel-system-image

Para obtener más información, consulte *Performing A Standard RHEL Installation* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_a_standard_rhel_installation/index

► Ejercicio Guiado

Instalación de Red Hat Enterprise Linux

En este ejercicio, reinstalará uno de sus servidores con una instalación mínima de Red Hat Enterprise Linux.

Resultados

- Instalar Red Hat Enterprise Linux 9 manualmente.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start installing-install
```

Instrucciones

- 1. Acceda a la consola de `servera` y reinicie el sistema en los medios de instalación.
 - 1.1. Localice la consola de `servera` para el entorno del aula. Abra la consola.
 - 1.2. Para reiniciar, envíe `Ctrl+Alt+Del` a su sistema usando la entrada del teclado, del menú o la entrada virtual relevantes.
 - 1.3. Cuando aparezca el menú del cargador de arranque, seleccione la entrada de menú **Install Red Hat Enterprise Linux 9**.
- 2. Deje seleccionado el idioma predeterminado y haga clic en el botón **Continue** (Continuar).
- 3. Use la partición automática en el disco `/dev/vda` predeterminado.
 - 3.1. Haga clic en el botón **Installation Destination**.
 - 3.2. Haga clic en el disco `vda` y, luego, en el botón **Done** (Listo) para usar la opción de partición automática predeterminada.
 - 3.3. En la ventana **Installation Options** (Opciones de instalación), haga clic en el botón **Reclaim space** (Recuperar espacio). El disco `/dev/vda` ya tiene particiones y sistemas de archivos de la instalación anterior. Esta selección le permite borrar el disco para la nueva instalación. En la ventana **Reclaim Disk Space** (Recuperar espacio del disco), haga clic en el botón **Delete all** (Eliminar todo) y, luego, en el botón **Reclaim space** (Recuperar espacio).
- 4. Establezca el nombre de host `servera.lab.example.com` para el sistema y verifique la configuración de la interfaz de red.
 - 4.1. Haga clic en el botón **Network & Host Name**.

- 4.2. En el campo **Host Name** (Nombre del host y de la red), ingrese el nombre de host `servera.lab.example.com` y, luego, haga clic en el botón **Apply** (Aplicar).
- 4.3. Haga clic en el botón **Configure** (Configurar) y, luego, en la pestaña **IPv4 Settings** (Configuración de IPv4).
- 4.4. Confirme que los parámetros de red usen la dirección IP `172.25.250.10`, 24 como la máscara de red y `172.25.250.254` como la puerta de enlace y el servidor de nombres. Haga clic en el botón **Save**.
- 4.5. Confirme que la interfaz de red esté habilitada al activar el switch **ON/OFF** (Habilitar/Deshabilitar).
- 4.6. Haga clic en el botón **Done**.
- ▶ 5. Defina el campo **Installation Source** (Fuente) en `http://content.example.com/rhel9.0/x86_64/dvd`.
 - 5.1. Haga clic en el botón **Installation Source**.
 - 5.2. Seleccione la opción **On the network** (En la red).
 - 5.3. En el campo **http://**, escriba la dirección `content.example.com/rhel9.0/x86_64/dvd`.
 - 5.4. Haga clic en **Done**.
- ▶ 6. Seleccione el software requerido para ejecutar una instalación mínima.
 - 6.1. Haga clic en el botón **Software Selection**.
 - 6.2. Seleccione **Minimal Install** (Instalación mínima) de la lista **Base Environment** (Entorno básico).
 - 6.3. Haga clic en el botón **Done**.
- ▶ 7. Establezca **redhat** como contraseña para el usuario **root**.
 - 7.1. Haga clic en el botón **Root Password**.
 - 7.2. Ingrese **redhat** en el campo **Root Password** (Contraseña root).
 - 7.3. Ingrese **redhat** en el campo **Confirm** (Confirmar).
 - 7.4. Haga clic en el botón **Done** (Listo) dos veces porque la contraseña no supera la verificación del diccionario.
- ▶ 8. Agregue el usuario **student** con la contraseña **student**.
 - 8.1. Haga clic en el botón **User Creation**.
 - 8.2. Ingrese **student** en el campo **Full Name** (Nombre completo). El **User name** (Nombre de usuario) completa automáticamente **student** como nombre de usuario.
 - 8.3. Marque **Make this user administrator** (Hacer administrador a este usuario) para que el usuario **student** use el comando **sudo** para ejecutar comandos con el usuario **root**.

- 8.4. Ingrese `student` en el campo **Password** (Contraseña).
 - 8.5. Ingrese `student` en el campo **Confirm password** (Confirmar contraseña).
 - 8.6. Haga clic en el botón **Done** (Listo) dos veces porque la contraseña no supera la verificación del diccionario.
- ▶ 9. Haga clic en el botón **Begin Installation**.
 - ▶ 10. Haga clic en el botón **Reboot** (Reiniciar) cuando la instalación esté completa.
 - ▶ 11. Cuando el sistema muestre el prompt de inicio de sesión, inicie sesión con el usuario `student`.
 - ▶ 12. Después de validar la instalación, restablezca la máquina `servera` desde la página web del entorno del aula.

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish installing-install
```

Esto concluye la sección.

Automatización de la instalación con Kickstart

Objetivos

Explicar los conceptos y la arquitectura de Kickstart, crear un archivo Kickstart con el sitio web Kickstart Generator, modificar un archivo Kickstart existente con un editor de texto y verificar su sintaxis con `ksvalidator`, publicar un archivo Kickstart en el instalador e instalar Kickstart en la red.

Introducción a Kickstart

La función *Kickstart* de Red Hat Enterprise Linux automatiza las instalaciones del sistema. Puede usar archivos de texto Kickstart para configurar la partición del disco, las interfaces de red, la selección de paquetes y personalizar la instalación. El instalador de Anaconda usa archivos Kickstart para realizar una instalación completa sin interacción del usuario. La función Kickstart es similar y usa un archivo de respuesta de configuración desatendido para Microsoft Windows.

Los archivos Kickstart comienzan con una lista de comandos que definen cómo se instalará la máquina de destino. El instalador ignora las líneas de comentarios indicadas por el carácter de signo de número (`#`). Las secciones adicionales comienzan con una directiva, denotada por el carácter del signo de porcentaje (`%`) y terminan en una línea con la directiva `&end`.

En la sección `%packages`, se especifica el software que se debe incluir en la instalación. Los paquetes individuales se especifican por nombre, sin versiones. El carácter ampersand (`@`) denota grupos de paquetes (ya sea por grupo o ID) y los caracteres `@^` denotan grupos de entorno (grupos de grupos de paquetes). Por último, use la sintaxis `@module:stream/profile` para denotar flujos de módulos.

Los grupos tienen componentes obligatorios, predeterminados y opcionales. Normalmente, Kickstart instala los componentes obligatorios y predeterminados. Para excluir un paquete o grupo de paquetes de la instalación, incluya antes un carácter de guión (`-`). Es posible que los paquetes o grupos de paquetes excluidos se instalen si son dependencias obligatorias de otros paquetes obligatorios.

Un archivo de configuración Kickstart generalmente incluye una o más secciones `%pre` y `%post`, que contienen scripts que configuran aún más el sistema. Los scripts `%pre` se excluyen antes de que se haga alguna partición del disco. Por lo general, usted usa scripts `%pre` para inicializar un dispositivo de almacenamiento o de red que requiere el resto de la instalación. Los scripts `%post` se ejecutan una vez que se completa la instalación inicial. Los scripts dentro de las secciones `%pre` y `%post` pueden usar cualquier intérprete disponible en el sistema, incluidos Bash o Python. Siempre que sea posible, debe evitar el uso de una sección `%pre`, ya que los errores que se produzcan en ella pueden ser difíciles de diagnosticar.

Por último, puede especificar tantas secciones como necesite, en cualquier orden. Por ejemplo, puede tener dos secciones `%post` y se interpretarán en orden de aparición.

**nota**

RHEL Image Builder es un método de instalación alternativo a los archivos Kickstart. En lugar de un archivo de texto que proporciona instrucciones de instalación, Image Builder crea una imagen que contiene todos los cambios requeridos que necesita un sistema. RHEL Image Builder puede crear imágenes para nubes públicas, como Amazon Web Services y Microsoft Azure, o nubes privadas, como OpenStack o VMware. Consulte las referencias de esta sección para obtener más información sobre RHEL Image Builder.

Comandos de instalación

Los siguientes comandos de Kickstart configuran la fuente y el método de instalación:

- **url**: especifica la URL que apunta a los medios de instalación.

```
url --url="http://classroom.example.com/content/rhel9.0/x86_64/dvd/"
```

- **repo**: especifica dónde encontrar paquetes adicionales para la instalación. Esta opción debe apuntar a un repositorio DNF válido.

```
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream/
```

- **text**: fuerza una instalación del modo de texto.
- **vnc**: habilita el visor de VNC para que pueda acceder a la instalación gráfica de forma remota a través de VNC.

```
vnc --password=redhat
```

Comandos de dispositivos y particiones

Los siguientes comandos de Kickstart configuran dispositivos y esquemas de partición:

- **clearpart**: elimina particiones del sistema antes de la creación de particiones nuevas.

```
clearpart --all --drives=vda,vdb
```

- **part**: especifica el tamaño, el formato y el nombre de una partición. Obligatorio, a menos que estén presentes los comandos **autopart** o **mount**.

```
part /home --fstype=ext4 --label=homes --size=4096 --maxsize=8192 --grow
```

- **autopart**: crea automáticamente una partición de root, una partición de intercambio (swap) y una partición de arranque adecuada para la arquitectura. En unidades con un tamaño lo suficientemente grande (más de 50 GB), esto también crea una partición de /home.
- **ignoredisk**: evita que Anaconda modifique los discos y es útil junto con el comando **autopart**.

```
ignoredisk --drives=sdz
```

- **boot loader**: define dónde instalar el cargador de arranque. Obligatorio.

```
bootloader --location=mbr --boot-drive=sda
```

- **volgroup, logvol**: crea grupos de volúmenes LVM y volúmenes lógicos.

```
part pv.01 --size=8192
volgroup myvg pv.01
logvol / --vgname=myvg --fstype=xfs --size=2048 --name=rootvol --grow
logvol /var --vgname=myvg --fstype=xfs --size=4096 --name=varvol
```

- **zerombr**: inicializa los discos cuyo formato no se reconoce.

Comandos de red

Los siguientes comandos de Kickstart configuran funciones relacionadas con la red:

- **network**: configura la información de la red para el sistema de destino. Activa los dispositivos de red en el entorno del instalador.

```
network --device=eth0 --bootproto=dhcp
```

- **firewall**: define la configuración del firewall para el sistema de destino.

```
firewall --enabled --service=ssh,http
```

Comandos de ubicación y seguridad

Los siguientes comandos de Kickstart configuran los ajustes de seguridad, idioma y región:

- **lang**: establece el idioma para usar durante la instalación y el idioma predeterminado del sistema instalado. Obligatorio.

```
lang en_US
```

- **keyboard**: establece el tipo de teclado del sistema. Obligatorio.

```
keyboard --vckeymap=us
```

- **timezone**: define la zona horaria, los servidores NTP y si el reloj del hardware usa UTC. Obligatorio.

```
timezone --utc Europe/Amsterdam
```

- **timesource**: habilita o deshabilita NTP. Si habilita NTP, debe especificar servidores o conjuntos (pools) de NTP.

```
timesource --ntp-server classroom.example.com
```

- **authselect**: establece opciones de autenticación. Las opciones reconocidas por **authselect** son válidas para este comando. Consulte **authselect (8)**.

- **rootpw**: define la contraseña inicial de **root**. Obligatorio.

```
rootpw --plaintext redhat
or
rootpw --iscrypted $6$KUnFfrTz08jv.PiH$YlBb0tXBkWoMuRfb0.SpbQ...XDR1UuchoMG1
```

- **selinux**: establece el modo de SELinux para el sistema instalado.

```
selinux --enforcing
```

- **services**: modifica el conjunto de servicios predeterminado que se ejecutan en el objetivo **systemd** predeterminado.

```
services --disabled=network,iptables,ip6tables --enabled=NetworkManager,firewalld
```

- **group, user**: crea un grupo o usuario locales en el sistema.

```
group --name=admins --gid=10001
user --name=jdoe --gecos="John Doe" --groups=admins --password=changeme --
plaintext
```

Comandos varios

Los siguientes comandos de Kickstart configuran el registro del estado de energía del host al finalizar:

- **logging**: este comando define cómo Anaconda realizará el registro durante la instalación.

```
logging --host=loghost.example.com
```

- **firstboot**: si está habilitado, el Agente de configuración se inicia la primera vez que el sistema arranca. Este comando requiere el paquete **initial-setup**.

```
firstboot --disabled
```

- **reboot, poweroff, halt**: especifican la medida final que se tomará cuando finalice la instalación. El valor predeterminado es la opción **halt**.



nota

La mayoría de los comandos de Kickstart tienen varias opciones disponibles. Revise la guía *Comandos y opciones de Kickstart* en las referencias de esta sección para obtener más información.

Ejemplo de un archivo Kickstart

En el siguiente ejemplo, la primera parte del archivo consta de los comandos de instalación, como la partición del disco y la fuente de instalación.

```
#version=RHEL9

# Define system bootloader options
bootloader --append="console=ttyS0 console=ttyS0,115200n8 no_timer_check
    net.ifnames=0 crashkernel=auto" --location=mbr --timeout=1 --boot-drive=vda

# Clear and partition disks
clearpart --all --initlabel
ignoredisk --only-use=vda
zerombr
part / --fstype="xfs" --ondisk=vda --size=10000

# Define installation options
text
repo --name="appstream" --baseurl="http://classroom.example.com/content/rhel9.0/
x86_64/dvd/AppStream/"
url --url="http://classroom.example.com/content/rhel9.0/x86_64/dvd/"

# Configure keyboard and language settings
keyboard --vckeymap=us
lang en_US

# Set a root password, authselect profile, and selinux policy
rootpw --plaintext redhat
authselect select sssd
selinux --enforcing
firstboot --disable

# Enable and disable system services
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chronyd"

# Configure the system timezone and NTP server
timezone America/New_York --utc
timesource --ntp-server classroom.example.com
```

La segunda parte de un archivo Kickstart contiene la sección `%packages` con detalles sobre qué paquetes y grupos de paquetes deben instalarse y qué paquetes no deben instalarse.

```
%packages

@core
chrony
cloud-init
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
-plymouth

%end
```

La última parte del archivo Kickstart contiene un script de instalación `%post`.

```
%post

echo "This system was deployed using Kickstart on $(date)" > /etc/motd

%end
```

También puede especificar un script de Python con la opción `--interpreter`.

```
%post --interpreter="/usr/libexec/platform-python"

print("This this a line of text printed with python")

%end
```



nota

En un archivo Kickstart, si no se determinan los valores obligatorios, el instalador solicitará una respuesta interactivamente o abortará la instalación por completo.

Pasos de instalación Kickstart

Siga estos pasos para automatizar la instalación de Red Hat Enterprise Linux con la función Kickstart:

1. Crear un archivo Kickstart.
2. Publicar el archivo Kickstart para que el instalador de Anaconda pueda acceder a él.
3. Arrancar el instalador de Anaconda y apuntarlo al archivo Kickstart.

Creación de un archivo Kickstart

Puede usar cualquiera de estos métodos para crear un archivo Kickstart:

- Usar el sitio web de Kickstart Generator.
- Usar un editor de textos.

El sitio de Kickstart Generator en <https://access.redhat.com/labs/kickstartconfig> presenta cuadros de diálogo para las entradas del usuario y crea un archivo de configuración Kickstart con las opciones del usuario. Cada cuadro de diálogo corresponde a los ítems del instalador de Anaconda que se pueden configurar.

Figura 10.2: Generador de Kickstart del portal de clientes de Red Hat

Crear un archivo Kickstart desde cero es complejo, por lo que debe comenzar por intentar editar un archivo existente. Cada instalación crea un archivo `/root/anaconda-ks.cfg` que contiene las directivas Kickstart que se usan en la instalación. Este archivo es un buen punto de partida para crear un nuevo archivo Kickstart.

La utilidad `ksvalidator` revisa errores de sintaxis en un archivo Kickstart. Asegura que las palabras clave y las opciones se usen adecuadamente, pero no confirma las rutas URL, ni los paquetes individuales, ni los grupos, ni ninguna parte de los scripts `%post` o `%pre`. Por ejemplo, si la directiva `firewall --disabled` está mal escrita, el comando `ksvalidator` podría producir uno de los siguientes errores:

```
[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
The following problem occurred on line 10 of the kickstart file:

Unknown command: firewall

[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
The following problem occurred on line 10 of the kickstart file:

no such option: --dsabled
```

La utilidad `ksverdiff` muestra las diferencias de sintaxis entre las diferentes versiones del sistema operativo. Por ejemplo, el siguiente comando muestra los cambios de sintaxis de Kickstart entre RHEL 8 y RHEL 9:

```
[user@host ~]$ ksverdiff -f RHEL8 -t RHEL9
The following commands were removed in RHEL9:
device deviceprobe dmraid install multipath

The following commands were deprecated in RHEL9:
autostep btrfs method

The following commands were added in RHEL9:
timesource
...output omitted...
```

El paquete `pykickstart` proporciona las utilidades `ksvalidator` y `ksverdiff`.

Publicación del archivo Kickstart para Anaconda

Para que el archivo Kickstart esté disponible para el instalador, colóquelo en una de las siguientes ubicaciones:

- un servidor de red disponible en el momento de la instalación mediante FTP, HTTP o NFS;
- un disco USB o CD-ROM disponibles;
- un disco duro local en el sistema.

El instalador debe acceder al archivo Kickstart para iniciar una instalación automatizada. Por lo general, el archivo está disponible a través de un servidor FTP, web o NFS. Los servidores de red facilitan el mantenimiento del archivo Kickstart porque los cambios se pueden hacer solo una vez y, luego, se usan para instalaciones futuras.

También es conveniente proporcionar archivos Kickstart en USB o CD-ROM. El archivo Kickstart se puede integrar en los medios de arranque usados para iniciar la instalación. Sin embargo, cuando se cambia el archivo Kickstart, debe generar nuevos medios de instalación.

Si proporciona el archivo Kickstart en un disco local, puede volver a compilar rápidamente un sistema.

Arrancar Anaconda y apuntarlo al archivo Kickstart

Una vez elegido un método de Kickstart, se le indica al instalador dónde ubicar el archivo Kickstart al pasar el parámetro `inst.ks=LOCATION` al kernel de instalación.

Considere los siguientes ejemplos:

- `inst.ks=http://server/dir/file`
- `inst.ks=ftp://server/dir/file`
- `inst.ks=nfs:server:/dir/file`
- `inst.ks=hd:device:/dir/file`
- `inst.ks=cdrom:device`

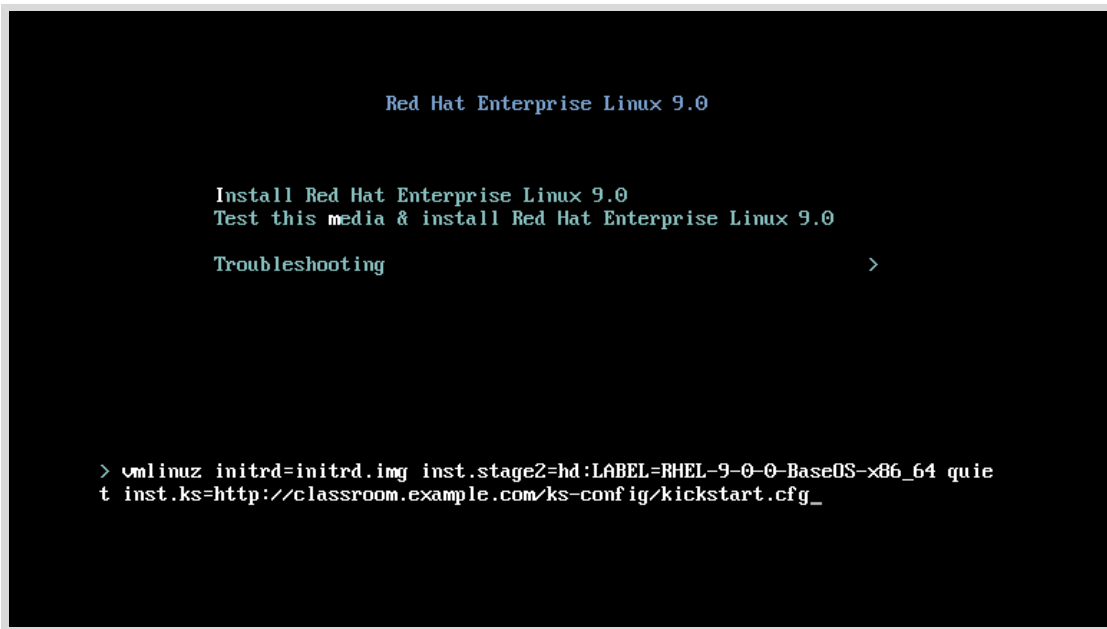


Figura 10.3: Especificación de la ubicación del archivo Kickstart durante la instalación

En el caso de instalaciones en máquinas virtuales con **Virtual Machine Manager** (Administrador de máquinas virtuales) o **virt -manager**, la URL Kickstart se puede especificar en una casilla en **URL Options** (Opciones de URL). Cuando realice una instalación en máquinas físicas, realice el arranque usando medios de instalación y presione la tecla **Tab** para interrumpir el proceso de arranque. Agregue un parámetro `inst.ks=LOCATION` para el kernel de instalación.



Referencias

Capítulo *Kickstart installation basics* en *Performing an advanced RHEL installation* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_an_advanced_rhel_installation/index#performing_an_automated_installation_using_kickstart

Kickstart commands and options reference en *Performing an advanced RHEL installation* en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_an_advanced_rhel_installation/index#kickstart-commands-and-options-reference_installing-rhel-as-an-experienced-user

Boot options capítulo en *Performing an advanced RHEL installation* en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-and-advanced-boot-options_installing-rhel-as-an-experienced-user

Composing a customized RHEL system image en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/composing_a_customized_rhel_system_image/index

► Ejercicio Guiado

Automatización de la instalación con Kickstart

En este trabajo de laboratorio, crea un archivo kickstart y valida la sintaxis.

Resultados

- Crear un archivo kickstart.
- Validar la sintaxis del archivo Kickstart.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start installing-kickstart
```

Instrucciones

- 1. Inicie sesión en `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Cree el archivo `/home/student/kickstart.cfg` copiando el contenido del archivo `/root/anaconda-ks.cfg` con acceso privilegiado.

```
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

- 3. Realice los siguientes cambios en el archivo `/home/student/kickstart.cfg`.

- 3.1. Comente el comando de reinicio:

```
#reboot
```

- 3.2. Modifique los comandos `repo` para especificar los repositorios de `BaseOS` y `AppStream` del servidor `classroom`:

```
repo --name="BaseOS" --baseurl="http://classroom.example.com/content/rhel9.0/
x86_64/dvd/BaseOS/"
repo --name="Appstream" --baseurl="http://classroom.example.com/content/rhel9.0/
x86_64/dvd/AppStream/"
```

- 3.3. Modifique el comando `url` para especificar la fuente de instalación HTTP del servidor `classroom`:

```
url --url="http://classroom.example.com/content/rhel9.0/x86_64/dvd/"
```

- 3.4. Comente el comando `network`:

```
#network --bootproto=dhcp --device=link --activate
```

- 3.5. Modifique el comando `rootpw` para establecer la contraseña de usuario de `root` en `redhat`.

```
rootpw --plaintext redhat
```

- 3.6. Modifique el comando `authselect` para configurar el servicio `sssd` como la fuente de identidad y autenticación.

```
authselect select sssd
```

- 3.7. Modifique el comando `services` para que se vea de la siguiente manera:

```
services --disabled="kdump,rhsmcertd" --enabled="sshd, rngd, chronyd"
```

- 3.8. Comente los comandos `part` y agregue el comando `autopart`:

```
# Disk partitioning information
ignoredisk --only-use=vda
#part biosboot --fstype="biosboot" --size=1
#part /boot/efi --fstype="efi" --size=100 --fsoptions="..."
#part / --fstype="xfs"
autopart
```

- 3.9. Elimine todo el contenido entre la sección `%post` y su directiva `%end`. Agregue la línea `echo "Kickstarted on $(date)" >> /etc/issue`.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

- 3.10. Modifique la sección `%packages` para incluir solo el siguiente contenido:

```
%packages
@core
chrony
```

```
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
httpd
-plymouth
%end
```

3.11. Guarde y salga del archivo.

- ▶ 4. Valide el archivo `kickstart.cfg` en busca de errores de sintaxis. Si no hay errores, entonces el comando no devuelve ninguna salida.

```
[student@servera ~]$ ksvalidator kickstart.cfg
```

- ▶ 5. Copie el archivo `kickstart.cfg` en el directorio `/var/www/html/ks-config`.

```
[student@servera ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
[sudo] password for student: student
```

- ▶ 6. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish installing-kickstart
```

Esto concluye la sección.

Instalación y configuración de máquinas virtuales

Objetivos

Instalar una máquina virtual en su servidor Red Hat Enterprise Linux con la consola web.

Presentación de la virtualización de KVM

La virtualización es una característica que permite que una sola máquina física se divida en varias *máquinas virtuales (VM)*, cada una de las cuales puede ejecutar un sistema operativo independiente.

Red Hat Enterprise Linux soporta la *máquina virtual basada en el kernel (KVM)*, una solución de virtualización completa compilada como parte del kernel de Linux. KVM puede ejecutar múltiples sistemas operativos huésped Windows y Linux.

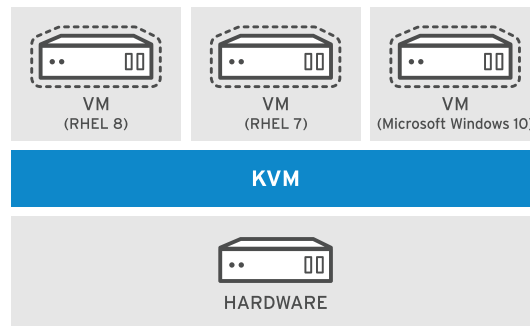


Figura 10.4: Virtualización de KVM

En Red Hat Enterprise Linux, puede administrar KVM desde la línea de comandos con el comando `virsh` o gráficamente con la herramienta de máquinas virtuales de la consola web.

La tecnología de máquinas virtuales de KVM está disponible en todos los productos Red Hat, que van desde instancias físicas independientes de Red Hat Enterprise Linux hasta Red Hat OpenStack Platform:

- Los sistemas de hardware físico ejecutan Red Hat Enterprise Linux para proporcionar la virtualización de KVM. Red Hat Enterprise Linux normalmente es un *thick host*, un sistema que soporta máquinas virtuales y que, al mismo tiempo, presta otros servicios locales y de red, aplicaciones y funciones de administración.
- *Red Hat Virtualization (RHV)* proporciona una interfaz web centralizada que pueden usar los administradores para administrar una infraestructura virtual completa. Incluye características avanzadas, como la migración de KVM, la redundancia y la alta disponibilidad. *Red Hat Virtualization Hypervisor* es una versión optimizada de Red Hat Enterprise Linux dedicada específicamente al aprovisionamiento y al soporte de máquinas virtuales.
- *Red Hat OpenStack Platform (RHOSP)* es la base para crear, implementar y escalar una nube pública o privada.
- *Red Hat OpenShift Virtualization* incluye componentes de RHV para permitir la implementación de contenedores en equipos sin sistema operativo.

En sistemas con SELinux habilitado, sVirt aísla a los huéspedes y al hipervisor. Cada proceso de la máquina virtual se etiqueta y se asigna automáticamente un nivel único, con los archivos de disco virtual asociados con etiquetas coincidentes.

Tipos de máquinas virtuales

Hay dos tipos principales de máquinas virtuales disponibles para instalar el sistema operativo huésped. El más antiguo de ellos usa el conjunto de chips i440FX y proporciona puentes ISA y PCI, un controlador IDE y adaptadores VGA y Ethernet. El tipo de máquina Q35 es el más nuevo de los dos y debe usarse en la mayoría de los casos. Soporta interfaces de hardware virtuales más nuevas, como PCIe y SATA, e incluye un controlador de bus SM.

En Red Hat Enterprise Linux 8 y versiones posteriores, se proporciona el soporte de UEFI y arranque seguro para máquinas virtuales mediante Open Virtual Machine Firmware (OVMF), en el paquete `edk2-ovmf`.

Sistemas operativos huésped soportados

La lista de sistemas operativos huéspedes soportados de Red Hat varía según el producto en uso. Los productos RHV, RHOSP y OpenShift Virtualization incluyen controladores certificados del programa de validación de virtualización de servidores (SVVP), lo que permite un mayor número de sistemas operativos huéspedes de Windows soportados. Un sistema RHEL que ejecuta KVM solo tiene soporte de SVVP para suscripciones específicas.

No hay reglas generales para determinar si un sistema operativo huésped cuenta con soporte; revise uno de los siguientes documentos:

- RHEL con KVM: Sistemas operativos huésped certificados para Red Hat Enterprise Linux con KVM [<https://access.redhat.com/articles/973133>]
- Virtualización de RHV, RHOSP y OpenShift: Sistemas operativos huésped certificados en Red Hat OpenStack Platform, Red Hat Virtualization y OpenShift Virtualization [<https://access.redhat.com/articles/973163>]

Configuración de un sistema físico Red Hat Enterprise Linux como host de virtualización

Los administradores pueden configurar un sistema Red Hat Enterprise Linux como host de virtualización para poder realizar tareas de desarrollo, pruebas o capacitación, o cuando se necesite trabajar en múltiples sistemas operativos simultáneamente.

Instalación de las herramientas de virtualización

Instale el grupo de paquetes DNF `Virtualization Host` para preparar un sistema para que se convierta en un host de virtualización.

```
[root@host ~]# dnf group list | grep -i virt
Virtualization Host
[root@host ~]# dnf group info "Virtualization Host"
...output omitted...
Environment Group: Virtualization Host
Description: Minimal virtualization host.
Mandatory Groups:
  Base
  Core
  Standard
```

```

Virtualization Hypervisor
Virtualization Tools
Optional Groups:
  Debugging Tools
  Network File System Client
  Remote Management for Linux
  Virtualization Platform

```

```

[root@host ~]# dnf group install "Virtualization Host"
...output omitted...

```

Verificación de los requisitos del sistema para la virtualización

KVM requiere un procesador Intel con las extensiones Intel VT-x e Intel 64 para los sistemas basados en x86, o un procesador AMD con las extensiones AMD-V y AMD64. Para verificar su hardware y los requisitos del sistema, use el comando `virt-host-validate`.

```

[root@host ~]# virt-host-validate
QEMU: Checking for hardware virtualization           : PASS
QEMU: Checking if device /dev/kvm exists              : PASS
QEMU: Checking if device /dev/kvm is accessible       : PASS
QEMU: Checking if device /dev/vhost-net exists         : PASS
QEMU: Checking if device /dev/net/tun exists           : PASS
QEMU: Checking for cgroup 'cpu' controller support    : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup 'blkio' controller support  : PASS
QEMU: Checking for device assignment IOMMU support    : PASS
QEMU: Checking for secure guest support                : PASS

```

El sistema debe aprobar todos los ítems de validación para funcionar como un host KVM.

Instalación de una máquina virtual desde la línea de comandos

Instale el paquete `virt-install`, luego, use el comando `virt-install` para implementar una máquina virtual desde la línea de comandos. Hay muchas fuentes de instalación soportadas, como ISO, NFS, HTTP y FTP. El siguiente ejemplo usa la opción `--cdrom` del comando `virt-install` para instalar el huésped desde una imagen ISO.

```

[root@host ~]# virt-install --name demo --memory 4096 --vcpus 2 --disk size=40 \
--os-type linux --cdrom /root/rhel.iso
...output omitted...

```

Administración de máquinas virtuales con la consola web

El paquete `libvirt-client` proporciona el comando `virsh`, que se usa para administrar máquinas virtuales desde la línea de comandos. Sin embargo, puede crear y administrar máquinas virtuales gráficamente con el complemento (plug-in) para máquinas virtuales de la consola web.

Instale el paquete `cockpit-machines` para agregar el menú **Virtual Machines (Máquinas virtuales)** en la consola web.

```
[root@host ~]# dnf install cockpit-machines
```

Si la consola web no se está ejecutando, iníciela y habilítela.

```
[root@host ~]# systemctl enable --now cockpit.socket
```

En un navegador web que se ejecuta en la máquina local, diríjase a `https://localhost/9090` e inicie sesión en la consola web. Cambie al modo administrativo e ingrese su contraseña, si es necesario.

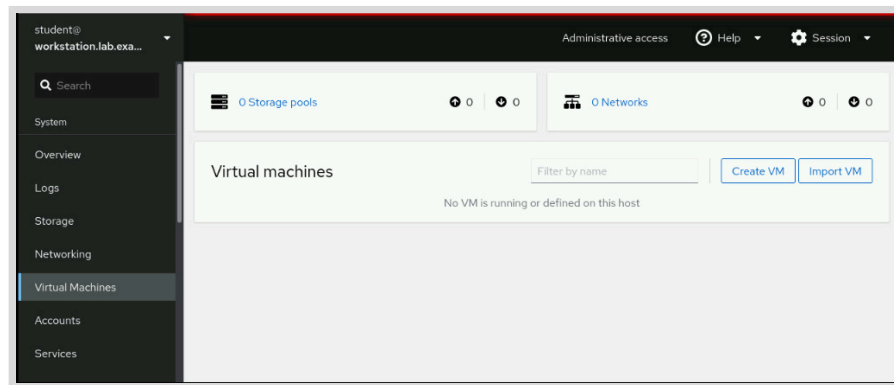


Figura 10.5: Administración de máquinas virtuales en la consola web

Para crear una nueva máquina virtual con la consola web, acceda al menú **Virtual Machines (Máquinas virtuales)**. Desde allí, haga clic en **Create VM** (Crear máquina virtual) e ingrese la configuración de la máquina virtual en la ventana **Create New Virtual Machine** (Crear nueva máquina virtual). Si es la primera vez que usa la consola web después de instalar el complemento (plug-in) para máquinas virtuales, debe reiniciar el sistema para iniciar correctamente la virtualización `libvirt`.

Create new virtual machine

Name: demo

Connection: ☒ System ☐ Session

Installation type: Local install media (ISO image or distro install tree)

Installation source: /root/rhel.iso

Operating system: Red Hat Enterprise Linux 9.0 (Plow)

Storage: Create new volume

Size: 50 GiB

Memory: 4 GiB
Up to 5.5 GiB available on the host

Immediately start VM: ☒

Create Cancel

Figura 10.6: Creación de máquinas virtuales en la consola web

- **Name** (Nombre) establece un nombre de *dominio* para la configuración de la máquina virtual. Este nombre no está relacionado con el nombre de host que le da a la máquina virtual durante la instalación.
- **Installation type** (Tipo de instalación) es el método para acceder a los medios de instalación. Las opciones incluyen el sistema de archivos local o una URL de HTTPS, FTP, NFS o PXE.
- **Installation source** (Fuente de instalación) proporciona la ruta de la fuente de instalación.
- **Operating system** (Sistema operativo) define el sistema operativo de la máquina virtual. La capa de virtualización presenta una emulación de hardware compatible con el sistema operativo elegido.
- **Storage** (Almacenamiento) define si se debe crear un nuevo volumen de almacenamiento o usar uno existente.
- **Size** (Tamaño) es el tamaño del disco al crear un nuevo volumen. Asocie discos adicionales con la máquina virtual después de la instalación.
- **Memory** (Memoria) es la cantidad de RAM disponible para la máquina virtual nueva.
- **Immediately start VM** (Iniciar máquina virtual de inmediato) indica si la máquina virtual se debe iniciar inmediatamente después de hacer clic en **Create** (Crear).

Haga clic en **Create** (Crear) para crear la máquina virtual y en **Install** (Instalar) para iniciar la instalación del sistema operativo. La consola web muestra la consola de la máquina virtual desde la que puede instalar el sistema.



Referencias

Para obtener más información, consulte la guía *Configuring and managing virtualization* en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/configuring_and_managing_virtualization/index

¿Qué es la virtualización?

<https://www.redhat.com/en/topics/virtualization/what-is-virtualization>

Sistemas operativos huésped certificados para Red Hat Enterprise Linux con KVM

<https://access.redhat.com/articles/973133>

Sistemas operativos huésped certificados en Red Hat OpenStack Platform, Red Hat Virtualization y OpenShift Virtualization

<https://access.redhat.com/articles/973163>

► Cuestionario

Instalación y configuración de máquinas virtuales

Elija las respuestas correctas para las siguientes preguntas:

- 1. **¿Qué paquete le permite seleccionar el firmware OVMF para una máquina virtual?**
 - a. open-firmware
 - b. edk2-ovmf
 - c. core-ovmf
 - d. ovmf
 - e. virt-open-firmware

- 2. **¿Qué dos componentes se requieren para configurar su sistema como un host de virtualización y administrar máquinas virtuales con la consola web? (Elija dos opciones).**
 - a. El grupo de paquetes de Virtualization Host
 - b. El grupo de paquetes de openstack
 - c. El paquete cockpit-machines
 - d. El grupo de paquetes de Virtualization Platform
 - e. El módulo DNF kvm
 - f. El paquete cockpit-virtualization

- 3. **¿Qué comando verifica si su sistema soporta la virtualización?**
 - a. grep kvm /proc/cpuinfo
 - b. virsh validate
 - c. virt-host-validate
 - d. rhv-validate
 - e. cockpit-validate

- 4. **¿Qué dos herramientas puede usar para iniciar y detener sus máquinas virtuales en un sistema Red Hat Enterprise Linux? (Elija dos opciones).**
 - a. vmctl
 - b. libvirtd
 - c. virsh
 - d. neutron
 - e. consola web

► Solución

Instalación y configuración de máquinas virtuales

Elija las respuestas correctas para las siguientes preguntas:

- 1. **¿Qué paquete le permite seleccionar el firmware OVMF para una máquina virtual?**
 - a. open-firmware
 - b. edk2-ovmf
 - c. core-ovmf
 - d. ovmf
 - e. virt-open-firmware

- 2. **¿Qué dos componentes se requieren para configurar su sistema como un host de virtualización y administrar máquinas virtuales con la consola web? (Elija dos opciones).**
 - a. El grupo de paquetes de Virtualization Host
 - b. El grupo de paquetes de openstack
 - c. El paquete cockpit-machines
 - d. El grupo de paquetes de Virtualization Platform
 - e. El módulo DNF kvm
 - f. El paquete cockpit-virtualization

- 3. **¿Qué comando verifica si su sistema soporta la virtualización?**
 - a. `grep kvm /proc/cpuinfo`
 - b. `virsh validate`
 - c. `virt-host-validate`
 - d. `rhv-validate`
 - e. `cockpit-validate`

- 4. **¿Qué dos herramientas puede usar para iniciar y detener sus máquinas virtuales en un sistema Red Hat Enterprise Linux? (Elija dos opciones).**
 - a. `vmctl`
 - b. `libvirtd`
 - c. `virsh`
 - d. `neutron`
 - e. consola web

► Trabajo de laboratorio

Instalación de Red Hat Enterprise Linux

En este trabajo de laboratorio, creará un archivo Kickstart y lo pondrá a disposición del instalador.

Resultados

- Crear un archivo kickstart.
- Poner el archivo kickstart a disposición del instalador.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start installing-review
```

Instrucciones

Prepare un archivo kickstart en la máquina `serverb` según lo especificado y póngalo a disposición en la dirección `http://serverb.lab.example.com/ks-config/kickstart.cfg`.

1. En la máquina `serverb`, copie el archivo kickstart `/root/anaconda-ks.cfg` en el archivo kickstart `/home/student/kickstart.cfg` para que el usuario `student` pueda editarlo.
2. Realice cambios en el archivo kickstart `/home/student/kickstart.cfg`.
 - Comente el comando `reboot`.
 - Modifique el comando `repo` para los repositorios de BaseOS y AppStream. Modifique el comando `repo` para que el repositorio de BaseOS use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS/`. Modifique el comando `repo` para que el repositorio de AppStream use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream/`.
 - Cambie el comando `url` para usar `http://classroom.example.com/content/rhel9.0/x86_64/dvd/` como la fuente de instalación.
 - Comente el comando `network`.
 - Cambie el comando `rootpw` para definir `redhat` como la contraseña de usuario `root`.
 - Modifique el comando `authselect` para configurar el servicio `sssd` como la fuente de identidad y autenticación.
 - Modifique el comando `services` para deshabilitar los servicios `kdump` y `rhsmcertd` y habilitar los servicios `sshd`, `rngd` y `chronyd`.
 - Elimine los comandos `part` y agregue el comando `autopart`.

- Simplifique la sección `%post` para que solo ejecute un script para agregar el texto `Kickstarted on DATE` en el final del archivo `/etc/issue`. Use el comando `date` para insertar la fecha sin opciones adicionales.
 - Simplifique la sección `%package` de la siguiente manera: incluya los paquetes `@core`, `chrony`, `dracut-config-generic`, `dracut-norescue`, `firewalld`, `grub2`, `kernel`, `rsync`, `tar` y `httpd`. Asegúrese de que no se instale el paquete `plymouth`.
3. Valide la sintaxis del archivo Kickstart `kickstart.cfg`.
 4. Ponga a disposición el archivo `/home/student/kickstart.cfg` en la dirección `http://serverb.lab.example.com/ks-config/kickstart.cfg`.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade installing-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish installing-review
```

Esto concluye la sección.

► Solución

Instalación de Red Hat Enterprise Linux

En este trabajo de laboratorio, creará un archivo Kickstart y lo pondrá a disposición del instalador.

Resultados

- Crear un archivo kickstart.
- Poner el archivo kickstart a disposición del instalador.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start installing-review
```

Instrucciones

Prepare un archivo kickstart en la máquina `serverb` según lo especificado y póngalo a disposición en la dirección `http://serverb.lab.example.com/ks-config/kickstart.cfg`.

1. En la máquina `serverb`, copie el archivo `kickstart /root/anaconda-ks.cfg` en el archivo `kickstart /home/student/kickstart.cfg` para que el usuario `student` pueda editarlo.
 - 1.1. Inicie sesión en la máquina `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. En la máquina `serverb`, copie el archivo `/root/anaconda-ks.cfg` en el archivo `/home/student/kickstart.cfg`.

```
[student@serverb ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

2. Realice cambios en el archivo `kickstart /home/student/kickstart.cfg`.
 - Comente el comando `reboot`.
 - Modifique el comando `repo` para los repositorios de BaseOS y AppStream. Modifique el comando `repo` para que el repositorio de BaseOS use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS/`. Modifique el comando `repo` para que el repositorio de AppStream use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream/`.

- Cambie el comando `url` para usar `http://classroom.example.com/content/rhel9.0/x86_64/dvd/` como la fuente de instalación.
- Comente el comando `network`.
- Cambie el comando `rootpw` para definir `redhat` como la contraseña de usuario `root`.
- Modifique el comando `authselect` para configurar el servicio `sssd` como la fuente de identidad y autenticación.
- Modifique el comando `services` para deshabilitar los servicios `kdump` y `rhsmcertd` y habilitar los servicios `sshd`, `rngd` y `chronyd`.
- Elimine los comandos `part` y agregue el comando `autopart`.
- Simplifique la sección `%post` para que solo ejecute un script para agregar el texto `Kickstarted on DATE` en el final del archivo `/etc/issue`. Use el comando `date` para insertar la fecha sin opciones adicionales.
- Simplifique la sección `%package` de la siguiente manera: incluya los paquetes `@core`, `chrony`, `dracut-config-generic`, `dracut-norescue`, `firewalld`, `grub2`, `kernel`, `rsync`, `tar` y `httpd`. Asegúrese de que no se instale el paquete `plymouth`.

2.1. Comente la directiva `reboot` (reiniciar).

```
#reboot
```

2.2. Modifique el comando `repo` para los repositorios de BaseOS y AppStream. Modifique el comando `repo` para que el repositorio de BaseOS use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS/`. Modifique el comando `repo` para que el repositorio de AppStream use la dirección `http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream/`.

```
repo --name="BaseOS" --baseurl="http://classroom.example.com/content/rhel9.0/x86_64/dvd/BaseOS/"
repo --name="Appstream" --baseurl="http://classroom.example.com/content/rhel9.0/x86_64/dvd/AppStream/"
```

2.3. Cambie el comando `url` para especificar los medios de fuentes de instalación HTTP proporcionados por la máquina `classroom`.

```
url --url="http://classroom.example.com/content/rhel9.0/x86_64/dvd/"
```

2.4. Comente el comando `network`.

```
#network --bootproto=dhcp --device=link --activate
```

2.5. Modifique el comando `rootpw` para definir `redhat` como la contraseña de usuario `root`.

```
rootpw --plaintext redhat
```

- 2.6. Modifique el comando `authselect` para configurar el servicio `sssd` como la fuente de identidad y autenticación.

```
authselect select sssd
```

- 2.7. Simplifique el comando `services` para que se vea de la siguiente manera.

```
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chronyd"
```

- 2.8. Comente los comandos `part` y agregue el comando `autopart`:

```
# Disk partitioning information
ignoredisk --only-use=vda
#part biosboot --fstype="biosboot" --size=1
#part /boot/efi --fstype="efi" --size=100 --
fsoptions="defaults,uid=0,gid=0,umask=077,shortname=winnt"
#part / --fstype="xfs" --size=10137 --label=root
autopart
```

- 2.9. Elimine todo el contenido entre la sección `%post` y `%end`. Agregue la línea `echo "Kickstarted on $(date)" >> /etc/issue`.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

- 2.10. Simplifique la especificación del paquete para que se vea exactamente de la siguiente manera.

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
httpd
-plymouth
%end
```

3. Valide la sintaxis del archivo Kickstart `kickstart.cfg`.

- 3.1. Valide el archivo `kickstart` en busca de errores de sintaxis.

```
[student@serverb ~]$ ksvvalidator kickstart.cfg
```

4. Ponga a disposición el archivo `/home/student/kickstart.cfg` en la dirección `http://serverb.lab.example.com/ks-config/kickstart.cfg`.

- 4.1. Copie el archivo `kickstart.cfg` en el directorio `/var/www/html/ks-config/`.

```
[student@serverb ~]$ sudo cp kickstart.cfg /var/www/html/ks-config/  
[sudo] password for student: student
```

- 4.2. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade installing-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish installing-review
```

Esto concluye la sección.

Resumen

- El DVD binario de RHEL 9 incluye Anaconda y todos los repositorios necesarios para la instalación.
- El archivo ISO de arranque de RHEL 9 incluye el instalador de Anaconda, que permite acceder a los repositorios por medio de la red durante la instalación.
- El sistema Kickstart puede realizar instalaciones desatendidas.
- Los archivos Kickstart se pueden crear por medio del sitio web de Kickstart Generator o al copiar y editar `/root/anaconda-ks.cfg`.
- El grupo de paquetes de `DNF Virtualization Host` proporciona los paquetes para que un sistema RHEL se convierta en un host de virtualización.
- El paquete `cockpit-machines` agrega el menú **Virtual Machines (Máquinas virtuales)** en Cockpit.

Ejecución de contenedores

Meta

Obtener, ejecutar y administrar servicios livianos simples como contenedores en un único servidor de Red Hat Enterprise Linux.

Objetivos

- Explicar los conceptos de contenedores y las tecnologías centrales (core) para crear, almacenar y ejecutar contenedores.
- Analizar las herramientas de administración de contenedores para usar registros para almacenar y recuperar imágenes, y para implementar, consultar y acceder a contenedores.
- Proporcionar almacenamiento persistente para los datos del contenedor al compartir el almacenamiento desde el host del contenedor y configurar una red de contenedores.
- Configurar un contenedor como servicio systemd y configurar un servicio de contenedor para que se inicie en el momento del arranque.

Secciones

- Conceptos de contenedores (y cuestionario)
- Implementación de contenedores (y ejercicio guiado)
- Administración del almacenamiento de contenedores y los recursos de red (y ejercicio guiado)
- Administración de contenedores como servicios de sistemas (y ejercicio guiado)

Trabajo de laboratorio

Ejecución de contenedores

Conceptos de contenedores

Objetivos

Explicar los conceptos de contenedores y las tecnologías centrales (core) para crear, almacenar y ejecutar contenedores.

Tecnología de contenedores

Por lo general, las aplicaciones de software dependen de otras librerías, archivos de configuración o servicios proporcionados por el entorno de tiempo de ejecución. En general, el entorno de tiempo de ejecución para una aplicación de software se instala en un sistema operativo que se ejecuta en un host físico o una máquina virtual. A continuación, los administradores instalan las dependencias de la aplicación en la parte superior del sistema operativo.

En Red Hat Enterprise Linux, los sistemas de paquetes como RPM se usan para ayudar a administrar las dependencias de las aplicaciones. Cuando instala el paquete `httpd`, el sistema RPM garantiza que también se instalen las librerías y las demás dependencias correspondientes para ese paquete.

El principal inconveniente de las aplicaciones de software implementadas tradicionalmente es que estas dependencias están entremezcladas con el entorno de tiempo de ejecución. Una aplicación puede necesitar versiones de software anteriores o posteriores al software provisto con el sistema operativo. De manera similar, dos aplicaciones en el mismo sistema pueden requerir versiones diferentes e incompatibles del mismo software.

Una forma de resolver estos conflictos es empaquetar e implementar la aplicación como un *contenedor*. Un contenedor es un conjunto de uno o más procesos que están aislados del resto del sistema. Los contenedores de software proporcionan una forma de empaquetar aplicaciones y simplificar la implementación y la administración.

Piense en un contenedor de transporte físico. Un contenedor de envío es una forma estándar de empaquetar y enviar mercancías. Se etiqueta, se carga, se descarga y se transporta de una ubicación a otra como si se tratase de una sola caja. El contenido del contenedor se aísla del contenido de otros contenedores para que no se afecten unos a otros. Estos principios subyacentes también se aplican a los contenedores de software.

Red Hat Enterprise Linux soporta contenedores mediante el uso de las siguientes tecnologías centrales (core):

- *Grupos de control* (*cgroups*) para administrar recursos.
- *Espacios de nombres* (Namespaces) para aislar procesos.
- SELinux y *Seccomp* (modo de computación segura) para aplicar (enforce) límites de seguridad.



nota

Para obtener un análisis más exhaustivo de la arquitectura y seguridad del contenedor, consulte el informe técnico "Diez capas de seguridad de contenedores" [<https://www.redhat.com/en/resources/container-security-openshift-cloud-devops-whitepaper>].

Diferencias entre contenedores y máquinas virtuales

Los contenedores ofrecen muchos de los mismos beneficios que las máquinas virtuales, como seguridad, almacenamiento y aislamiento de redes.

Ambas tecnologías aíslan las librerías de aplicaciones y los recursos de tiempo de ejecución del sistema operativo del host o del hipervisor y viceversa.

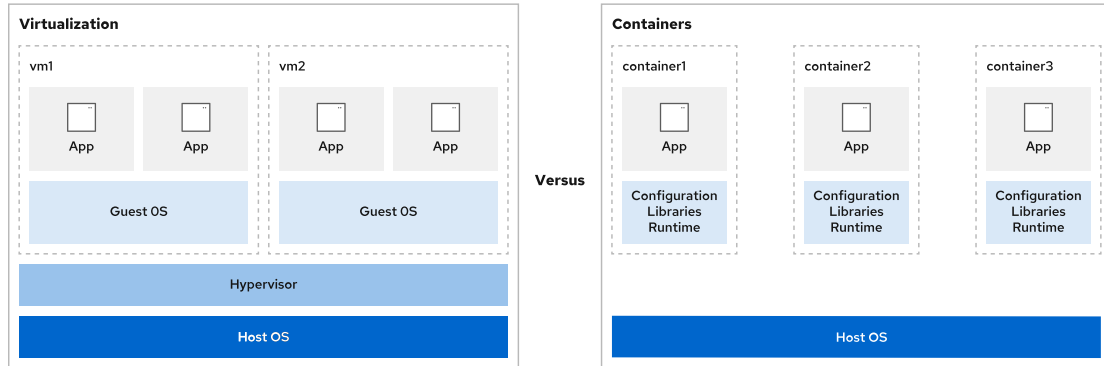


Figura 11.1: Comparación entre virtualización y contenerización

Los contenedores y las máquinas virtuales interactúan de manera diferente con el hardware y el sistema operativo subyacente.

Una máquina virtual tiene las siguientes características:

- Permite que varios sistemas operativos se ejecuten simultáneamente en una única plataforma de hardware.
- Usa un hipervisor para dividir el hardware en varios sistemas de hardware virtuales.
- Requiere un entorno de sistema operativo completo para dar soporte a la aplicación.

Un contenedor tiene las siguientes características:

- Se ejecuta directamente en el sistema operativo del host y comparte recursos con todos los contenedores del sistema.
- Comparte el kernel del host, pero aísla los procesos de la aplicación del resto del sistema.
- Requiere muchos menos recursos de hardware que las máquinas virtuales, por lo que los contenedores también se inician más rápido.
- Incluye todas las dependencias, como las dependencias del sistema y de programación, y los ajustes de configuración.



nota

Es posible que algunas aplicaciones no sean adecuadas para ejecutarse como un contenedor. Por ejemplo, en general, es posible que las aplicaciones que acceden a información de hardware de bajo nivel necesiten un acceso de hardware más directo que los contenedores.

Contenedores rootless y rootful

En el host de contenedores, puede ejecutar contenedores como usuario root o como usuario normal sin privilegios. Los contenedores que ejecuta un usuario con privilegios se denominan *contenedores rootful*. Los contenedores que ejecuta un usuario sin privilegios se denominan *contenedores rootless*.

Un contenedor rootless no puede usar recursos del sistema que generalmente están reservados para usuarios con privilegios, como el acceso a directorios restringidos o para publicar servicios de red en puertos restringidos (aquellos puertos por debajo de 1024). Esta función evita que un posible atacante obtenga privilegios de root en el host del contenedor.

Puede ejecutar contenedores directamente como `root`, si es necesario, pero este escenario debilita la seguridad del sistema si un error permite que un atacante comprometa el contenedor.

Diseño de arquitecturas basadas en contenedores

Los contenedores son una forma eficaz de volver a usar las aplicaciones alojadas y hacerlas portátiles. Los contenedores se pueden trasladar fácilmente de un entorno a otro, como de desarrollo a producción. Puede guardar varias versiones de un contenedor y acceder rápidamente a cada una según sea necesario.

Por lo general, los contenedores son temporales o *efímeros*. Puede guardar de forma permanente en un almacenamiento persistente los datos generados por un contenedor en ejecución, pero los contenedores, por lo general, se ejecutan cuando es necesario; luego, se detienen y se eliminan. Se inicia un nuevo proceso de contenedor la próxima vez que se necesita un contenedor en particular.

Podría instalar una aplicación de software compleja con varios servicios en un único contenedor. Por ejemplo, es posible que un servidor web necesite usar una base de datos y un sistema de mensajería. Sin embargo, el uso de un contenedor para varios servicios es difícil de administrar.

Un mejor diseño ejecuta en contenedores separados cada componente, el servidor web, la base de datos y el sistema de mensajería. De esta manera, las actualizaciones y el mantenimiento de los componentes de las aplicaciones individuales no afectan otros componentes ni la pila (stack) de aplicaciones.

Herramientas de administración de contenedores

Red Hat Enterprise Linux proporciona un conjunto de herramientas de contenedores que puede usar para ejecutar contenedores en un único servidor.

- `podman` administra contenedores e imágenes de contenedores.
- `skopeo` inspecciona, copia, elimina y firma imágenes.
- `buildah` crea imágenes de contenedores.

Estas herramientas son compatibles con Open Container Initiative (OCI). Con estas herramientas, usted puede administrar los contenedores de Linux creados por motores de contenedores compatibles con OCI, como Podman o Docker. Estas herramientas están diseñadas específicamente para ejecutar contenedores en Red Hat Enterprise Linux en un host de contenedores de un solo nodo.

En este capítulo, usará las utilidades `podman` y `skopeo` para ejecutar y administrar contenedores e imágenes de contenedores existentes.



nota

El uso de `buildah` para crear sus propias imágenes de contenedores no está incluido en el alcance de este curso, pero se trata en el curso de capacitación de Red Hat *Red Hat OpenShift I: Containers & Kubernetes* (DO180).

Imágenes de contenedores y registros

Para ejecutar contenedores, debe usar una *imagen de contenedor*. Una imagen de contenedor es un archivo estático que contiene pasos codificados y sirve como modelo para crear contenedores. Las imágenes de contenedor empaquetan una aplicación con todas sus dependencias, como las librerías del sistema, los tiempos de ejecución y las librerías del lenguaje de programación y otros valores de configuración.

Las imágenes de contenedores se compilan de acuerdo con las especificaciones, como la especificación de formato de imagen de Open Container Initiative (OCI). Estas especificaciones definen el formato para las imágenes de contenedores, así como los metadatos acerca de los sistemas operativos del host de contenedores y las arquitecturas de hardware que soporta la imagen.

Un *registro de contenedor* es un repositorio para almacenar y recuperar imágenes de contenedores. Un desarrollador *envía* o carga las imágenes de contenedores a un registro de contenedores. Puede *extraer* o descargar esas imágenes de contenedores del registro a un sistema local para ejecutar contenedores.

Puede usar un registro público que contenga imágenes de terceros o puede usar un registro privado controlado por su organización. El origen de las imágenes de contenedores es importante. Al igual que cualquier otro paquete de software, debe saber si puede confiar en el código de la imagen de contenedor. Las políticas varían entre registros sobre si proporcionan, evalúan y prueban las imágenes de contenedores que se les envían, y cómo lo hacen.

Red Hat distribuye las imágenes de contenedores certificadas a través de dos registros de contenedores principales a los que puede acceder con sus credenciales de inicio de sesión de Red Hat.

- `registry.redhat.io` para contenedores basados en productos oficiales de Red Hat.
- `registry.connect.redhat.com` para contenedores basados en productos de terceros.

Red Hat Container Catalog (<https://access.redhat.com/containers>) proporciona una interfaz web para buscar contenido certificado en estos registros.



nota

Red Hat proporciona la *imagen base universal (UBI)* como capa inicial para crear contenedores. La imagen de UBI es una imagen minimizada del contenedor que puede ser una primera capa para la compilación de una aplicación.

Necesita una cuenta de desarrollador de Red Hat para descargar una imagen de los registros de Red Hat. Puede usar el comando `podman login` para autenticarse en los registros. Si no proporciona una URL de registro al comando `podman login`, se autenticará en el registro configurado de forma predeterminada.

```
[user@host ~]$ podman login registry.lab.example.com
Username: RH134
Password: EXAMPLEPASSWORD
Login Succeeded!
```

También puede usar las opciones `--username` y `--password-stdin` del comando `podman login` para especificar el usuario y la contraseña para iniciar sesión en el registro. La opción `--password-stdin` lee la contraseña de `stdin`. Red Hat no recomienda usar la opción `--password`.

para proporcionar la contraseña directamente, ya que esta opción almacena la contraseña en los archivos de registro.

```
[user@host ~]# echo $PASSWORDVAR | podman login --username RH134 \
--password-stdin registry.access.redhat.com
```

Para verificar que haya iniciado sesión en un registro, use la opción `--get-login` del comando `podman login`.

```
[user01@rhel-vm ~]$ podman login registry.access.redhat.com --get-login
RH134
[user01@rhel-vm ~]$ podman login quay.io --get-login
Error: not logged into quay.io
```

En la salida anterior, la utilidad `podman` se autentica en el registro `registry.access.redhat.com` con las credenciales de usuario `RH134`, pero la utilidad `podman` no se autentica en el registro `quay.io`.

Configuración de registros de contenedores

El archivo de configuración predeterminado para los registros de contenedores es el archivo `/etc/containers/registries.conf`.

```
[user@host ~]$ cat /etc/containers/registries.conf
...output omitted...
[registries.search]
registries = ['registry.redhat.io', 'quay.io', 'docker.io']

# If you need to access insecure registries, add the registry's fully-qualified
# name.
# An insecure registry is one that does not have a valid SSL certificate or only
# does HTTP.
[registries.insecure]
registries = []
...output omitted...
```

Debido a que Red Hat recomienda usar un usuario sin privilegios para administrar contenedores, puede crear un archivo `registries.conf` para los registros de contenedores en el directorio `$HOME/.config/containers`. El archivo de configuración en este directorio anula la configuración en el archivo `/etc/containers/registries.conf`.

La lista de registros para buscar contenedores se configura en la sección `[registries.search]` de este archivo. Si especifica el nombre completo de una imagen de contenedor desde la línea de comandos, la utilidad de contenedor no busca en esta sección.

Los registros no seguros se enumeran en la sección `[registries.insecure]` del archivo `registries.conf`. Si un registro se detalla como no seguro, las conexiones a ese registro no están protegidas con cifrado TLS. Si un registro admite búsqueda y no es seguro, puede mostrarse en `[registries.search]` y en `[registries.insecure]`.

**nota**

En el aula se ejecuta un registro privado no seguro basado en Red Hat Quay para proporcionar imágenes de contenedores. Este registro satisface las necesidades del aula; sin embargo, no esperaría trabajar con registros inseguros en escenarios del mundo real. Para obtener más información sobre este software, consulte <https://access.redhat.com/products/red-hat-quay>.

Compilación de imágenes de contenedores con archivos de contenedores

Un *archivo contenedor* es un archivo de texto con instrucciones para compilar una imagen de contenedor. Un archivo contenedor generalmente tiene un *contexto* que define la ruta o URL donde se encuentran sus archivos y directorios. La imagen de contenedor resultante consta de capas de solo lectura, donde cada capa representa una instrucción del archivo contenedor.

La siguiente salida es un ejemplo de un archivo contenedor que usa la imagen de UBI del registro `registry.access.redhat.com`, instala el paquete `python3` e imprime la cadena `hello` en la consola.

```
[user@host ~]$ cat Containerfile
FROM registry.access.redhat.com/ubi8/ubi:latest
RUN dnf install -y python3
CMD ["/bin/bash", "-c", "echo hello"]
```

**nota**

La creación de un archivo contenedor y sus instrucciones de uso están fuera del alcance de este curso. Para obtener más información sobre los archivos de contenedor, consulte el curso DO180.

Administración de contenedores a escala

Las nuevas aplicaciones usan contenedores cada vez más para implementar componentes funcionales. Estos contenedores proporcionan servicios que consumen otras partes de la aplicación. En una organización, la administración de una cantidad cada vez mayor de contenedores puede convertirse rápidamente en una tarea abrumadora.

La implementación de contenedores a escala en la producción requiere un entorno que se puede adaptarse a los siguientes desafíos:

- La plataforma debe garantizar la disponibilidad de contenedores que proporcionen servicios esenciales.
- El entorno debe responder a los picos de uso de la aplicación mediante el aumento o la reducción de la cantidad de contenedores en ejecución y el balanceo de carga del tráfico.
- La plataforma debería detectar la falla de un contenedor o un host y reaccionar en consecuencia.
- Es posible que los desarrolladores necesiten un flujo de trabajo automatizado para ofrecer nuevas versiones de aplicaciones de forma transparente y segura.

Kubernetes es un servicio de orquestación que implementa, administra y escala aplicaciones basadas en contenedores en un clúster de hosts de contenedores. Kubernetes dirige el tráfico a sus contenedores con un balanceador de carga para que pueda escalar la cantidad de

contenedores que brindan un servicio. Kubernetes también soporta comprobaciones de estado definidas por el usuario para monitorear sus contenedores y reiniciarlos en caso de que fallen.

Red Hat proporciona una distribución de Kubernetes denominada *Red Hat OpenShift*. Red Hat OpenShift es un conjunto de componentes y servicios modulares desarrollado sobre la base de una infraestructura de Kubernetes. Ofrece características adicionales, como administración web remota, multiinquilino, monitoreo y auditoría, características avanzadas de seguridad, administración del ciclo de vida de la aplicación e instancias de autoservicio para desarrolladores.

Red Hat OpenShift no está incluido en el alcance de este curso, pero puede obtener más información sobre este en <https://www.openshift.com>.



nota

En una empresa, los contenedores individuales no se ejecutan generalmente desde la línea de comandos. En cambio, se prefiere ejecutar contenedores en producción mediante una plataforma basada en Kubernetes, como Red Hat OpenShift.

Sin embargo, es posible que necesite usar comandos para administrar contenedores e imágenes de forma manual o a escala pequeña. Este capítulo se centra en este caso de uso para comprender mejor los conceptos centrales (core) subyacentes de los contenedores, cómo funcionan y cómo pueden ser útiles.



Referencias

Páginas del manual `cgroups(7)`, `namespaces(7)`, `seccomp(2)`.

Especificación de imágenes de Open Container Initiative (OCI)

<https://github.com/opencontainers/image-spec/blob/master/spec.md>

Para obtener más información, consulte el capítulo *Starting with containers* de la *Red Hat Enterprise Linux 9 Building, Running, and Managing Containers Guide* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/building_running_and_managing_containers/index

► Cuestionario

Conceptos de contenedores

Elija las respuestas correctas para las siguientes preguntas:

- 1. **¿Qué herramienta de Red Hat Enterprise Linux ejecuta contenedores?**
 - a. buildah
 - b. container
 - c. podman
 - d. skopeo

- 2. **¿Cuáles dos enunciados describen la tecnología de contenedores? (Elija dos opciones).**
 - a. Los contenedores empaquetan sistemas operativos completos, con la adición de dependencias de librerías.
 - b. Los contenedores ejecutan procesos que están aislados del resto del sistema.
 - c. Cada contenedor incluye su propia versión del kernel y librerías.
 - d. Los contenedores proporcionan una forma estándar de empaquetar aplicaciones para facilitar la implementación y la administración.

- 3. **¿Cuáles dos enunciados son verdaderos sobre las imágenes de contenedores? (Elija dos opciones).**
 - a. Las imágenes de contenedores empaquetan una aplicación con todas las dependencias del tiempo de ejecución que necesita.
 - b. Las imágenes de contenedores que funcionan con Docker no pueden funcionar con Podman.
 - c. Las imágenes de contenedores solo se pueden ejecutar en un host de contenedores con la misma versión del software instalada en la imagen.
 - d. Las imágenes de contenedores sirven como modelos para crear contenedores.

- 4. **¿Cuáles son las tres tecnologías centrales (core) que se usan para implementar contenedores en Red Hat Enterprise Linux? (Elija tres opciones).**
 - a. Código de hipervisor para alojar máquinas virtuales.
 - b. Grupos de control (cgroups) para administrar recursos.
 - c. Espacios de nombres (Namespaces) para aislar procesos.
 - d. Sistema operativo completo para compatibilidad con el host de contenedores.
 - e. SELinux y Seccomp para la seguridad.

► 5. ¿Qué oración es verdadera acerca de los archivos contenedores?

- a. Un archivo contenedor es un archivo ejecutable que ejecuta un contenedor.
- b. Un archivo contenedor es un archivo ejecutable que compila una imagen de contenedor.
- c. Un archivo contenedor es un archivo comprimido que contiene librerías y configuración para un contenedor.
- d. Un archivo contenedor es un archivo de texto con instrucciones para compilar un contenedor.
- e. Un archivo contenedor es un archivo de texto con instrucciones para compilar una imagen de contenedor.

► Solución

Conceptos de contenedores

Elija las respuestas correctas para las siguientes preguntas:

- 1. **¿Qué herramienta de Red Hat Enterprise Linux ejecuta contenedores?**
 - a. buildah
 - b. container
 - c. podman
 - d. skopeo

- 2. **¿Cuáles dos enunciados describen la tecnología de contenedores? (Elija dos opciones).**
 - a. Los contenedores empaquetan sistemas operativos completos, con la adición de dependencias de librerías.
 - b. Los contenedores ejecutan procesos que están aislados del resto del sistema.
 - c. Cada contenedor incluye su propia versión del kernel y librerías.
 - d. Los contenedores proporcionan una forma estándar de empaquetar aplicaciones para facilitar la implementación y la administración.

- 3. **¿Cuáles dos enunciados son verdaderos sobre las imágenes de contenedores? (Elija dos opciones).**
 - a. Las imágenes de contenedores empaquetan una aplicación con todas las dependencias del tiempo de ejecución que necesita.
 - b. Las imágenes de contenedores que funcionan con Docker no pueden funcionar con Podman.
 - c. Las imágenes de contenedores solo se pueden ejecutar en un host de contenedores con la misma versión del software instalada en la imagen.
 - d. Las imágenes de contenedores sirven como modelos para crear contenedores.

- 4. **¿Cuáles son las tres tecnologías centrales (core) que se usan para implementar contenedores en Red Hat Enterprise Linux? (Elija tres opciones).**
 - a. Código de hipervisor para alojar máquinas virtuales.
 - b. Grupos de control (cgroups) para administrar recursos.
 - c. Espacios de nombres (Namespaces) para aislar procesos.
 - d. Sistema operativo completo para compatibilidad con el host de contenedores.
 - e. SELinux y Seccomp para la seguridad.

► 5. ¿Qué oración es verdadera acerca de los archivos contenedores?

- a. Un archivo contenedor es un archivo ejecutable que ejecuta un contenedor.
- b. Un archivo contenedor es un archivo ejecutable que compila una imagen de contenedor.
- c. Un archivo contenedor es un archivo comprimido que contiene librerías y configuración para un contenedor.
- d. Un archivo contenedor es un archivo de texto con instrucciones para compilar un contenedor.
- e. Un archivo contenedor es un archivo de texto con instrucciones para compilar una imagen de contenedor.

Implementación de contenedores

Objetivos

Analizar las herramientas de administración de contenedores para usar registros para almacenar y recuperar imágenes, y para implementar, consultar y acceder a contenedores.

La utilidad de Podman

Podman es un motor de contenedores con todas las funciones del metapaquete `container-tools` para administrar contenedores e imágenes de *Open Container Initiative (OCI)*. La utilidad `podman` no usa un daemon para funcionar, por lo que los desarrolladores no necesitan una cuenta de usuario privilegiada en el sistema para iniciar o detener contenedores. Podman proporciona varios subcomandos para interactuar con contenedores e imágenes. En la siguiente lista, se muestran los subcomandos que se usan en esta sección:

Comandos de Podman

Comando	Descripción
<code>podman-build</code>	Compilar una imagen de contenedor con un archivo de contenedor.
<code>podman-run</code>	Ejecutar un comando en un nuevo contenedor.
<code>podman-images</code>	Enumerar las imágenes en el almacenamiento local.
<code>podman-ps</code>	Imprimir información sobre los contenedores.
<code>podman-inspect</code>	Mostrar la configuración de un contenedor, una imagen, un volumen, una red o un pod.
<code>podman-pull</code>	Descargar una imagen de un registro.
<code>podman-cp</code>	Copiar archivos o carpetas entre un contenedor y el sistema de archivos local.
<code>podman-exec</code>	Ejecutar un comando en un contenedor en ejecución.
<code>podman-rm</code>	Eliminar uno o más contenedores.
<code>podman-rmi</code>	Eliminar una o más imágenes almacenadas localmente.
<code>podman-search</code>	Buscar una imagen en un registro.

Para tratar los temas de esta clase, imagine el siguiente escenario.

Como administrador del sistema, debe ejecutar un contenedor que se basa en la imagen de contenedor de RHEL 8 UBI denominada `python38` con el paquete `python-38`. También tiene la tarea de crear una imagen de contenedor a partir de un archivo de contenedor y ejecutar un contenedor denominado `python36` desde esa imagen de contenedor. La imagen de contenedor que se crea con el archivo de contenedor debe tener la etiqueta `python36:1.0`. Identifique

las diferencias entre los dos contenedores. Además, asegúrese de que los paquetes `python` instalados en los contenedores no entren en conflicto con la versión de Python instalada en su máquina local.

Instalación de utilidades de contenedores

El metapaquete `container-tools` contiene las utilidades necesarias para interactuar con los contenedores y las imágenes de contenedores. Para descargar, ejecutar y comparar contenedores en su sistema, instale el metapaquete `container-tools` con el comando `dnf install`. Use el comando `dnf info` para visualizar la versión y el contenido del paquete `container-tools`.

```
[root@host ~]# dnf install container-tools
...output omitted...
[user@host ~]$ dnf info container-tools
...output omitted...
Summary      : A meta-package witch container tools such as podman, buildah,
              : skopeo, etc.
License      : MIT
Description   : Latest versions of podman, buildah, skopeo, runc, common, CRIU,
              : Udica, etc as well as dependencies such as container-selinux
              : built and tested together, and updated.
...output omitted...
```

El metapaquete `container-tools` proporciona las utilidades `podman` y `skopeo` necesarias para realizar las tareas asignadas.

Descarga de una imagen de contenedor desde un registro.

Primero, asegúrese de que la utilidad `podman` esté configurada para buscar y descargar contenedores del registro `registry.redhat.io`. El comando `podman info` muestra la información de configuración de la utilidad `podman`, incluidos sus registros configurados.

```
[user@host ~]$ podman info
...output omitted...
insecure registries:
  registries: []
registries:
  registries:
    - registry.redhat.io
    - quay.io
    - docker.io
...output omitted...
```

El comando `podman search` busca una imagen coincidente en la lista de registros configurados. De manera predeterminada, Podman busca en todos los registros de búsqueda no calificada. Según la API de distribución de Docker que se implemente con el registro, es posible que algunos registros no soporten la función de búsqueda.

Use el comando `podman search` para mostrar una lista de imágenes en los registros configurados que contienen el paquete `python-38`.


```
[user@host ~]$ podman search python-38
```

NAME	DESCRIPTION
registry.access.redhat.com/ubi7/python-38	Python 3.8 platform for building and running applications
registry.access.redhat.com/ubi8/python-38	Platform for building and running Python 3.8 applications

...output omitted...

La imagen `registry.access.redhat.com/ubi8/python-38` parece coincidir con los criterios para el contenedor requerido.

Puede usar el comando `skopeo inspect` para examinar diferentes formatos de imagen de contenedor desde un directorio local o un registro remoto sin descargar la imagen. La salida de este comando muestra una lista de las etiquetas de versión disponibles, los puertos expuestos de la aplicación contenerizada y los metadatos de la imagen de contenedor. Use el comando `skopeo inspect` para verificar que la imagen contenga el paquete `python-38` requerido.

```
[user@host ~]$ skopeo inspect docker://registry.access.redhat.com/ubi8/python-38
{
  "Name": "registry.access.redhat.com/ubi8/python-38",
  "Digest": "sha256:c6e522cba2cf2b3ae4a875d5210fb94aa1e7ba71b6cebd902a4f4df73cb090b8",
  "RepoTags": [
    "1-68",
    "1-77-source",
    "latest"
  ],
  "name": "ubi8/python-38",
  "release": "86.1648121386",
  "summary": "Platform for building and running Python 3.8 applications",
  ...output omitted...
```

La imagen `registry.access.redhat.com/ubi8/python-38` contiene el paquete requerido y se basa en la imagen requerida. Use el comando `podman pull` para descargar la imagen seleccionada en la máquina local. Puede usar el nombre completo de la imagen de la salida anterior para evitar la ambigüedad en las versiones o registros del contenedor.

```
[user@host ~]$ podman pull registry.access.redhat.com/ubi8/python-38
Trying to pull registry.access.redhat.com/ubi8/python-38:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob c530010fb61c done
...output omitted...
```

A continuación, use el comando `podman images` para mostrar las imágenes locales.

```
[user@host ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
registry.access.redhat.com/ubi8/python-38	latest	a33d92f90990	1 hour ago	901 MB

Creación de una imagen de contenedor a partir de un archivo de contenedor

Se le proporciona el siguiente archivo de contenedor para crear la imagen de contenedor en el directorio `python36-app`:

```
[user@host python36-app]$ cat Containerfile
FROM registry.access.redhat.com/ubi8/ubi:latest
RUN dnf install -y python36
CMD ["/bin/bash", "-c", "sleep infinity"]
```

El archivo contenedor anterior usa la imagen `registry.access.redhat.com/ubi8/ubi:latest` como imagen base. A continuación, el archivo contenedor instala el paquete `python36` y ejecuta el comando `sleep infinity` para evitar que el contenedor salga.

Normalmente, un contenedor ejecuta un proceso y, luego, sale después de que se completa ese proceso. El comando `sleep infinity` evita que el contenedor salga, ya que el proceso nunca se completa. A continuación, puede probar, desarrollar y depurar dentro del contenedor.

Después de examinar el archivo contenedor, use el comando `podman build` para compilar la imagen. El comando `podman build` crea una imagen de contenedor mediante el uso de instrucciones de uno o más archivos de contenedor. Debe estar en el directorio con el archivo contenedor para compilar la imagen con el comando `podman build`. Puede usar la opción `-t` del comando `podman build` para proporcionar el nombre y la etiqueta `python36:1.0` para la nueva imagen.

```
[user@host python36-app]$ podman build -t python36:1.0 .
STEP 1/3: FROM registry.access.redhat.com/ubi8/ubi:latest
STEP 2/3: RUN dnf install -y python36
...output omitted...
STEP 3/3: CMD ["/bin/bash", "-c", "sleep infinity"]
COMMIT python36:1.0
--> 35ab820880f
Successfully tagged localhost/python36:1.0
35ab820880f1708fa310f835407ffc94cb4b4fe2506b882c162a421827b156fc
```

La última línea de la salida anterior muestra el ID de la imagen del contenedor. La mayoría de los comandos de Podman usan los primeros 12 caracteres del ID de imagen de contenedor para hacer referencia a la imagen de contenedor. Puede usar este ID corto o el nombre de un contenedor o una imagen de contenedor como argumentos para la mayoría de los comandos de Podman.



nota

Si no se especifica un número de versión en la etiqueta, la imagen se crea con la etiqueta `:latest`. Si no se especifica un nombre de imagen, los campos de imagen y etiqueta muestran la cadena `<none>`.

Use el comando `podman images` para verificar que la imagen se crea con el nombre y la etiqueta definidos.

```
[user@host ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
localhost/python36	1.0	35ab820880f1	3 minute ago	266 MB
registry.access.redhat.com/ubi8/python-38	latest	a33d92f90990	1 hour ago	901 MB

A continuación, use el comando `podman inspect` para ver la información de bajo nivel de la imagen de contenedor y verificar que su contenido coincida con los requisitos del contenedor.

```
[user@host ~]$ podman inspect localhost/python36:1.0
...output omitted...
    "Cmd": [
        "/bin/bash",
        "-c",
        "sleep infinity"
    ],
...output omitted...
    {
        "created": "2022-04-18T19:47:52.708227513Z",
        "created_by": "/bin/sh -c dnf install -y python36",
        "comment": "FROM registry.access.redhat.com/ubi8/ubi:latest"
    },
...output omitted...
```

La salida del comando `podman inspect` muestra la imagen base `registry.access.redhat.com/ubi8/ubi:latest`, el comando `dnf` para instalar el paquete `python36` y el comando `bash sleep infinity` que se ejecuta en el tiempo de ejecución para evitar que el contenedor salga.



nota

La salida del comando `podman inspect` varía de la imagen `python-38` a la imagen `python36`, ya que creó la imagen `/python36` al agregar una capa con cambios a la imagen base `registry.access.redhat.com/ubi8/ubi:latest` existente, mientras que la imagen `python-38` es en sí misma una imagen base.

Ejecución de contenedores

Ahora que tiene las imágenes de contenedor requeridas, puede usarlas para ejecutar contenedores. Un contenedor puede estar en uno de los siguientes estados:

Creado

Un contenedor que se crea pero no se inicia.

En ejecución

Un contenedor que se está ejecutando con sus procesos.

Detenido

Un contenedor con sus procesos detenidos.

En pausa

Un contenedor con sus procesos en pausa. Sin soporte para contenedores rootless.

Eliminado

Un contenedor con sus procesos en estado inactivo.

El comando `podman ps` enumera los contenedores en ejecución en el sistema. Use el comando `podman ps -a` para ver todos los contenedores (creados, detenidos, en pausa o en ejecución) en la máquina.

Use el comando `podman create` para crear el contenedor para ejecutarse más tarde. Para crear el contenedor, use el ID de la imagen de contenedor `localhost/python36`. También puede usar la opción `--name` para establecer un nombre para identificar el contenedor. La salida del comando es el ID largo del contenedor.

```
[user@host ~]$ podman create --name python36 dd6ca291f097
c54c7ee281581c198cb96b07d78a0f94be083ae94dacbae69c05bd8cd354bbec
```

**nota**

Si no define un nombre para el contenedor con el comando `podman create` o `podman run` con la opción `--name`, la utilidad `podman` asigna un nombre aleatorio al contenedor.

A continuación, use los comandos `podman ps` y `podman ps -a` para verificar que el contenedor se haya creado pero no se haya iniciado. Puede ver información sobre el contenedor `python36`, como el ID corto, el nombre y el estado del contenedor, el comando que ejecuta cuando se inicia y la imagen para crear el contenedor.

```
[user@host ~]$ podman create --name python36 dd6ca291f097
c54c7ee281581c198cb96b07d78a0f94be083ae94dacbae69c05bd8cd354bbec
[user@host ~]$ podman ps
CONTAINER ID  IMAGE          COMMAND          CREATED          STATUS          PORTS          NAMES
[user@host ~]$ podman ps -a
CONTAINER ID  IMAGE          COMMAND          CREATED          STATUS          PORTS          NAMES
c54c7ee28158  localhost/python36:1.0  /bin/bash -c slee...  5 seconds ago  Created
python36
```

Ahora que verificó que el contenedor se creó correctamente, decide iniciar el contenedor, por lo que ejecuta el comando `podman start`. Puede usar el nombre o el ID del contenedor para iniciar el contenedor. La salida de este comando es el nombre del contenedor.

```
[user@host ~]$ podman start python36
python36
[user@host ~]$ podman ps
CONTAINER ID  IMAGE          COMMAND          CREATED          STATUS          PORTS          NAMES
c54c7ee28158  localhost/python36:1.0  /bin/bash -c slee...  6 minutes ago  Up 3
seconds ago          python36
```

Ejecución de un contenedor desde un repositorio remoto

Puede usar el comando `podman run` para crear y ejecutar el contenedor más tarde. El comando `podman run` ejecuta un proceso dentro de un contenedor y este proceso inicia el nuevo contenedor.

Use la opción `-d` del comando `podman run` para ejecutar un contenedor en *modo separado*, que ejecuta el contenedor en segundo plano en lugar de primer plano de la sesión. En el ejemplo del contenedor `python36`, no necesita proporcionar un comando para que se ejecute el contenedor, ya que el comando `sleep infinity` ya se proporcionó en el archivo de contenedor que creó la imagen para ese contenedor.

Para crear el contenedor `python38`, usted opta por usar el comando `podman run` y hacer referencia a la imagen `registry.access.redhat.com/ubi8/python-38`. También opta por usar el comando `sleep infinity` para evitar que el contenedor salga.

```
[user@host ~]$ podman run -d --name python38 \
registry.access.redhat.com/ubi8/python-38 \
sleep infinity
a60f71a1dc1b997f5ef244aaed232e5de71dd1e8a2565428ccfebde73a2f9462
[user@host ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
c54c7ee28158	localhost/python36:1.0	/bin/bash -c
slee...	37 minutes ago	Up 30 minutes ago
a60f71a1dc1b	registry.access.redhat.com/ubi8/python-38:latest	sleep infinity
32 seconds ago	Up 33 seconds ago	python38



Importante

Si ejecuta un contenedor con el nombre completo de la imagen, pero la imagen aún no está almacenada localmente, el comando `podman run` primero extrae la imagen del registro y, luego, se ejecuta.

Aislamiento del entorno en contenedores

Los contenedores aíslan el entorno de una aplicación. Cada contenedor tiene su propio sistema de archivos, redes y procesos. Puede observar la función de aislamiento cuando observa la salida del comando `ps` y la compara entre la máquina host y un contenedor en ejecución.

Primero ejecuta el comando `ps -ax` en la máquina local y el comando devuelve un resultado esperado con muchos procesos.

```
[root@host ~]# ps -ax
```

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:01	/usr/lib/systemd/systemd --switched-root --system --deseriali
2	?	S	0:00	[kthreadd]
3	?	I<	0:00	[rcu_gp]
4	?	I<	0:00	[rcu_par_gp]

...output omitted...

El comando `podman exec` ejecuta un comando en un contenedor en ejecución. El comando toma el nombre o ID del contenedor como primer argumento y los siguientes argumentos como comandos para ejecutarse dentro del contenedor. Use el comando `podman exec` para ver los procesos en ejecución en el contenedor `python36`. La salida del comando `ps aux` se ve diferente, ya que ejecuta procesos diferentes desde la máquina local.

```
[student@host ~]$ podman exec python38 ps -ax
PID TTY      STAT   TIME COMMAND
  1 ?        Ss     0:00 /usr/bin/coreutils --coreutils-prog-shebang=sleep /
usr/bin/sleep infinity
  7 ?        R       0:00 ps -ax
```

Puede usar el comando `sh -c` para encapsular el comando que se ejecutará en el contenedor. En el siguiente ejemplo, el comando `ps -ax > /tmp/process-data.log` se interpreta como el comando que se ejecutará en el contenedor. Si no encapsula el comando, Podman podría interpretar el carácter mayor que (`>`) como parte del comando `podman` en lugar de como un argumento para la opción `podman exec`.

```
[student@host ~]$ podman exec python38 sh -c 'ps -ax > /tmp/process-data.log'
PID TTY      STAT   TIME COMMAND
  1 ?        Ss     0:00 /usr/bin/coreutils --coreutils-prog-shebang=sleep /
usr/bin/sleep infinity
  7 ?        R       0:00 ps -ax
```

Usted decide comparar la versión instalada `python` en el sistema host con la versión instalada `python` en los contenedores.

```
[user@host ~]$ python3 --version
Python 3.9.10
[user@host ~]$ podman exec python36 python3 --version
Python 3.6.8
[user@host ~]$ podman exec python38 python3 --version
Python 3.8.8
```

Aislamiento de sistemas de archivos en contenedores

Los desarrolladores pueden usar la función de aislamiento del sistema de archivos para escribir y probar aplicaciones para diferentes versiones de lenguajes de programación sin la necesidad de usar varias máquinas físicas o virtuales.

Cree un script bash simple que muestre `hello world` en el terminal en el directorio `/tmp`.

```
[user@host ~]$ echo "echo 'hello world'" > /tmp/hello.sh
```

El archivo `/tmp/hello.sh` existe en la máquina host y no existe en el sistema de archivos dentro de los contenedores. Si intenta usar `podman exec` para ejecutar el script, se produce un error, ya que el script `/tmp/hello.sh` no existe en el contenedor.

```
[user@host ~]$ stat /tmp/hello.sh
File: /tmp/hello.sh
Size: 19          Blocks: 8          IO Block: 4096   regular file
Device: fc04h/64516d Inode: 17655599   Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/ user)   Gid: ( 1000/ user)
Context: unconfined_u:object_r:user_tmp_t:s0
Access: 2022-04-19 21:47:40.101601412 -0400
Modify: 2022-04-19 21:47:36.497558132 -0400
Change: 2022-04-19 21:47:36.497558132 -0400
Birth: 2022-04-19 21:45:24.785976758 -0400
```

```
[user@host ~]$ podman exec python38 stat /tmp/hello.sh
stat: cannot statx '/tmp/hello.sh': No such file or directory
```

El comando `podman cp` copia archivos y carpetas entre los sistemas de archivos del host y del contenedor. Puede copiar el archivo `/tmp/hello.sh` en el contenedor `python38` con el comando `podman cp`.

```
[user@host ~]$ podman cp /tmp/hello.sh python38:/tmp/hello.sh

[user@host ~]$ podman exec python38 stat /tmp/hello.sh
  File: /tmp/hello.sh
  Size: 19          Blocks: 8          IO Block: 4096   regular file
Device: 3bh/59d Inode: 12280058      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1001/ default)   Gid: (   0/  root)
Access: 2022-04-20 01:47:36.000000000 +0000
Modify: 2022-04-20 01:47:36.000000000 +0000
Change: 2022-04-20 02:02:04.732982187 +0000
 Birth: 2022-04-20 02:02:04.732982187 +0000
```

Una vez que se copia el script en el sistema de archivos del contenedor, se puede ejecutar desde el contenedor.

```
[user@host ~]$ podman exec python38 bash /tmp/hello.sh
hello world
```

Eliminación de contenedores e imágenes

Puede eliminar contenedores e imágenes con los comandos `podman rm` y `podman rmi`, respectivamente. Antes de eliminar una imagen de contenedor, se deben eliminar todos los contenedores en ejecución existentes de esa imagen.

Decide eliminar el contenedor `python38` y su imagen relacionada. Si intenta eliminar la imagen `registry.access.redhat.com/ubi8/python-38` mientras existe el contenedor `python38`, se produce un error.

```
[user@host ~]$ podman rmi registry.access.redhat.com/ubi8/python-38
Error: Image used by
a60f71a1dc1b997f5ef244aaed232e5de71dd1e8a2565428ccfebde73a2f9462: image is in use
by a container
```

Debe detener el contenedor para poder eliminarlo. Para detener un contenedor, use el comando `podman stop`.

```
[user@host ~]$ podman stop python38
```

Después de detener el contenedor, use el comando `podman rm` para eliminar el contenedor.

```
[user@host ~]$ podman rm python38
a60f71a1dc1b997f5ef244aaed232e5de71dd1e8a2565428ccfebde73a2f9462
```

Cuando el contenedor ya no existe, el `registry.access.redhat.com/ubi8/python-38` se puede eliminar con el comando `podman rmi`.

```
[user@host ~]$ podman rmi registry.access.redhat.com/ubi8/python-38
Untagged: registry.access.redhat.com/ubi8/python-38:latest
Deleted: a33d92f90990c9b1bad9aa98fe017e48f30c711b49527dcc797135352ea57d12
```



Referencias

Páginas del manual: `podman(1)`, `podman-build(1)`, `podman-cp(1)`, `podman-exec(1)`, `podman-images(1)`, `podman-inspect(1)`, `podman-ps(1)`, `podman-pull(1)`, `podman-rm(1)`, `podman-rmi(1)`, `podman-run(1)`, `podman-search(1)` y `podman-stop(1)`

Para obtener más información, consulte el capítulo Comenzar con contenedores en la Guía de creación, ejecución y administración de contenedores de Red Hat Enterprise Linux 9 en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/building_running_and_managing_containers/index#starting-with-containers_building-running-and-managing-containers

► Ejercicio Guiado

Implementación de contenedores

En este ejercicio, usará herramientas de administración de contenedores para compilar una imagen, ejecutar un contenedor y consultar el entorno de contenedores en ejecución.

Resultados

- Configurar un registro de imágenes de contenedor y crear un contenedor a partir de una imagen existente.
- Creación de un contenedor a partir de un archivo de contenedor.
- Copiar un script de una máquina host en contenedores y ejecutar el script.
- Elimine los contenedores y las imágenes.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start containers-deploy
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Instale el metapaquete `container-tools`.

```
[student@servera ~]$ sudo dnf install container-tools
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 3. Configure el registro del aula `registry.lab.example.com` en su directorio de inicio. Inicie sesión en el registro de contenedor con el usuario `admin` y `redhat321` como contraseña.

- 3.1. Cree el directorio `/home/student/.config/containers`.

```
[student@servera ~]$ mkdir -p /home/student/.config/containers
```

- 3.2. Cree el archivo `/home/student/.config/containers/registries.conf` con el siguiente contenido:

```
unqualified-search-registries = ['registry.lab.example.com']

[[registry]]
location = "registry.lab.example.com"
insecure = true
blocked = false
```

- 3.3. Verifique que se haya agregado el registro del aula.

```
[student@servera ~]$ podman info
...output omitted...
registries:
  registry.lab.example.com:
    Blocked: false
    Insecure: true
    Location: registry.lab.example.com
    MirrorByDigestOnly: false
    Mirrors: null
    Prefix: registry.lab.example.com
  search:
    - registry.lab.example.com
...output omitted...
```

- 3.4. Inicie sesión en el registro del aula.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 4. Ejecute el contenedor `python38` en modo separado desde una imagen con el paquete `python 3.8` y basado en la imagen `ubi8`. La imagen se aloja en un registro remoto.

- 4.1. Busque un contenedor `python-38` en el registro `registry.lab.example.com`.

```
[student@servera ~]$ podman search registry.lab.example.com/
NAME                                DESCRIPTION
...output omitted...
registry.lab.example.com/ubi8/python-38
registry.lab.example.com/ubi8/httpd-24
registry.lab.example.com/rhel8/php-74
```

- 4.2. Inspeccione la imagen.

```
[student@servera ~]$ skopeo inspect \
docker://registry.lab.example.com/ubi8/python-38
...output omitted...
    "description": "Python 3.8 available as container is a base platform for
building and running various Python 3.8 applications and frameworks.
...output omitted...
```

4.3. Extraiga la imagen de contenedor python-38.

```
[student@servera ~]$ podman pull registry.lab.example.com/ubi8/python-38
Trying to pull registry.lab.example.com/ubi8/python-38:latest...
...output omitted...
671cc3cb42984e338733ebb5a9a68e69e267cb7f9cb802283d3bc066f6321617
```

4.4. Verifique que el contenedor se descargue en el repositorio de imágenes local.

```
[student@servera ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED
SIZE			
registry.lab.example.com/ubi8/python-38	latest	671cc3cb4298	5 days ago
901 MB			

4.5. Inicie el contenedor python38.

```
[student@servera ~]$ podman run -d --name python38 \
registry.lab.example.com/ubi8/python-38 sleep infinity
004756b52d3d3326545f5075594cffa858afd474b903288723a3aa299e72b1af
```

4.6. Verifique que el contenedor se haya creado.

```
[student@servera ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
CREATED	STATUS	PORTS
004756b52d3d	registry.lab.example.com/ubi8/python-38:latest	sleep infinity
About a minute ago	Up About a minute ago	python38

- 5. Compile una imagen de contenedor denominada `python36:1.0` a partir de un archivo de contenedor y use la imagen para crear un contenedor denominado `python36`.

5.1. Cambie al directorio `/home/student/python36` y examine el archivo contenedor llamado `Containerfile`.

```
[student@servera ~]$ cd /home/student/python36
[student@servera python36]$ cat Containerfile
FROM registry.lab.example.com/ubi8/ubi:latest
RUN dnf install -y python36
CMD ["/bin/bash", "-c", "sleep infinity"]
```

5.2. Cree una imagen de contenedor a partir de un archivo de contenedor.

```
[student@servera python36]$ podman build -t python36:1.0 .
STEP 1/3: FROM registry.lab.example.com/ubi8/ubi:latest
...output omitted...
STEP 2/3: RUN dnf install -y python36
...output omitted...
STEP 3/3: CMD ["/bin/bash", "-c", "sleep infinity"]
...output omitted...
Successfully tagged localhost/python36:1.0
80e68c195925beafe3b2ad7a54fe1e5673993db847276bc62d5f9d109e9eb499
```

5.3. Verifique que exista la imagen del contenedor en el repositorio de imágenes local.

```
[student@servera ~]$ podman images
```

REPOSITORY	TAG	IMAGE ID	CREATED
SIZE			
localhost/python36	1.0	80e68c195925	3 minutes ago
266 MB			
registry.lab.example.com/ubi8/python-38	latest	671cc3cb4298	5 days ago
901 MB			
registry.lab.example.com/ubi8/ubi	latest	fca12da1dc30	4 months ago
235 MB			

5.4. Inspeccione el contenedor python36.

```
[student@servera ~]$ podman inspect localhost/python36:1.0
...output omitted...
    "created": "2022-04-26T20:58:13.448031264Z",
    "created_by": "/bin/sh -c dnf install -y python36",
    "comment": "FROM registry.lab.example.com/ubi8/ubi:latest"
  },
  {
    "created": "2022-04-26T20:58:14.057396235Z",
    "created_by": "/bin/sh -c #(nop) CMD [\"/bin/bash\", \"-c\",
    \"sleep infinity\"]",
    ...output omitted...
```

5.5. Cree el contenedor python36.

```
[student@servera ~]$ podman create --name python36 localhost/python36:1.0
3db4eabe9043224a7bdf195ab5fd810bf95db98dc29193392cef7b94489e1aae
```

5.6. Inicie el contenedor python36.

```
[student@servera ~]$ podman start python36
python36
```

5.7. Verifique que el contenedor esté ejecutándose.

```
[student@servera ~]$ podman ps
```

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND
004756b52d3d	registry.lab.example.com/ubi8/python-38:latest	Up 33 minutes ago		python38	sleep infinity
3db4eabe9043	localhost/python36:1.0	Up 42 seconds ago		python36	/bin/bash -c slee...

- 6. Copie el script `/home/student/script.py` en el directorio `/tmp` de los contenedores en ejecución y ejecute el script en cada contenedor.

- 6.1. Copie el script de python `/home/student/script.py` en el directorio `/tmp` en ambos contenedores.

```
[student@servera ~]$ podman cp /home/student/script.py python36:/tmp/script.py
[student@servera ~]$ podman cp /home/student/script.py python38:/tmp/script.py
```

- 6.2. Ejecute el script de Python en ambos contenedores y, luego, ejecútelo en el host.

```
[student@servera ~]$ podman exec -it python36 python3 /tmp/script.py
This script was not run on the correct version of Python
Expected version of Python is 3.8
Current version of python is 3.6
[student@servera ~]$ podman exec -it python38 python3 /tmp/script.py
This script was correctly run on Python 3.8
[student@servera ~]$ python3 /home/student/script.py
This script was not run on the correct version of Python
Expected version of Python is 3.8
Current version of python is 3.9
```

- 7. Elimine los contenedores y las imágenes.

- 7.1. Detenga ambos contenedores.

```
[student@servera ~]$ podman stop python36 python38
...output omitted...
python38
python36
```

- 7.2. Elimine ambos contenedores.

```
[student@servera ~]$ podman rm python36 python38
3db4eabe9043224a7bdf195ab5fd810bf95db98dc29193392cef7b94489e1aae
004756b52d3d3326545f5075594cffa858afd474b903288723a3aa299e72b1af
```

- 7.3. Elimine ambas imágenes de contenedores.

```
[student@servera ~]$ podman rmi localhost/python36:1.0 \
registry.lab.example.com/ubi8/python-38:latest \
registry.lab.example.com/ubi8/ubi
Untagged: localhost/python36:1.0
Untagged: registry.lab.example.com/ubi8/python-38:latest
Deleted: 80e68c195925beafe3b2ad7a54fe1e5673993db847276bc62d5f9d109e9eb499
Deleted: 219e43f6ff96fd11ea64f67cd6411c354dacbc5cbe296ff1fdbf5b717f01d89a
Deleted: 671cc3cb42984e338733ebb5a9a68e69e267cb7f9cb802283d3bc066f6321617
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish containers-deploy
```

Esto concluye la sección.

Administración del almacenamiento del contenedor y los recursos de red

Objetivos

Proporcionar almacenamiento persistente para los datos del contenedor al compartir el almacenamiento desde el host del contenedor y configurar una red de contenedores.

Administración de recursos del contenedor

Puede usar contenedores para ejecutar un proceso simple y salir. También puede configurar un contenedor para ejecutar un servicio de forma continua, como un servidor de base de datos. En este escenario, puede considerar agregar más recursos al contenedor, como el almacenamiento persistente o la resolución de DNS para otros contenedores.

Puede usar diferentes estrategias para configurar el almacenamiento persistente para contenedores. En una plataforma de contenedores empresarial, como Red Hat OpenShift, puede usar soluciones de almacenamiento sofisticadas para proporcionar almacenamiento a sus contenedores sin conocer la infraestructura subyacente.

Para implementaciones pequeñas en las que usa solo un host de contenedor, y sin la necesidad de escalar, puede crear almacenamiento persistente desde el host de contenedor mediante la creación de un directorio para montar en el contenedor en ejecución.

Cuando un contenedor, como un servidor web o un servidor de base de datos, proporciona contenido para clientes fuera del host del contenedor, debe configurar un canal de comunicación para que esos clientes accedan al contenido del contenedor. Puede configurar la *asignación de puertos* para habilitar la comunicación con un contenedor. Con la asignación de puertos, las solicitudes destinadas a un puerto en el host del contenedor se reenvían a un puerto dentro del contenedor.

Para tratar los temas de esta clase, imagine el siguiente escenario.

Como administrador del sistema, tiene la tarea de crear la base de datos contenerizada **db01**, basada en MariaDB, para usar la máquina local para permitir el tráfico del puerto 3306 con la configuración de firewall adecuada. El contenedor **db01** debe usar almacenamiento persistente con el contexto de SELinux adecuado. Agregue la configuración de red adecuada para que el contenedor **client01** pueda comunicarse con el contenedor **db01** con DNS.

Variables del entorno para contenedores

Algunas imágenes de contenedor permiten el paso de variables de entorno para personalizar el contenedor en el momento de la creación. Puede usar variables de entorno para establecer parámetros en el contenedor para adaptarlo a su entorno sin la necesidad de crear su propia imagen personalizada. Por lo general, no modificaría la imagen del contenedor, ya que agregaría capas a la imagen, lo que podría ser más difícil de mantener.

Usa el comando `podman run -d registry.lab.example.com/rhel8/mariadb-105` para ejecutar una base de datos contenerizada, pero observa que el contenedor no se inicia.

```
[user@host ~]$ podman run -d registry.lab.example.com/rhel8/mariadb-105 \
--name db01
20751a03897f14764fb0e7c58c74564258595026124179de4456d26c49c435ad
[user@host ~]$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND
20751a03897f	registry.lab.example.com/rhel8/mariadb-105:latest	run-mysqld
29 seconds ago	Exited (1) 29 seconds ago	db01

Usa el comando `podman container logs` para investigar el motivo del estado del contenedor.

```
[user@host ~]$ podman container logs db01
...output omitted...
You must either specify the following environment variables:
  MYSQL_USER (regex: '^[a-zA-Z0-9_]+$')
  MYSQL_PASSWORD (regex: '^[a-zA-Z0-9_~!@#%&*()-=<>,.?;:|]+$')
  MYSQL_DATABASE (regex: '^[a-zA-Z0-9_]+$')
Or the following environment variable:
  MYSQL_ROOT_PASSWORD (regex: '^[a-zA-Z0-9_~!@#%&*()-=<>,.?;:|]+$')
Or both.
...output omitted...
```

A partir de la salida anterior, determina que el contenedor no continuó ejecutándose porque las variables de entorno requeridas no se pasaron al contenedor. Por lo tanto, inspeccione la imagen de contenedor `mariadb-105` para encontrar más información acerca de las variables de entorno para personalizar el contenedor.

```
[user@host ~]$ skopeo inspect docker://registry.lab.example.com/rhel8/mariadb-105
...output omitted...
  "name": "rhel8/mariadb-105",
  "release": "40.1647451927",
  "summary": "MariaDB 10.5 SQL database server",
  "url": "https://access.redhat.com/containers/#/registry.access.redhat.com/
rhel8/mariadb-105/
images/1-40.1647451927",
  "usage": "podman run -d -e MYSQL_USER=user -e MYSQL_PASSWORD=pass -e
MYSQL_DATABASE=db -p 3306:3306 rhel8/mariadb-105",
  "vcs-ref": "c04193b96a119e176ada62d779bd44a0e0edf7a6",
  "vcs-type": "git",
  "vendor": "Red Hat, Inc.",
...output omitted...
```

La etiqueta `usage` de la salida proporciona un ejemplo de cómo ejecutar la imagen. La etiqueta `url` apunta a una página web en Red Hat Container Catalog que documenta las variables de entorno y otra información acerca de cómo usar la imagen de contenedor.

La documentación para esta imagen muestra que el contenedor usa el puerto 3306 para el servicio de base de datos. La documentación también muestra que las siguientes variables de entorno están disponibles para configurar el servicio de base de datos:

Variables del entorno para la imagen mariadb

Variable	Descripción
MYSQL_USER	Nombre de usuario para la cuenta MySQL que se creará
MYSQL_PASSWORD	Contraseña para la cuenta de usuario
MYSQL_DATABASE	Nombre de la base de datos
MYSQL_ROOT_PASSWORD	Contraseña para el usuario root (opcional)

Después de examinar las variables de entorno disponibles para la imagen, use la opción `-e` del comando `podman run` para pasar variables de entorno al contenedor y use el comando `podman ps` para verificar que se esté ejecutando.

```
[user@host ~]$ podman run -d --name db01 \
-e MYSQL_USER=student \
-e MYSQL_PASSWORD=student \
-e MYSQL_DATABASE=dev_data \
-e MYSQL_ROOT_PASSWORD=redhat \
registry.lab.example.com/rhel8/mariadb-105
[user@host ~]$ podman ps
```

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND
4b8f01be7fd6	registry.lab.example.com/rhel8/mariadb-105:latest	Up 6 seconds ago		db01	run-mysqld 6

Almacenamiento persistente de contenedores

De manera predeterminada, el almacenamiento que usa un contenedor es efímero. La naturaleza efímera del almacenamiento en el contenedor significa que su contenido se pierde después de eliminar el contenedor. Debe tener en cuenta los permisos de nivel de sistema de archivos cuando monta un volumen persistente en un contenedor.

En la imagen MariaDB, el usuario `mysql` debe ser propietario del directorio `/var/lib/mysql`, al igual que si MariaDB se estuviera ejecutando en la máquina `host`. El directorio que desea montar en el contenedor debe tener `mysql` como el usuario y el propietario del grupo (o el UID/GID del usuario `mysql`, si MariaDB no está instalado en la máquina `host`). Si ejecuta un contenedor con el usuario `root`, los UID y GID en su máquina `host` coinciden con los UID y GID dentro del contenedor.

La configuración de coincidencia de UID y GID no se produce de la misma manera en un contenedor `rootless`. En un contenedor `rootless`, el usuario tiene acceso `root` desde dentro del contenedor, ya que Podman inicia un contenedor dentro del espacio de nombres del usuario.

Puede usar el comando `podman unshare` para ejecutar un comando dentro del espacio de nombres del usuario. Para obtener la asignación de UID para su espacio de nombres de usuario, use el comando `podman unshare cat`.

```
[user@host ~]$ podman unshare cat /proc/self/uid_map
      0           1000           1
      1       100000       65536
[user@host ~]$ podman unshare cat /proc/self/gid_map
      0           1000           1
      1       100000       65536
```

La salida anterior muestra que en el contenedor, el usuario root (UID y GID de 0) se asigna a su usuario (UID y GID de 1000) en la máquina host. En el contenedor, el UID y el GID de 1 se asignan al UID y al GID de 100000 en la máquina host. Cada UID y GID después de 1 se incrementa en 1. Por ejemplo, el UID y GID de 30 dentro de un contenedor se asigna al UID y GID de 100029 en la máquina host.

Use el comando `podman exec` para ver el UID y el GID del usuario `mysql` dentro del contenedor que se está ejecutando con almacenamiento efímero.

```
[user@host ~]$ podman exec -it db01 grep mysql /etc/passwd
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
```

Decide montar el directorio `/home/user/db_data` en el contenedor `db01` para proporcionar almacenamiento persistente en el directorio `/var/lib/mysql` del contenedor. A continuación, cree el directorio `/home/user/db_data` y use el comando `podman unshare` para establecer el UID y el GID del espacio de nombres de usuario de 27 como propietario del directorio.

```
[user@host ~]$ mkdir /home/user/db_data
[user@host ~]$ podman unshare chown 27:27 /home/user/db_data
```

El UID y el GID de 27 en el contenedor se asignan al UID y al GID de 100026 en la máquina host. Puede verificar la asignación al ver la propiedad del directorio `/home/user/db_data` con el comando `ls`.

```
[student@workstation ~]$ ls -l /home/user/
total 0
drwxrwxr-x. 3 100026 100026 18 May  5 14:37 db_data
...output omitted...
```

Ahora que se han establecido los permisos de nivel de sistema de archivos correctos, use la opción `-v` del comando `podman run` para montar el directorio.

```
[user@host ~]$ podman run -d --name db01 \
-e MYSQL_USER=student \
-e MYSQL_PASSWORD=student \
-e MYSQL_DATABASE=dev_data \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/user/db_data:/var/lib/mysql \
registry.lab.example.com/rhel8/mariadb-105
```

Observa que el contenedor `db01` no se está ejecutando.

```
[user@host ~]$ podman ps -a
```

CONTAINER ID	IMAGE	PORTS	NAMES	COMMAND
dfdc20cf9a7e	registry.lab.example.com/rhel8/mariadb-105:latest		db01	run-mysqld
29 seconds ago	Exited (1) 29 seconds ago			

El comando `podman container logs` muestra un error de permiso para el directorio `/var/lib/mysql/data`.

```
[user@host ~]$ podman container logs db01
...output omitted...
---> 16:41:25      Initializing database ...
---> 16:41:25      Running mysql_install_db ...
mkdir: cannot create directory '/var/lib/mysql/data': Permission denied
Fatal error Can't create database directory '/var/lib/mysql/data'
```

Este error ocurre debido al contexto de SELinux incorrecto que está configurado en el directorio `/home/user/db_data` en la máquina `host`.

Contextos de SELinux para almacenamiento en contenedores

Debe establecer el tipo de contexto de SELinux `container_file_t` para poder montar el directorio como almacenamiento persistente en un contenedor. Si el directorio no tiene el contexto `container_file_t` de SELinux, el contenedor no puede acceder al directorio. Puede agregar la opción `Z` al argumento de la opción `-v` del comando `podman run` para establecer automáticamente el contexto de SELinux en el directorio.

Por lo tanto, use el comando `podman run -v /home/user/dbfiles:/var/lib/mysql:Z` para establecer el contexto de SELinux para el directorio `/home/user/dbfiles` cuando lo monte como almacenamiento persistente para el directorio `/var/lib/mysql`.

```
[user@host ~]$ podman run -d --name db01 \
-e MYSQL_USER=student \
-e MYSQL_PASSWORD=student \
-e MYSQL_DATABASE=dev_data \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/user/db_data:/var/lib/mysql:Z \
registry.lab.example.com/rhel8/mariadb-105
```

A continuación, verifica que el contexto SELinux correcto esté configurado en el directorio `/home/user/dbfiles` con la opción `-Z` del comando `ls`.

```
[user@host ~]$ ls -Z /home/user/
system_u:object_r:container_file_t:s0:c81,c1009 dbfiles
...output omitted...
```

Asignar una asignación de puertos a contenedores

Para proporcionar acceso de red a los contenedores, los clientes deben conectarse a los puertos en el `host` de contenedores que pasan el tráfico de red a través de los puertos en el contenedor. Cuando asigna un puerto de red en el `host` de contenedores a un puerto en el contenedor, el contenedor recibe el tráfico de red enviado al puerto de red del `host`.

Por ejemplo, puede asignar el puerto 13306 en el host del contenedor al puerto 3306 en el contenedor para la comunicación con el contenedor MariaDB. Por lo tanto, el tráfico enviado al puerto del host de contenedores 13306 sería recibido por MariaDB que se ejecuta en el contenedor.

Use la opción `-p` del comando `podman run` para definir una asignación de puertos desde el puerto 13306 desde el host del contenedor al puerto 3306 en el contenedor `db01`.

```
[user@host ~]$ podman run -d --name db01 \
-e MYSQL_USER=student \
-e MYSQL_PASSWORD=student \
-e MYSQL_DATABASE=dev_data \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/user/db_data:/var/lib/mysql:Z \
-p 13306:3306 \
registry.lab.example.com/rhel8/mariadb-105
```

Use la opción `-a` del comando `podman port` para mostrar todas las asignaciones de puertos de contenedores en uso. También puede usar el comando `podman port db01` para mostrar los puertos asignados para el contenedor `db01`.

```
[user@host ~]$ podman port -a
1c22fd905120 3306/tcp -> 0.0.0.0:13306
[user@host ~]$ podman port db01
3306/tcp -> 0.0.0.0:13306
```

Use el comando `firewall-cmd` para permitir el tráfico del puerto 13306 en la máquina host del contenedor para que pueda ser redirigido al contenedor.

```
[root@host ~]# firewall-cmd --add-port=13306/tcp --permanent
[root@host ~]# firewall-cmd --reload
```



Importante

Un contenedor rootless no puede abrir un puerto privilegiado (puertos inferiores a 1024) en el contenedor. Es decir, el comando `podman run -p 80:8080` normalmente no funciona para un contenedor rootless en ejecución. Para asignar un puerto en el host de contenedores inferior a 1024 a un puerto de contenedores, debe ejecutar Podman como usuario root o realizar otros ajustes en el sistema.

Puede asignar un puerto superior a 1024 en el host de contenedores a un puerto con privilegios en el contenedor, incluso si está ejecutando un contenedor rootless. La asignación de `8080:80` funciona si el contenedor proporciona un servicio que escucha en el puerto 80.

Configuración de DNS en un contenedor

Podman v4.0 soporta dos backends de red para contenedores: Netavark y CNI. A partir de RHEL 9, los sistemas usan Netavark de manera predeterminada. Para verificar qué backend de red se usa, ejecute el siguiente comando `podman info`.

```
[user@host ~]$ podman info --format {{.Host.NetworkBackend}}
netavark
```

**nota**

El metapaquete `container-tools` incluye los paquetes `netavark` y `aardvark-dns`. Si Podman se instaló como un paquete independiente, o si el metapaquete `container-tools` se instaló más tarde, el resultado del comando anterior podría ser `cni`. Para cambiar el backend de la red, defina la siguiente configuración en el archivo `/usr/share/containers/containers.conf`:

```
[network]
...output omitted...
network_backend = "netavark"
```

Los contenedores existentes en el host que usan la red Podman predeterminada no pueden resolver los nombres de host de los demás porque el DNS no está habilitado en la red predeterminada.

Use el comando `podman network create` para crear una red habilitada para DNS. Use el comando `podman network create` para crear la red denominada `db_net` y especifique la subred como `10.87.0.0/16` y la puerta de enlace como `10.87.0.1`.

```
[user@host ~]$ podman network create --gateway 10.87.0.1 \
--subnet 10.87.0.0/16 db_net
db_net
```

Si no especifica las opciones `--gateway` o `--subnet`, se crean con los valores predeterminados.

El comando `podman network inspect` muestra información acerca de una red específica. Use el comando `podman network inspect` para verificar que la puerta de enlace y la subred se hayan configurado correctamente y que la nueva red `db_net` esté habilitada para DNS.

```
[user@host ~]$ podman network inspect db_net
[
  {
    "name": "db_net",
    ...output omitted...
    "subnets": [
      {
        "subnet": "10.87.0.0/16",
        "gateway": "10.87.0.1"
      }
    ],
    ...output omitted...
    "dns_enabled": true,
    ...output omitted...
```

Puede agregar la red habilitada para DNS `db_net` a un nuevo contenedor con la opción `--network` del comando `podman run`. Use la opción `--network` del comando `podman run` para crear los contenedores `db01` y `client01` que están conectados a la red `db_net`.

```
[user@host ~]$ podman run -d --name db01 \
-e MYSQL_USER=student \
-e MYSQL_PASSWORD=student \
-e MYSQL_DATABASE=dev_data \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/user/db_data:/var/lib/mysql:Z \
-p 13306:3306 \
--network db_net \
registry.lab.example.com/rhel8/mariadb-105
[user@host ~]$ podman run -d --name client01 \
--network db_net \
registry.lab.example.com/ubi8/ubi:latest \
sleep infinity
```

Debido a que los contenedores están diseñados para tener solo los paquetes mínimos requeridos, es posible que los contenedores no tengan las utilidades requeridas para probar la comunicación, como los comandos `ping` y `ip`. Puede instalar estas utilidades en el contenedor con el comando `podman exec`.

```
[user@host ~]$ podman exec -it db01 dnf install -y iputils iproute
...output omitted...
[user@host ~]$ podman exec -it client01 dnf install -y iputils iproute
...output omitted...
```

Los contenedores ahora pueden hacerse ping entre sí por nombre de contenedor. Pruebe la resolución de DNS con el comando `podman exec`. Los nombres se resuelven en las IP dentro de la subred que se configuró manualmente para la red `db_net`.

```
[user@host ~]$ podman exec -it db01 ping -c3 client01
PING client01.dns.podman (10.87.0.4) 56(84) bytes of data.
64 bytes from 10.87.0.4 (10.87.0.4): icmp_seq=1 ttl=64 time=0.049 ms
...output omitted...
--- client01.dns.podman ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.049/0.060/0.072/0.013 ms

[user@host ~]$ podman exec -it client01 ping -c3 db01
PING db01.dns.podman (10.87.0.3) 56(84) bytes of data.
64 bytes from 10.87.0.3 (10.87.0.3): icmp_seq=1 ttl=64 time=0.021 ms
...output omitted...
--- db01.dns.podman ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2047ms
rtt min/avg/max/mdev = 0.021/0.040/0.050/0.013 ms
```

Verifique que las direcciones IP en cada contenedor coincidan con la resolución de DNS con el comando `podman exec`.

```
[user@host ~]$ podman exec -it db01 ip a | grep 10.8
    inet 10.87.0.3/16 brd 10.87.255.255 scope global eth0
    inet 10.87.0.4/16 brd 10.87.255.255 scope global eth0
[user@host ~]$ podman exec -it client01 ip a | grep 10.8
    inet 10.87.0.3/16 brd 10.87.255.255 scope global eth0
    inet 10.87.0.4/16 brd 10.87.255.255 scope global eth0
```

Varias redes en un solo contenedor

Se pueden conectar varias redes a un contenedor al mismo tiempo para ayudar a separar diferentes tipos de tráfico.

Puede usar el comando `podman network create` para crear la red `backend`.

```
[user@host ~]$ podman network create backend
```

A continuación, use el comando `podman network ls` para ver todas las redes de Podman.

```
[user@host ~]$ podman network ls
NETWORK ID      NAME      DRIVER
a7fea510a6d1    backend   bridge
fe680efc5276    db01      bridge
2f259bab93aa    podman    bridge
```

La subred y la puerta de enlace no se especificaron con las opciones `--gateway` y `--subnet` del comando `podman network create`.

Use el comando `podman network inspect` para obtener la información IP de la red `backend`.

```
[user@host ~]$ podman network inspect backend
[
  {
    "name": "backend",
    ...output omitted...
    "subnets": [
      {
        "subnet": "10.89.1.0/24",
        "gateway": "10.89.1.1"
      }
    ]
  }
]
...output omitted...
```

Puede usar el comando `podman network connect` para conectar redes adicionales a un contenedor cuando se está ejecutando. Puede usar el comando `podman network connect` para conectar la red `backend` a los contenedores `db01` y `client01`.

```
[user@host ~]$ podman network connect backend db01
[user@host ~]$ podman network connect backend client01
```

**Importante**

Si no se especifica una red con el comando `podman run`, el contenedor se conecta a la red predeterminada. La red predeterminada usa el modo de red `slirp4netns`, y las redes que crea con el comando `podman network create` usan el modo de red `punto`. Si intenta conectar una red de puente a un contenedor con el modo de red `slirp4netns`, el comando falla:

```
Error: "slirp4netns" is not supported: invalid network mode
```

Use el comando `podman inspect` para verificar que ambas redes estén conectadas a cada contenedor y para mostrar la información de IP.

```
[user@host ~]$ podman inspect db01
...output omitted...
      "backend": {
        "EndpointID": "",
        "Gateway": "10.89.1.1",
        "IPAddress": "10.89.1.4",
...output omitted...
      },
      "db_net": {
        "EndpointID": "",
        "Gateway": "10.87.0.1",
        "IPAddress": "10.87.0.3",
...output omitted...
[user@host ~]$ podman inspect client01
...output omitted...
      "backend": {
        "EndpointID": "",
        "Gateway": "10.89.1.1",
        "IPAddress": "10.89.1.5",
...output omitted...
      },
      "db_net": {
        "EndpointID": "",
        "Gateway": "10.87.0.1",
        "IPAddress": "10.87.0.4",
...output omitted...
```

El contenedor `client01` ahora puede comunicarse con el contenedor `db01` en ambas redes. Use el comando `podman exec` para hacer ping a ambas redes en el contenedor `db01` desde el contenedor `client01`.

```
[user@host ~]$ podman exec -it client01 ping -c3 10.89.1.4 | grep 'packet loss'
3 packets transmitted, 3 received, 0% packet loss, time 2052ms
[user@host ~]$ podman exec -it client01 ping -c3 10.87.0.3 | grep 'packet loss'
3 packets transmitted, 3 received, 0% packet loss, time 2054ms
```




Referencias

Páginas del manual: `podman(1)`, `podman-exec(1)`, `podman-info(1)`, `podman-network(1)`, `podman-network-create(1)`, `podman-network-inspect(1)`, `podman-network-ls(1)`, `podman-port(1)`, `podman-run(1)` y `podman-unshare(1)`

Para obtener más información, consulte el capítulo Trabajar con contenedores en la Guía de creación, ejecución y administración de contenedores de Red Hat Enterprise Linux 9 en

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/building_running_and_managing_containers/assembly_working-with-containers_building-running-and-managing-containers

► Ejercicio Guiado

Administración del almacenamiento del contenedor y los recursos de red

En este ejercicio, pasará variables de entorno a un contenedor durante la creación, montará almacenamiento persistente en un contenedor, creará y conectará varias redes de contenedores y expondrá puertos de contenedor desde la máquina host.

Resultados

- Crear redes de contenedores y conectarlas a contenedores.
- Solucionar problemas de contenedores fallidos.
- Pasar variables de entorno a contenedores durante la creación.
- Crear y montar almacenamiento persistente en contenedores.
- Asignar puertos de host a puertos dentro de contenedores.

Antes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start containers-resources
```

Instrucciones

- 1. Inicie sesión en la máquina servera como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Cree la red del contenedor `frontend`. Cree los contenedores `db_client` y `db_01` que está conectado a la red `frontend`.
 - 2.1. Use las opciones `--subnet` y `--gateway` del comando `podman network create` para crear la red `frontend` con la subred `10.89.1.0/24` y la puerta de enlace `10.89.1.1`.

```
[student@servera ~]$ podman network create --subnet 10.89.1.0/24 \
--gateway 10.89.1.1 frontend
frontend
```

- 2.2. Inicie sesión en el registro `registry.lab.example.com`.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 2.3. Cree el contenedor `db_client` que está conectado a la red `frontend`. Monte el directorio de repositorios DNF `/etc/yum.repos.d` dentro del contenedor en `/etc/yum.repos.d` para permitir la instalación de paquetes.

```
[student@servera ~]$ podman run -d --name db_client \
--network frontend \
-v /etc/yum.repos.d:/etc/yum.repos.d \
registry.lab.example.com/ubi9-beta/ubi \
sleep infinity
e20dfed7e392abe4b7bea3c25e9cb17ef95d16af9cedd50d68f997a663ba6c15
```

- 2.4. Cree el contenedor `db_01` que está conectado a la red `frontend`.

```
[student@servera ~]$ podman run -d --name db_01 --network frontend \
registry.lab.example.com/rhel8/mariadb-105
3e767ae6eea4578152a216beb5ae98c8ef03a2d66098debe2736b8b458bab405
```

- 2.5. Ver todos los contenedores.

```
[student@servera ~]$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND
e20dfed7e392	registry.lab.example.com/ubi8/ubi:latest	sleep infinity
56 seconds ago	Up 56 seconds ago	db_client
3e767ae6eea4	registry.lab.example.com/rhel8/mariadb-105:latest	run-mysqld 1
second ago	Exited (1) 1 second ago	db_01

- 3. Solucione los problemas del contenedor `db_01` y determine por qué no se está ejecutando. Vuelva a crear el contenedor `db_01` con las variables de entorno requeridas.

- 3.1. Vea los registros del contenedor y determine por qué se cerró el contenedor.

```
[student@servera ~]$ podman container logs db_01
...output omitted...
You must either specify the following environment variables:
  MYSQL_USER (regex: '[a-zA-Z0-9_]+$')
  MYSQL_PASSWORD (regex: '[a-zA-Z0-9_~!@#%&*()-=<>,.;:|]+$')
  MYSQL_DATABASE (regex: '[a-zA-Z0-9_]+$')
Or the following environment variable:
  MYSQL_ROOT_PASSWORD (regex: '[a-zA-Z0-9_~!@#%&*()-=<>,.;:|]+$')
Or both.
...output omitted...
```

- 3.2. Elimine el contenedor `db_01` y vuelva a crearlo con variables de entorno. Proporcione variables de entorno requeridas.

```
[student@servera ~]$ podman rm db_01
3e767ae6eea4578152a216beb5ae98c8ef03a2d66098debe2736b8b458bab405
[student@servera ~]$ podman run -d --name db_01 \
--network frontend \
-e MYSQL_USER=dev1 \
-e MYSQL_PASSWORD=devpass \
-e MYSQL_DATABASE=devdb \
-e MYSQL_ROOT_PASSWORD=redhat \
registry.lab.example.com/rhel8/mariadb-105
948c4cd767b561432056e77adb261ab4024c1b66a22af17861aba0f16c66273b
```

3.3. Vea los contenedores en ejecución actuales.

```
[student@servera ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
e20dfed7e392	registry.lab.example.com/ubi8/ubi:latest	sleep infinity
56 seconds ago	Up 56 seconds ago	db_client
948c4cd767b5	registry.lab.example.com/rhel8/mariadb-105:latest	run-mysqld
11 seconds ago	Up 12 seconds ago	db_01

- 4. Cree almacenamiento persistente para el servicio MariaDB contenerizada y asigne el puerto 13306 de la máquina local al puerto 3306 en el contenedor. Permita el tráfico al puerto 13306 en la máquina servera.

4.1. Cree el directorio /home/student/database en la máquina servera.

```
[student@servera ~]$ mkdir /home/student/databases
```

4.2. Obtenga el mysql UID y el GID del contenedor db_01 y, luego, elimine el contenedor db01.

```
[student@servera ~]$ podman exec -it db_01 grep mysql /etc/passwd
mysql:x:27:27:MySQL Server:/var/lib/mysql:/sbin/nologin
[student@servera ~]$ podman stop db_01
db_01
[student@servera ~]$ podman rm db_01
948c4cd767b561432056e77adb261ab4024c1b66a22af17861aba0f16c66273b
```

4.3. Ejecute el comando chown dentro del espacio de nombres del contenedor y establezca el usuario y el propietario del grupo en 27 en el directorio /home/student/database.

```
[student@servera ~]$ podman unshare chown 27:27 /home/student/databases/
[student@servera ~]$ ls -l /home/student/
total 0
drwxr-xr-x. 2 100026 100026 6 May  9 17:40 databases
```

4.4. Cree el contenedor db_01 y monte el directorio /home/student/databases desde la máquina servera al directorio /var/lib/mysql dentro del contenedor db_01. Use la opción Z para aplicar el contexto de SELinux requerido.

```
[student@servera ~]$ podman run -d --name db_01 \
--network frontend \
-e MYSQL_USER=dev1 \
-e MYSQL_PASSWORD=devpass \
-e MYSQL_DATABASE=devdb \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/student/databases:/var/lib/mysql:Z \
-p 13306:3306 \
registry.lab.example.com/rhel8/mariadb-105
```

4.5. Instale el paquete mariadb en el contenedor db_client.

```
[student@servera ~]$ podman exec -it db_client dnf install -y mariadb
...output omitted...
Complete!
```

4.6. Cree la tabla crucial_data en la base de datos dev_db en el contenedor db_01 desde el contenedor db_client.

```
[student@servera ~]$ podman exec -it db_client mysql -u dev1 -p -h db_01
Enter password: devpass
...output omitted...
MariaDB [(none)]> USE devdb;
Database changed
MariaDB [devdb]> CREATE TABLE crucial_data(column1 int);
Query OK, 0 rows affected (0.036 sec)

MariaDB [devdb]> SHOW TABLES;
+-----+
| Tables_in_devdb |
+-----+
| crucial_data    |
+-----+
1 row in set (0.001 sec)

MariaDB [devdb]> quit
Bye
```

4.7. Permita el tráfico al puerto 13306 en el firewall en la máquina servera.

```
[student@servera ~]$ sudo firewall-cmd --add-port=13306/tcp --permanent
[sudo] password for student: student
success
[student@servera ~]$ sudo firewall-cmd --reload
success
```

4.8. Abra un segundo terminal en la máquina Workstation y use el cliente MariaDB para conectarse a la máquina servera en el puerto 13306 para mostrar las tablas en el contenedor db_01 que están almacenadas en el almacenamiento persistente.

```
[student@workstation ~]$ mysql -u dev1 -p -h servera --port 13306 \
devdb -e 'SHOW TABLES';
Enter password: devpass
+-----+
| Tables_in_devdb |
+-----+
| crucial_data    |
+-----+
```

- 5. Cree una segunda red de contenedores denominada **backend** y conecte la red **backend** a los contenedores **db_client** y **db_01**. Pruebe la conectividad de red y la resolución de DNS entre los contenedores.

- 5.1. Cree la red **backend** con la subred **10.90.0.0/24** y la puerta de enlace **10.90.0.1**.

```
[student@servera ~]$ podman network create --subnet 10.90.0.0/24 \
--gateway 10.90.0.1 backend
backend
```

- 5.2. Conecte la red del contenedor **backend** a los contenedores **db_client** y **db_01**.

```
[student@servera ~]$ podman network connect backend db_client
[student@servera ~]$ podman network connect backend db_01
```

- 5.3. Obtenga las direcciones IP del contenedor **db_01**.

```
[student@servera ~]$ podman inspect db_01
...output omitted...
    "Networks": {
      "backend": {
        "EndpointID": "",
        "Gateway": "10.90.0.1",
        "IPAddress": "10.90.0.3",
...output omitted...
      "frontend": {
        "EndpointID": "",
        "Gateway": "10.89.1.1",
        "IPAddress": "10.89.1.6",
...output omitted...
```

- 5.4. Instale el paquete **iputils** en el contenedor **db_client**.

```
[student@servera ~]$ podman exec -it db_client dnf install -y iputils
...output omitted...
Complete!
```

- 5.5. Haga ping al nombre del contenedor **db_01** desde el contenedor **db_client**.

```
[student@servera ~]$ podman exec -it db_client ping -c4 db_01
PING db_01.dns.podman (10.90.0.3) 56(84) bytes of data.
...output omitted...
--- db_01.dns.podman ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.043/0.049/0.054/0.004 ms
```

5.6. Salga de la máquina servera.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish containers-resources
```

Esto concluye la sección.

Administración de contenedores como servicios del sistema

Objetivos

Configurar un contenedor como servicio `systemd` y configurar un servicio de contenedor para que se inicie en el momento del arranque.

Administración de entornos de contenedores pequeños con unidades `systemd`

Puede ejecutar un contenedor para completar una tarea del sistema o para obtener la salida de una serie de comandos. También es posible que desee ejecutar contenedores que ejecutan un servicio de forma indefinida, como servidores web o bases de datos. En un entorno tradicional, un usuario con privilegios generalmente configura estos servicios para que se ejecuten en el arranque del sistema y los administra con el comando `systemctl`.

Como usuario regular, puede crear una unidad `systemd` para configurar sus contenedores `rootless`. Puede usar esta configuración para administrar su contenedor como un servicio del sistema regular con el comando `systemctl`.

La administración de contenedores basados en unidades `systemd` es principalmente útil para implementaciones básicas y pequeñas que no necesitan escalarse. Para un escalamiento y una orquestación más sofisticados de muchas aplicaciones y servicios basados en contenedores, puede usar una plataforma de orquestación empresarial basada en Kubernetes, como Red Hat OpenShift Container Platform.

Para analizar los temas de esta clase, imagine el siguiente escenario.

Como administrador del sistema, debe configurar el contenedor `webserver1` que se basa en la imagen de contenedor `http24` para iniciar un arranque del sistema. También debe montar el directorio `/app-artifacts` para el contenido del servidor web y asignar el puerto 8080 desde la máquina local al contenedor. Configure el contenedor para que se inicie y se detenga con los comandos `systemctl`.

Requisitos para los servicios de usuario de `systemd`

Como usuario regular, puede habilitar un servicio con el comando `systemctl`. El servicio se inicia cuando abre una sesión (interfaz gráfica, consola de texto o SSH) y se detiene cuando cierra la última sesión. Este comportamiento difiere de un servicio del sistema, que se inician cuando arranca el sistema y se detienen cuando el sistema se apaga.

De manera predeterminada, cuando crea una cuenta de usuario con el comando `useradd`, el sistema usa el siguiente ID disponible del rango de ID de usuario regular. El sistema también reserva un rango de ID para los contenedores del usuario en el archivo `/etc/subuid`. Si crea una cuenta de usuario con la opción `--system` del comando `useradd`, el sistema no reserva un rango para los contenedores de usuarios. Como consecuencia, no puede iniciar contenedores `rootless` con cuentas del sistema.

Decide crear una cuenta de usuario dedicada para administrar contenedores. Use el comando `useradd` para crear el usuario `appdev-adm` y use la contraseña `redhat`.


```
[user@host ~]$ sudo useradd appdev-adm
myapp.service
[user@host ~]$ sudo passwd appdev-adm
Changing password for user appdev-adm.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

A continuación, use el comando `su` para cambiar al usuario `appdev-adm` y comience a usar el comando `podman`.

```
[user@host ~]$ su appdev-adm
Password: redhat
[appdev-adm@host ~]$ podman info
ERROR[0000] XDG_RUNTIME_DIR directory "/run/user/1000" is not owned by the current
user
[appdev-adm@host ~]$
```

Podman es una utilidad sin estado que requiere una sesión de inicio de sesión completa. Podman debe usarse dentro de una sesión SSH y no puede usarse en una shell `sudo` o `su`. Por lo tanto, salga de la shell `su` e inicie sesión en la máquina mediante SSH.

```
[appdev-adm@host ~]$ exit
[user@host ~]$ exit
[user@example ~]$ ssh appdev-adm@host
[appdev-adm@host ~]$
```

A continuación, configure el registro de contenedores y realice la autenticación con sus credenciales. Ejecute el contenedor `http` con el siguiente comando.

```
[appdev-adm@host ~]$ podman run -d --name webserver1 -p 8080:8080 -v \
~/app-artifacts:/var/www:Z registry.access.redhat.com/ubi8/httpd-24
af84e1ec33ea2f0d9787c56fbe7a62a4b9ce8ac03911be9e97f95575b306c297
[appdev-adm@host ~]$ podman ps -a
```

CONTAINER ID	IMAGE	STATUS	PORTS	COMMAND
af84e1ec33ea	registry.access.redhat.com/ubi8/httpd-24:latest	Exited (1) 16 seconds ago	15 seconds ago 0.0.0.0:8080->8080/tcp	/usr/bin/run-httpd
webserver1				

Observa que el contenedor `webserver1` no se inició, por lo que ejecuta el comando `podman container logs` para ver los registros del contenedor.

```
[appdev-adm@host ~]$ podman container logs webserver1
=> sourcing 10-set-mpm.sh ...
=> sourcing 20-copy-config.sh ...
=> sourcing 40-ssl-certs.sh ...
---> Generating SSL key pair for httpd...
AH00526: Syntax error on line 122 of /etc/httpd/conf/httpd.conf:
DocumentRoot '/var/www/html' is not a directory, or is not readable
[appdev-adm@servera ~]$
```

Los registros del contenedor `webserver1` muestran que el archivo `/var/www/html` no se puede leer. Este error se produce porque el directorio de montaje del contenedor no puede encontrar el subdirectorio `html`. Puede eliminar el contenedor existente, actualizar el comando y ejecutarlo de la siguiente manera:

```
[appdev-adm@host ~]$ podman run -d --name webserver1 -p 8080:8080 -v \
~/app-artifacts:/var/www/html:Z registry.access.redhat.com/ubi8/httpd-24
cde4a3d8c9563fd50cc39de8a4873dcf15a7e881ba4548d5646760eae7a35d81
[appdev-adm@host ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
cde4a3d8c956	registry.access.redhat.com/ubi8/httpd-24:latest	/usr/bin/run-http... 4 seconds ago Up 5 seconds ago 0.0.0.0:8080->8080/tcp webserver1

Creación de archivos usuarios systemd para contenedores.

Puede definir manualmente `systemd` servicios en el directorio `~/.config/systemd/user/`. La sintaxis de archivos para los servicios de usuario es la misma que para los archivos de servicios del sistema. Para obtener más detalles, consulte las páginas del manual `systemd.unit(5)` y `systemd.service(5)`.

Use el comando `podman generate systemd` para generar archivos de servicio `systemd` para un contenedor existente. El comando `podman generate systemd` usa un contenedor como modelo para crear el archivo de configuración.

La opción `--new` del comando `podman generate systemd` le indica a la utilidad `podman` que configure el servicio `systemd` para crear el contenedor cuando se inicie el servicio y eliminarlo cuando este se detenga.



Importante

Sin la opción `--new`, la utilidad `podman` configura el archivo de unidad del servicio para iniciar y detener el contenedor existente sin eliminarlo.

Use el comando `podman generate systemd` con la opción `--name` para mostrar el archivo de servicio `systemd` que se modela para el contenedor `webserver1`.

```
[appdev-adm@host ~]$ podman generate systemd --name webserver1
...output omitted...
ExecStart=/usr/bin/podman start webserver1 ❶
ExecStop=/usr/bin/podman stop -t 10 webserver1 ❷
ExecStopPost=/usr/bin/podman stop -t 10 webserver1
...output omitted...
```

- ❶ En start (iniciar), el daemon `systemd` ejecuta el comando `podman start` para iniciar el contenedor existente.
- ❷ En stop (detener), el daemon `systemd` ejecuta el comando `podman stop` para detener el contenedor. Tenga en cuenta que el daemon `systemd` no elimina el contenedor en esta acción.

A continuación, use el comando anterior con la opción `--new` para comparar la configuración `systemd`.

```
[appdev-adm@host ~]$ podman generate systemd --name webserver1 --new
...output omitted...
ExecStartPre=/bin/rm -f %t/%n.ctr-id
ExecStart=/usr/bin/podman run --cidfile=%t/%n.ctr-id --cgroups=no-common --rm --
sdnotify=common --replace -d --name webserver1 -p 8080:8080 -v /home/appdev-adm/
app-artifacts:/var/www/html:Z registry.access.redhat.com/ubi8/httpd-24 ❶
ExecStop=/usr/bin/podman stop --ignore --cidfile=%t/%n.ctr-id ❷
ExecStopPost=/usr/bin/podman rm -f --ignore --cidfile=%t/%n.ctr-id ❸
...output omitted...
```

- ❶ En start (iniciar), el daemon `systemd` ejecuta el comando `podman run` para crear y, luego, iniciar un nuevo contenedor. Esta acción usa la opción `--rm` del comando `podman run` que elimina el contenedor al detenerse.
- ❷ En stop (detener), `systemd` ejecuta el comando `podman stop` para detener el contenedor.
- ❸ Después de que `systemd` haya detenido el contenedor, `systemd` lo elimina mediante el comando `podman rm -f`.

Verifique la salida del comando `podman generate systemd` y ejecute el comando anterior con la opción `--files` para crear el archivo de usuario `systemd` en el directorio actual. Debido a que el contenedor `webserver1` usa el almacenamiento persistente, usted opta por usar el comando `podman generate systemd` con la opción `--new`. A continuación, cree el directorio `~/ .config/systemd/user/` y mueva el archivo a esta ubicación.

```
[appdev-adm@host ~]$ podman generate systemd --name webserver1 --new --files
/home/appdev-adm/container-webserver1.service
[appdev-adm@host ~]$ mkdir -p ~/.config/systemd/user/
[appdev-adm@host ~]$ mv container-webserver1.service ~/.config/systemd/user/
```

Administración de archivos usuarios `systemd` para contenedores.

Ahora que creó el archivo de usuario `systemd`, puede usar la opción `--user` del comando `systemctl` para administrar el contenedor `webserver1`.

Primero, recargue el daemon `systemd` para que el comando `systemctl` conozca el nuevo archivo de usuario. Use el comando `systemctl --user start` para iniciar el contenedor `webserver1`. Use el nombre del archivo de usuario `systemd` generado para el contenedor.

```
[appdev-adm@host ~]$ systemctl --user start container-webserver1.service
[appdev-adm@host ~]$ systemctl --user status container-webserver1.service
• container-webserver1.service - Podman container-webserver1.service
  Loaded: loaded (/home/appdev-adm/.config/systemd/user/container-webserver1.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2022-04-28 21:22:26 EDT; 18s ago
  Docs: man:podman-generate-systemd(1)
  Process: 31560 ExecStartPre=/bin/rm -f /run/user/1003/container-webserver1.service.ctr-id (code=exited, status=0/SUCCESS)
  Main PID: 31600 (comon)
  ...output omitted...
[appdev-adm@host ~]$ podman ps
CONTAINER ID   IMAGE                                     COMMAND
CREATED       STATUS      PORTS          NAMES
18eb00f42324   registry.access.redhat.com/ubi8/httpd-24:latest /usr/bin/run-http... 28 seconds ago Up 29 seconds ago 0.0.0.0:8080->8080/tcp webserver1
Created symlink /home/appdev-adm/.config/systemd/user/default.target.wants/container-webserver1.service -> /home/appdev-adm/.config/systemd/user/container-webserver1.service.
```



Importante

Al configurar un contenedor con el daemon `systemd`, el daemon monitorea el estado del contenedor y lo reinicia si falla. No use el comando `podman` para iniciar o detener estos contenedores. Si lo hace, puede interferir con el monitoreo del daemon `systemd`.

En la siguiente tabla, se resumen los diferentes directorios y comandos que se usan entre los servicios de usuario y del sistema `systemd`.

Comparación de servicios de usuario y del sistema

Almacenamiento de archivos de unidad personalizados	Servicios del sistema	<code>/etc/systemd/system/unit.service</code>
	Servicios de usuario	<code>~/.config/systemd/user/unit.service</code>
Recarga de archivos de unidad	Servicios del sistema	<code># systemctl daemon-reload</code>
	Servicios de usuario	<code>\$ systemctl --user daemon-reload</code>
Inicio y detención de un servicio	Servicios del sistema	<code># systemctl start UNIT</code> <code># systemctl stop UNIT</code>

	Servicios de usuario	<pre>\$ systemctl --user start UNIT \$ systemctl --user stop UNIT</pre>
Inicio de un servicio cuando se inicia la máquina	Servicios del sistema	<pre># systemctl enable UNIT</pre>
	Servicios de usuario	<pre>\$ loginctl enable-linger \$ systemctl --user enable UNIT</pre>

Configuración de contenedores para que se inicien en el arranque del sistema

Ahora que la configuración `systemd` para el contenedor está completa, sale de la sesión SSH. Algún tiempo después, se le notifica que el contenedor se detiene después de salir de la sesión.

Puede cambiar este comportamiento predeterminado y obligar a los servicios habilitados a que se inicien con el servidor y se detengan durante el apagado mediante la ejecución del comando `loginctl enable-linger`.

Use el comando `loginctl` para configurar el servicio de usuario `systemd` para que persista después de que se cierre la última sesión de usuario del servicio configurado. A continuación, verifique la configuración correcta con el comando `loginctl show-user`.

```
[user@host ~]$ loginctl show-user appdev-adm
...output omitted...
Linger=no
[user@host ~]$ loginctl enable-linger
[user@host ~]$ loginctl show-user appdev-adm
...output omitted...
Linger=yes
```

Para revertir la operación, use el comando `loginctl disable-linger`.

Administración de contenedores como root con systemd

También puede configurar contenedores para que se ejecuten como root y administrarlos con los archivos de servicio `systemd`. Una de las ventajas de este enfoque es que puede configurar los archivos de unidad para que funcionen exactamente igual que los archivos de unidad `systemd` normales, en lugar de como un usuario en particular.

El procedimiento para definir el archivo de servicio como root es similar al procedimiento descrito anteriormente para contenedores rootless, con las siguientes excepciones:

- No cree un usuario dedicado para la administración de contenedores.
- El archivo de servicio debe estar en el directorio `/etc/systemd/system` en lugar del directorio `~/.config/systemd/user`.
- Los contenedores se administran con el comando `systemctl` sin la opción `--user`.
- No ejecute el comando `loginctl enable-linger` con el usuario `root`.

Para obtener una demostración, consulte el video de YouTube del canal de videos de Red Hat que se detalla en las referencias al final de esta sección.



Referencias

Páginas del manual `loginctl(1)`, `systemd.unit(5)`, `systemd.service(5)`, `subuid(5)` y `podman-generate-systemd(1)`

Administración de contenedores en Podman con archivos de unidad systemd

<https://www.youtube.com/watch?v=AGkM2jGT61Y>

Para obtener más información, consulte el capítulo *Running Containers as Systemd Services with Podman* de la *Red Hat Enterprise Linux 9 Building, Running, and Managing Containers Guide* en https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html-single/building_running_and_managing_containers/index

► Ejercicio Guiado

Administración de contenedores como servicios del sistema

En este ejercicio, configurará un contenedor para administrar como un servicio `systemd` y, luego, usa los comandos `systemctl` para administrar ese contenedor, de modo que se inicie automáticamente cuando se inicia la máquina `host`.

Resultados

- Crear archivos de servicio `systemd` para administrar un contenedor.
- Configurar un contenedor para que pueda administrarlo con los comandos `systemctl`.
- Configurar cuentas de usuario para que los servicios de usuario `systemd` inicien un contenedor cuando se inicia la máquina `host`.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start containers-services
```

Instrucciones

- 1. Inicie sesión en la máquina `servera` como el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 2. Cree una cuenta de usuario denominada `contsvc` y use `redhat` como contraseña. Use esta cuenta de usuario para ejecutar contenedores como servicios `systemd`.

- 2.1. Cree el usuario `contsvc`. Establezca `redhat` como contraseña para el usuario `contsvc`.

```
[student@servera ~]$ sudo useradd contsvc
[sudo] password for student: student
[student@servera ~]$ sudo passwd contsvc
Changing password for user contsvc.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

- 2.2. Para administrar los servicios de usuario `systemd` con la cuenta `contsvc`, debe iniciar sesión directamente con el usuario `contsvc`. No puede usar los comandos `su` y `sudo` para crear una sesión con el usuario `contsvc`.

Regrese a la máquina `workstation` como el usuario `student` y, luego, inicie sesión como el usuario `contsvc`.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$ ssh contsvc@servera
...output omitted...
[contsvc@servera ~]$
```

- ▶ 3. Configure el acceso al registro del aula `registry.lab.example.com` en su directorio de inicio. Use el archivo `/tmp/containers-services/registries.conf` como plantilla.

- 3.1. Cree el directorio `~/.config/containers/`.

```
[contsvc@servera ~]$ mkdir -p ~/.config/containers/
```

- 3.2. El script `lab` prepara el archivo `registries.conf` en el directorio `/tmp/containers-services/`. Copie ese archivo en el directorio `~/.config/containers/`.

```
[contsvc@servera ~]$ cp /tmp/containers-services/registries.conf \
~/.config/containers/
```

- 3.3. Verifique que pueda acceder al registro `registry.lab.example.com`. Si todo funciona según lo previsto, el comando debe enumerar algunas imágenes.

```
[contsvc@servera ~]$ podman search ubi
NAME                                DESCRIPTION
registry.lab.example.com/ubi7/ubi
registry.lab.example.com/ubi8/ubi
registry.lab.example.com/ubi9-beta/ubi
```

- ▶ 4. Use el directorio `/home/contsvc/webcontent/html/` como almacenamiento persistente para el contenedor del servidor web. Cree la página de prueba `index.html` con la línea `Hello World` dentro del directorio.

- 4.1. Cree el directorio `~/webcontent/html/`.

```
[contsvc@servera ~]$ mkdir -p ~/webcontent/html/
```

- 4.2. Cree el archivo `index.html` y agregue la línea `Hello World`.

```
[contsvc@servera ~]$ echo "Hello World" > ~/webcontent/html/index.html
```

- 4.3. Confirme que el permiso para otros esté definido en `r--` en el archivo `index.html`. El contenedor usa un usuario sin privilegios que debe ser capaz de leer el archivo `index.html`.


```
[contsvc@servera ~]$ ls -ld webcontent/html/
drwxr-xr-x. 2 contsvc contsvc 24 Aug 28 04:56 webcontent/html/
[contsvc@servera ~]$ ls -l webcontent/html/index.html
-rw-r--r--. 1 contsvc contsvc 12 Aug 28 04:56 webcontent/html/index.html
```

- 5. Use la imagen `registry.lab.example.com/rhel8/httpd-24:1-105` para ejecutar un contenedor denominado `webapp` en modo separado. Redirija el puerto 8080 en el host local al puerto del contenedor 8080. Monte el directorio `~/webcontent` desde el host en el directorio `/var/www` del contenedor.

- 5.1. Inicie sesión en el registro `registry.lab.example.com` con el usuario `admin` y `redhat321` como contraseña.

```
[contsvc@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 5.2. Use la imagen `registry.lab.example.com/rhel8/httpd-24:1-163` para ejecutar un contenedor denominado `webapp` en modo separado. Use la opción `-p` para asignar el puerto 8080 en `servera` al puerto 8080 en el contenedor. Use la opción `-v` para montar el directorio `~/webcontent` en `servera` para el directorio `/var/www` en el contenedor.

```
[contsvc@servera ~]$ podman run -d --name webapp -p 8080:8080 -v \
~/webcontent:/var/www:Z registry.access.redhat.com/ubi8/httpd-24:1-163
750a681bd37cb6825907e9be4347eec2c4cd79550439110fc6d41092194d0e06
...output omitted...
```

- 5.3. Verifique que el servicio web esté funcionando en el puerto 8080.

```
[contsvc@servera ~]$ curl http://localhost:8080
Hello World
```

- 6. Cree un archivo de servicio `systemd` para administrar el contenedor `webapp` con los comandos `systemctl`. Configure el servicio `systemd` para que cuando inicie el servicio, el daemon `systemd` cree un contenedor. Después de finalizar la configuración, detenga y elimine el contenedor `webapp`. Recuerde que el daemon `systemd` espera que el contenedor no exista inicialmente.

- 6.1. Cree y cambie el directorio `~/config/systemd/user/`.

```
[contsvc@servera ~]$ mkdir -p ~/.config/systemd/user/
[contsvc@servera ~]$ cd ~/.config/systemd/user
```

- 6.2. Cree el archivo de unidad para el contenedor `webapp`. Use la opción `--new` para que `systemd` cree un contenedor al iniciar el servicio y elimine el contenedor al detenerlo.

```
[contsvc@servera user]$ podman generate systemd --name webapp --files --new
/home/contsvc/.config/systemd/user/container-webapp.service
```

6.3. Detenga y, luego, elimine el contenedor `webapp`.

```
[contsvc@servera user]$ podman stop webapp
webapp
[contsvc@servera user]$ podman rm webapp
750a681bd37cb6825907e9be4347eec2c4cd79550439110fc6d41092194d0e06
[contsvc@servera user]$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

- ▶ 7. Vuelva a cargar la configuración del daemon `systemd` y, luego, habilite e inicie su nuevo servicio de usuario `container-webapp`. Verifique la configuración del servicio `systemd`, detenga e inicie el servicio y muestre la respuesta del servidor web y el estado del contenedor.

7.1. Recargue la configuración para reconocer el nuevo archivo de unidad.

```
[contsvc@servera user]$ systemctl --user daemon-reload
```

7.2. Habilite e inicie el servicio `container-webapp`.

```
[contsvc@servera user]$ systemctl --user enable --now container-webapp
Created symlink /home/contsvc/.config/systemd/user/multi-user.target.wants/
container-webapp.service → /home/contsvc/.config/systemd/user/container-
webapp.service.
Created symlink /home/contsvc/.config/systemd/user/default.target.wants/container-
webapp.service → /home/contsvc/.config/systemd/user/container-webapp.service.
```

7.3. Verifique que el servidor web responda a solicitudes.

```
[contsvc@servera user]$ curl http://localhost:8080
Hello World
```

7.4. Verifique que el contenedor esté ejecutándose.

```
[contsvc@servera user]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
3e996db98071	registry.access.redhat.com/ubi8/httpd-24:1-163	/usr/bin/run-http...

```
3 minutes ago Up 3 minutes ago 0.0.0.0:8080->8080/tcp webapp
```

Observe el ID del contenedor. Use esta información para confirmar que `systemd` crea un contenedor cuando reinicia el servicio.

7.5. Detenga el servicio `container-webapp` y confirme que el contenedor ya no existe. Cuando detenga el servicio, `systemd` se detiene y, luego, elimina el contenedor.

```
[contsvc@servera user]$ systemctl --user stop container-webapp
[contsvc@servera user]$ podman ps --all
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES

7.6. Inicie el servicio `container-webapp` y, luego, confirme que el contenedor se esté ejecutando.

El ID del contenedor es diferente, ya que el daemon `systemd` crea un contenedor con la instrucción `start` y elimina el contenedor con la instrucción `stop`.

```
[contsvc@servera user]$ systemctl --user start container-webapp
[contsvc@servera user]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
4584b4df514c	registry.access.redhat.com/ubi8/httpd-24:1-163	/usr/bin/run-http...
6 seconds ago	Up 7 seconds ago	0.0.0.0:8080->8080/tcp webapp

- ▶ 8. Asegúrese de que los servicios para el usuario `contsvc` se inicien en el arranque del sistema. Cuando finalice, reinicie la máquina `servera`.

8.1. Ejecute el comando `loginctl enable-linger`.

```
[contsvc@servera user]$ loginctl enable-linger
```

8.2. Confirme que la opción `Linger` esté establecida para el usuario `contsvc`.

```
[contsvc@servera user]$ loginctl show-user contsvc
...output omitted...
Linger=yes
```

8.3. Cambie al usuario `root` y, luego, use el comando `systemctl reboot` para reiniciar `servera`.

```
[contsvc@servera user]$ su -
Password: redhat
Last login: Fri Aug 28 07:43:40 EDT 2020 on pts/0
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- ▶ 9. Cuando la máquina `servera` vuelva a estar activa, inicie sesión en `servera` con el usuario `contsvc`. Confirme que `systemd` inició el contenedor `webapp` y que el contenido web esté disponible.

9.1. Inicie sesión en `servera` con el usuario `contsvc`.

```
[student@workstation ~]$ ssh contsvc@servera
...output omitted...
```

9.2. Verifique que el contenedor esté ejecutándose.

```
[contsvc@servera ~]$ podman ps
```

CONTAINER ID	IMAGE	COMMAND
6c325bf49f84	registry.access.redhat.com/ubi8/httpd-24:1-163	/usr/bin/run-http...
2 minutes ago	Up 2 minutes ago	0.0.0.0:8080->8080/tcp webapp

9.3. Acceda al contenido web.

```
[contsvc@servera ~]$ curl http://localhost:8080
Hello World
```

9.4. Regrese a la máquina `workstation` como el usuario `student`.

```
[contsvc@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Finalizar

En la máquina `workstation`, ejecute el script `lab finish containers-services` para terminar este ejercicio.

```
[student@workstation ~]$ lab finish containers-services
```

Esto concluye la sección.

► Trabajo de laboratorio

Ejecución de contenedores

En este trabajo de laboratorio, configura un contenedor en su servidor que proporciona un servicio de base de datos MariaDB, almacena su base de datos en un almacenamiento persistente y se inicia automáticamente con el servidor.

Resultados

- Crear contenedores independientes.
- Configurar el redireccionamiento de puertos y el almacenamiento persistente.
- Configurar `systemd` para que los contenedores se inicien cuando se inicia la máquina host.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start containers-review
```

Instrucciones

1. En `serverb`, instale los paquetes de herramientas del contenedor.
2. El registro de imágenes de contenedores en `registry.lab.example.com` almacena la imagen `rhel8/mariadb-103` con varias etiquetas. Use el usuario `podsvc` para enumerar las etiquetas disponibles y observe la etiqueta con el número de versión *más bajo*. Use el usuario `admin` y la contraseña `redhat321` para autenticarse en el registro. Use el archivo `/tmp/registries.conf` como plantilla para la configuración del registro.
3. Con el usuario `podsvc`, cree el directorio `/home/podsvc/db_data`. Configure el directorio para que los contenedores tengan acceso de lectura y escritura.
4. Cree el contenedor independiente `inventorydb`. Use la imagen `rhel8/mariadb-103` del registro `registry.lab.example.com` y especifique la etiqueta con el número de versión más bajo de esa imagen, que encontró en un paso anterior. Asigne el puerto 3306 en el contenedor al puerto 13306 en el host. Monte el directorio `/home/podsvc/db_data` en el host como `/var/lib/mysql/data` en el contenedor. Declare los siguientes valores de variables para el contenedor:

Variable	Valor
MYSQL_USER	operator1
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

Puede copiar y pegar estos parámetros desde el archivo `/home/podsvc/containers-review/variables` en `serverb`. Ejecute el script `/home/podsvc/containers-review/testdb.sh` para confirmar que la base de datos de MariaDB se esté ejecutando.

5. Configure el daemon `systemd` para que el contenedor `inventorydb` se inicie automáticamente cuando arranque el sistema.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade containers-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish containers-review
```

Esto concluye la sección.

► Solución

Ejecución de contenedores

En este trabajo de laboratorio, configura un contenedor en su servidor que proporciona un servicio de base de datos MariaDB, almacena su base de datos en un almacenamiento persistente y se inicia automáticamente con el servidor.

Resultados

- Crear contenedores independientes.
- Configurar el redireccionamiento de puertos y el almacenamiento persistente.
- Configurar `systemd` para que los contenedores se inicien cuando se inicia la máquina host.

Andes De Comenzar

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start containers-review
```

Instrucciones

1. En `serverb`, instale los paquetes de herramientas del contenedor.

- 1.1. Inicie sesión en `serverb` con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Instale el paquete `container-tools`.

```
[student@serverb ~]$ sudo dnf install container-tools
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

2. El registro de imágenes de contenedores en `registry.lab.example.com` almacena la imagen `rhel8/mariadb-103` con varias etiquetas. Use el usuario `podsvc` para enumerar las etiquetas disponibles y observe la etiqueta con el número de versión *más bajo*. Use el usuario `admin` y la contraseña `redhat321` para autenticarse en el registro. Use el archivo `/tmp/registries.conf` como plantilla para la configuración del registro.

- 2.1. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

- 2.2. Inicie sesión en `serverb` con el usuario `podsvc`.

```
[student@workstation ~]$ ssh podsvc@serverb
...output omitted...
[podsvc@serverb ~]$
```

- 2.3. Configure el acceso al registro del aula `registry.lab.example.com` en su directorio de inicio. Use el archivo `/tmp/registries.conf` como plantilla.

```
[podsvc@serverb ~]$ mkdir -p ~/.config/containers/
[podsvc@serverb ~]$ cp /tmp/registries.conf \
~/.config/containers/
```

- 2.4. Inicie sesión en el registro de contenedores con el comando `podman login`.

```
[podsvc@serverb ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 2.5. Vea la información acerca de la imagen `registry.lab.example.com/rhel8/mariadb-103`.

```
[podsvc@serverb ~]$ skopeo inspect \
docker://registry.lab.example.com/rhel8/mariadb-103
{
  "Name": "registry.lab.example.com/rhel8/mariadb-103",
  "Digest": "sha256:a95b...4816",
  "RepoTags": [
    "1-86",
    "1-102",
    "latest"
  ],
  ...output omitted...
```

La etiqueta de versión más baja es la versión 1-86.

3. Con el usuario `podsvc`, cree el directorio `/home/podsvc/db_data`. Configure el directorio para que los contenedores tengan acceso de lectura y escritura.

- 3.1. Cree el directorio `/home/podsvc/db_data`.

```
[podsvc@serverb ~]$ mkdir /home/podsvc/db_data
```


- 3.2. Defina el modo de acceso del directorio en 777 para que todos tengan acceso de lectura y escritura.

```
[podsvc@serverb ~]$ chmod 777 /home/podsvc/db_data
```

4. Cree el contenedor independiente `inventorydb`. Use la imagen `rhel8/mariadb-103` del registro `registry.lab.example.com` y especifique la etiqueta con el número de versión más bajo de esa imagen, que encontró en un paso anterior. Asigne el puerto 3306 en el contenedor al puerto 13306 en el host. Monte el directorio `/home/podsvc/db_data` en el host como `/var/lib/mysql/data` en el contenedor. Declare los siguientes valores de variables para el contenedor:

Variable	Valor
MYSQL_USER	operator1
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

Puede copiar y pegar estos parámetros desde el archivo `/home/podsvc/containers-review/variables` en `serverb`. Ejecute el script `/home/podsvc/containers-review/testdb.sh` para confirmar que la base de datos de MariaDB se esté ejecutando.

- 4.1. Cree el contenedor.

```
[podsvc@serverb ~]$ podman run -d --name inventorydb -p 13306:3306 \
-e MYSQL_USER=operator1 \
-e MYSQL_PASSWORD=redhat \
-e MYSQL_DATABASE=inventory \
-e MYSQL_ROOT_PASSWORD=redhat \
-v /home/podsvc/db_data:/var/lib/mysql/data:Z \
registry.lab.example.com/rhel8/mariadb-103:1-86
...output omitted...
```

- 4.2. Confirme que la base de datos se esté ejecutando.

```
[podsvc@serverb ~]$ ~/containers-review/testdb.sh
Testing the access to the database...
SUCCESS
```

5. Configure el daemon `systemd` para que el contenedor `inventorydb` se inicie automáticamente cuando arranque el sistema.
 - 5.1. Si usó `sudo` o `su` para iniciar sesión con el usuario `podsvc`, salga de `serverb` y use el comando `ssh` para iniciar sesión directamente en `serverb` con el usuario `podsvc`. Recuerde que el daemon `systemd` requiere que el usuario abra una sesión directa desde la consola o a través de SSH. Omita este paso si ya inició sesión en la máquina `serverb` con el usuario `podsvc` mediante SSH.

```
[student@workstation ~]$ ssh podsvc@serverb
...output omitted...
[podsvc@serverb ~]$
```

5.2. Cree el directorio `~/.config/systemd/user/`.

```
[podsvc@serverb ~]$ mkdir -p ~/.config/systemd/user/
```

5.3. Cree el archivo de unidad systemd desde el contenedor en ejecución.

```
[podsvc@serverb ~]$ cd ~/.config/systemd/user/
[podsvc@serverb user]$ podman generate systemd --name inventorydb --files --new
/home/podsvc/.config/systemd/user/container-inventorydb.service
```

5.4. Detenga y, luego, elimine el contenedor `inventorydb`.

```
[podsvc@serverb user]$ podman stop inventorydb
inventorydb
[podsvc@serverb user]$ podman rm inventorydb
0d28f0e0a4118ff019691e34afe09b4d28ee526079b58d19f03b324bd04fd545
```

5.5. Indique al daemon `systemd` que vuelva a cargar su configuración y, luego, habilite e inicie el servicio `container-inventorydb`.

```
[podsvc@serverb user]$ systemctl --user daemon-reload
[podsvc@serverb user]$ systemctl --user enable --now container-inventorydb.service
Created symlink /home/podsvc/.config/systemd/user/multi-user.target.wants/
container-inventorydb.service → /home/podsvc/.config/systemd/user/container-
inventorydb.service.
Created symlink /home/podsvc/.config/systemd/user/default.target.wants/
container-inventorydb.service → /home/podsvc/.config/systemd/user/container-
inventorydb.service.
```

5.6. Confirme que el contenedor se esté ejecutando.

```
[podsvc@serverb user]$ ~/containers-review/testdb.sh
Testing the access to the database...
SUCCESS
[podsvc@serverb user]$ podman ps
```

CONTAINER ID	IMAGE	PORTS	NAMES	COMMAND	CREATED
3ab24e7f000d	registry.lab.example.com/rhel8/mariadb-103:1-86	run-mysqld	47		
	seconds ago	Up 46 seconds ago	0.0.0.0:13306->3306/tcp	inventorydb	

5.7. Ejecute el comando `loginctl enable-linger` para que los servicios de usuario se inicien automáticamente cuando se inicia el servidor.

```
[podsvc@serverb ~]$ loginctl enable-linger
```

5.8. Regrese a la máquina `workstation` como el usuario `student`.

```
[podsvc@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade containers-review
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish containers-review
```

Esto concluye la sección.

Resumen

- Los contenedores proporcionan una manera ligera de distribuir y ejecutar una aplicación con sus dependencias para no entrar en conflicto con el software instalado en el host.
- Los contenedores se ejecutan desde imágenes de contenedores que puede descargar desde un registro de contenedor o que puede crear usted mismo.
- Puede usar archivos de contenedor con instrucciones para crear una imagen de contenedor personalizada.
- Podman, provisto por Red Hat Enterprise Linux, ejecuta y administra directamente los contenedores y las imágenes de contenedores en un único host.
- Los contenedores se pueden ejecutar como `root` o como contenedores `rootless` sin privilegios para una mayor seguridad.
- Puede asignar puertos de red en el host del contenedor para pasar el tráfico a los servicios que se ejecutan en sus contenedores.
- Puede usar variables de entorno para configurar el software en contenedores en el momento de compilación.
- El almacenamiento de contenedores es temporal, pero puede conectar almacenamiento persistente a un contenedor con el contenido de un directorio en el host del contenedor, por ejemplo.
- Puede configurar un archivo de unidad `systemd` para ejecutar automáticamente contenedores cuando se inicia el sistema.

Revisión exhaustiva

Meta

Revisar tareas de *Red Hat System Administration II*

Objetivos

- Revisar tareas de *Red Hat System Administration II*

Secciones

- Revisión exhaustiva

Trabajos de laboratorio

- Corrección de problemas de arranque y mantenimiento de servidores
- Configuración y administración de sistemas de archivos y almacenamiento
- Configuración y administración de seguridad del servidor
- Ejecución de contenedores

Revisión exhaustiva

Objetivos

Demostrar conocimientos y habilidades adquiridos en *Red Hat System Administration II*.

Revisión Red Hat System Administration II

Antes de comenzar la revisión exhaustiva de este curso, debe sentirse cómodo con los temas que se abordaron en cada capítulo.

Puede consultar las secciones anteriores del libro de texto para realizar lecturas complementarias.

Capítulo 1, Mejora de la productividad con la línea de comandos

Ejecutar los comandos de manera más eficiente mediante el uso de las funciones avanzadas de la shell de Bash, los scripts de shell y diversas utilidades proporcionadas por Red Hat Enterprise Linux.

- Ejecutar los comandos de manera más eficiente mediante el uso de las funciones avanzadas de la shell de Bash, los scripts de shell y diversas utilidades proporcionadas por Red Hat Enterprise Linux.
- Ejecutar tareas repetitivas con bucles `for`, evaluar códigos de salida de comandos y scripts, ejecutar pruebas con operadores y crear estructuras condicionales con sentencias `if`.
- Crear expresiones regulares para hacer coincidir datos, aplicar expresiones regulares a archivos de texto con el comando `grep` y usar `grep` para buscar archivos y datos de comandos canalizados.

Capítulo 2, Programación de tareas futuras

Programar tareas para que se ejecuten en una fecha y hora específicas.

- Configurar un comando para que se ejecute una vez en el futuro.
- Programar comandos para que se ejecuten en un horario de repetición usando el archivo `crontab` de un usuario.
- Programar comandos para que se ejecuten en un horario de repetición con el archivo `crontab` del sistema y directorios.
- Habilitar y deshabilitar los temporizadores de `systemd` y configurar un temporizador que administre archivos temporales.

Capítulo 3, Ajuste del rendimiento del sistema

Mejorar el rendimiento del sistema estableciendo parámetros de ajuste y ajustando la prioridad de programación de los procesos.

- Optimizar el rendimiento del sistema seleccionando un perfil de ajuste administrado por el daemon `tuned`.

- Dar o quitar la prioridad a procesos específicos con los comandos `nice` y `renice`.

Capítulo 4, Administración de seguridad de SELinux

Proteger y administrar la seguridad del servidor con SELinux.

- Explicar cómo SELinux protege los recursos, cambiar el modo de SELinux actual de un sistema y definir el modo de SELinux predeterminado de un sistema.
- Administrar las reglas de política de SELinux que determinan el contexto predeterminado para archivos y directorios con el comando `semanage fcontext` y aplicar el contexto definido por la política de SELinux a archivos y directorios con el comando `restorecon`.
- Activar y desactivar las reglas de política de SELinux con el comando `setsebool`, administrar el valor persistente de los booleanos de SELinux con el comando `semanage boolean -l` y consultar las páginas `man` que terminan con `_selinux` para encontrar información útil acerca de los booleanos de SELinux.
- Usar las herramientas de análisis de registros de SELinux y visualizar información útil durante la solución de problemas de SELinux con el comando `sealert`.

Capítulo 5, Administración de almacenamiento básico

Crear y administrar dispositivos de almacenamiento, particiones, sistemas de archivos y espacios de intercambio (swap) desde la línea de comandos.

- Crear particiones de almacenamiento, formatearlas con sistemas de archivos y montarlas para su uso.
- Crear y administrar espacios de intercambio (swap) para complementar la memoria física.

Capítulo 6, Administración de la pila (stack) de almacenamiento

Crear y administrar volúmenes lógicos que contengan sistemas de archivos o espacios de intercambio (swap) desde la línea de comandos.

- Describir los componentes y conceptos del administrador de volúmenes lógicos e implementar el almacenamiento de LVM y mostrar la información de los componentes de LVM.
- Analizar los múltiples componentes de almacenamiento que conforman las capas de la pila (stack) de almacenamiento.

Capítulo 7, Acceso al almacenamiento conectado a la red

Acceder al almacenamiento conectado a la red con el protocolo NFS.

- Identificar la información de exportación de NFS, crear un directorio para usar como punto de montaje, montar una exportación de NFS con el comando `mount` o mediante la configuración del archivo `/etc/fstab` y desmonte una exportación de NFS con el comando `umount`.
- Describir los beneficios de usar el servicio de automontaje y las exportaciones de NFS de automontaje mediante el uso de asignaciones directas e indirectas.

Capítulo 8, Control del proceso de arranque

Administrar el proceso de arranque para controlar los servicios ofrecidos y para solucionar y reparar problemas.

- Describir el proceso de arranque de Red Hat Enterprise Linux, configurar el objetivo predeterminado que se usa en el arranque e iniciar un sistema con un objetivo no predeterminado.
- Iniciar sesión en un sistema y cambiar la contraseña de root cuando la actual se haya perdido.
- Reparar manualmente la configuración del sistema de archivos o problemas de daños que detengan el proceso de arranque.

Capítulo 9, Administración de la seguridad de redes

Controlar las conexiones de red a los servicios con el firewall del sistema y las reglas de SELinux.

- Aceptar o rechazar las conexiones de red a los servicios del sistema con reglas de `firewalld`.
- Verificar que los puertos de red tengan el tipo de SELinux correcto para que los servicios hagan referencia a ellos.

Capítulo 10, Instalación de Red Hat Enterprise Linux

Instalar Red Hat Enterprise Linux en servidores y máquinas virtuales.

- Instalar Red Hat Enterprise Linux en un servidor.
- Explicar los conceptos y la arquitectura de Kickstart, crear un archivo Kickstart con el sitio web Kickstart Generator, modificar un archivo Kickstart existente con un editor de texto y verificar su sintaxis con `ksvalidator`, publicar un archivo Kickstart en el instalador e instalar Kickstart en la red.
- Instalar una máquina virtual en su servidor Red Hat Enterprise Linux con la consola web.

Capítulo 11, Ejecución de contenedores

Obtener, ejecutar y administrar servicios livianos simples como contenedores en un único servidor de Red Hat Enterprise Linux.

- Explicar los conceptos de contenedores y las tecnologías centrales (core) para crear, almacenar y ejecutar contenedores.
- Analizar las herramientas de administración de contenedores para usar registros para almacenar y recuperar imágenes, y para implementar, consultar y acceder a contenedores.
- Proporcionar almacenamiento persistente para los datos del contenedor al compartir el almacenamiento desde el host del contenedor y configurar una red de contenedores.
- Configurar un contenedor como servicio `systemd` y configurar un servicio de contenedor para que se inicie en el momento del arranque.

► Trabajo de laboratorio

Corrección de problemas de arranque y mantenimiento de servidores



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, soluciona y repara los problemas de arranque y actualiza el objetivo predeterminado del sistema. También programa tareas para que se ejecuten en un cronograma recurrente como un usuario normal.

Resultados

- Diagnosticar problemas y recuperar el sistema del modo de emergencia.
- Cambiar el objetivo predeterminado de `graphical.target` a `multi-user.target`.
- Programar trabajos recurrentes para que se ejecuten como un usuario normal.

Andes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview1
```

Especificaciones

- En `workstation`, ejecute el script `/tmp/rhcsa-break1`. Este script causa un problema con el proceso de arranque en `serverb` y luego reinicia la máquina. Solucione la causa y repare el problema de arranque. Cuando se le solicite, use la contraseña `redhat` del usuario `root`.
- En `workstation`, ejecute el script `/tmp/rhcsa-break2`. Este script hace que el objetivo predeterminado cambie del objetivo `multi-user` al objetivo `graphical` en la máquina `serverb` y, luego, reinicie la máquina. En `serverb`, restablezca el objetivo predeterminado para usar el objetivo `multi-user`. La configuración de objetivo predeterminada debe persistir después del reinicio sin intervención manual. Con el usuario `student`, use el comando `sudo` para ejecutar comandos con privilegios. Use `student` como la contraseña cuando se le solicite.

- En `serverb`, programe un trabajo recurrente con el usuario `student` que ejecuta el script `/home/student/backup-home.sh` por hora entre las 7 p. m. y las 9 p. m. todos los días excepto sábados y domingos. Descargue el script de copia de seguridad desde `http://materials.example.com/labs/backup-home.sh`. El script `backup-home.sh` realiza una copia de seguridad del directorio `/home/student` de `serverb` a `servera` en el directorio `/home/student/serverb-backup`. Use el script `backup-home.sh` para programar el trabajo recurrente con el usuario `student`.
- Reinicie la máquina `serverb` y espere a que el arranque finalice antes de calificar.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview1
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview1
```

Esto concluye la sección.

► Solución

Corrección de problemas de arranque y mantenimiento de servidores



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, soluciona y repara los problemas de arranque y actualiza el objetivo predeterminado del sistema. También programa tareas para que se ejecuten en un cronograma recurrente como un usuario normal.

Resultados

- Diagnosticar problemas y recuperar el sistema del modo de emergencia.
- Cambiar el objetivo predeterminado de `graphical.target` a `multi-user.target`.
- Programar trabajos recurrentes para que se ejecuten como un usuario normal.

Andes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview1
```

1. En `workstation`, ejecute el script `/tmp/rhcsa-break1`.

```
[student@workstation ~]$ sh /tmp/rhcsa-break1
```

2. Una vez que la máquina `serverb` se haya iniciado, acceda a la consola y observe que el proceso de arranque se detuvo antes de tiempo. Tómese un momento para especular sobre la posible causa de este comportamiento.
 - 2.1. Localice el icono de la consola de `serverb`, según corresponda para el entorno del aula. Abra la consola e inspeccione el error. El error puede tardar unos segundos en aparecer.

- 2.2. Presione **Ctrl+Alt+Del** para reiniciar la máquina **serverb**. Cuando el menú del cargador de arranque aparezca, presione cualquier tecla excepto **Enter** para interrumpir la cuenta regresiva.
- 2.3. Edite la entrada predeterminada del cargador de arranque en la memoria para iniciar sesión en el modo de emergencia. Presione **e** para editar la entrada actual.
- 2.4. Use las teclas de dirección para navegar hacia la línea que comienza con **linux**. Anexe **systemd.unit=emergency.target**.
- 2.5. Presione **Ctrl+x** para arrancar con la configuración modificada.
- 2.6. Inicie sesión en el modo de emergencia. Use **redhat** como contraseña del usuario **root**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@serverb ~]#
```

3. Vuelva a montar el sistema de archivos / con capacidades de lectura y escritura. Use el comando **mount -a** para intentar montar todos los demás sistemas de archivos.
- 3.1. Vuelva a montar el sistema de archivos / con capacidades de lectura y escritura para editar el sistema de archivos.

```
[root@serverb ~]# mount -o remount,rw /
```

- 3.2. Intente montar todos los demás sistemas de archivos. Observe que uno de los sistemas de archivos no se puede montar.

```
[root@serverb ~]# mount -a
...output omitted...
mount: /FakeMount: can't find UUID=fake.
```

- 3.3. Edite el archivo **/etc/fstab** para corregir el problema. Elimine o comente la línea incorrecta.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=fake      /FakeMount    xfs    defaults    0 0
```

- 3.4. Actualice el daemon **systemd** para que el sistema registre la nueva configuración del archivo **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[ 206.828912] systemd[1]: Reloading.
```

- 3.5. Intente montar todas las entradas para verificar que el archivo **/etc/fstab** ahora sea correcto.

```
[root@serverb ~]# mount -a
```

- 3.6. Reinicie `serverb` y espere a que el arranque finalice. El sistema ahora debería arrancar sin errores.

```
[root@serverb ~]# systemctl reboot
```

4. En `workstation`, ejecute el script `/tmp/rhcsa-break2`. Espere a que la máquina `serverb` complete el reinicio antes de continuar.

```
[student@workstation ~]$ sh /tmp/rhcsa-break2
```

5. En `serverb`, defina el objetivo `multi-user` como destino actual y predeterminado.

- 5.1. Inicie sesión en `serverb` con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 5.2. Determine el objetivo predeterminado.

```
[student@serverb ~]$ systemctl get-default
graphical.target
```

- 5.3. Cambie al objetivo `multi-user`.

```
[student@serverb ~]$ sudo systemctl isolate multi-user.target
[sudo] password for student: student
```

- 5.4. Defina el objetivo `multi-user` como objetivo predeterminado.

```
[student@serverb ~]$ sudo systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/multi-user.target.
```

- 5.5. Reinicie `serverb` y verifique que el objetivo `multi-user` esté configurado como objetivo predeterminado.

```
[student@serverb ~]$ sudo systemctl reboot
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

- 5.6. Después del reinicio del sistema, abra una sesión de SSH en `serverb` con el usuario `student`. Verifique que el objetivo `multi-user` esté configurado como objetivo predeterminado.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ systemctl get-default
multi-user.target
```

6. En `serverb`, programe un trabajo recurrente con el usuario `student` que ejecuta el script `/home/student/backup-home.sh` por hora entre las 7 p. m. y las 9 p. m. todos los días excepto sábados y domingos. Use el script `backup-home.sh` para programar el trabajo recurrente. Descargue el script de copia de seguridad desde `http://materials.example.com/labs/backup-home.sh`.
- 6.1. En `serverb`, descargue el script de copia de seguridad desde `http://materials.example.com/labs/backup-home.sh`. Use `chmod` para hacer que el script de copia de seguridad sea ejecutable.

```
[student@serverb ~]$ wget http://materials.example.com/labs/backup-home.sh
...output omitted...
[student@serverb ~]$ chmod +x backup-home.sh
```

- 6.2. Abra el archivo `crontab` con el editor de textos predeterminado.

```
[student@serverb ~]$ crontab -e
```

- 6.3. Edite el archivo y agregue la siguiente línea:

```
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

Guarde los cambios y salga del editor.

- 6.4. Use el comando `crontab -l` para enumerar los trabajos recurrentes programados.

```
[student@serverb ~]$ crontab -l
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

7. Reinicie `serverb` y espere a que el arranque finalice antes de calificar.

```
[student@serverb ~]$ sudo systemctl reboot
[sudo] password for student: student
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview1
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview1
```

Esto concluye la sección.

► Trabajo de laboratorio

Configuración y administración de sistemas de archivos y almacenamiento



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, creará un volumen lógico, montará un sistema de archivos de red y creará una partición de intercambio (swap) que se activará automáticamente en el arranque. También configure directorios para almacenar archivos temporales.

Resultados

- Crear un volumen lógico.
- Montar un sistema de archivos de red.
- Crear una partición de intercambio (swap) que se active automáticamente en el arranque.
- Configure el directorio para almacenar archivos temporales.

Andes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview2
```

Especificaciones

- En `serverb`, configure un nuevo volumen lógico `vol_home` de 1 GiB en un nuevo grupo de volúmenes `extra_storage` de 2 GiB. Use el disco sin particiones `/dev/vdb` para crear la partición.
- Formatee el volumen lógico `vol_home` con el tipo de sistema de archivos XFS y móntelo en `/user-homes` de manera persistente.
- En `serverb`, monte de forma persistente el sistema de archivos de red `/share` que exporta `servera` en el directorio `/local-share`. La máquina `servera` exporta la ruta `servera.lab.example.com:/share`.

- En `serverb`, cree una nueva partición de intercambio (swap) de 512 MiB en el disco `/dev/vdc`. Monte la partición de intercambio (swap) de manera persistente.
- Cree el grupo de usuarios `production`. Cree los usuarios `production1`, `production2`, `production3` y `production4` con el grupo `production` como grupo adicional.
- En `serverb`, configure el directorio `/run/volatile` para almacenar archivos temporales. Si no se accede a los archivos de este directorio durante más de 30 segundos, el sistema los elimina automáticamente. Defina `0700` como los permisos octales para el directorio. Use el archivo `/etc/tmpfiles.d/volatile.conf` para configurar la eliminación basada en el tiempo de los archivos en el directorio `/run/volatile`.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview2
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview2
```

Esto concluye la sección.

► Solución

Configuración y administración de sistemas de archivos y almacenamiento



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, creará un volumen lógico, montará un sistema de archivos de red y creará una partición de intercambio (swap) que se activará automáticamente en el arranque. También configure directorios para almacenar archivos temporales.

Resultados

- Crear un volumen lógico.
- Montar un sistema de archivos de red.
- Crear una partición de intercambio (swap) que se active automáticamente en el arranque.
- Configure el directorio para almacenar archivos temporales.

Antes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview2
```

1. En `serverb`, configure un nuevo volumen lógico `vol_home` de 1 GiB en un nuevo grupo de volúmenes `extra_storage` de 2 GiB. Use el disco sin particiones `/dev/vdb` para crear la partición.
 - 1.1. Inicie sesión en `serverb` como el usuario `student` y cambie al usuario `root`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 1.2. Cree una partición de 2 GiB en el disco /dev/vdb.

```
[root@serverb ~]# parted /dev/vdb mklabel msdos
...output omitted...
[root@serverb ~]# parted /dev/vdb mkpart primary 1GiB 3GiB
...output omitted...
[root@serverb ~]# parted /dev/vdb set 1 lvm on
...output omitted...
```

- 1.3. Declare el dispositivo de bloque /dev/vdb1 como volumen físico.

```
[root@serverb ~]# pvcreate /dev/vdb1
...output omitted...
```

- 1.4. Cree el grupo de volúmenes extra_storage con la partición /dev/vdb1.

```
[root@serverb ~]# vgcreate extra_storage /dev/vdb1
...output omitted...
```

- 1.5. Cree un volumen lógico vol_home de 1 GiB.

```
[root@serverb ~]# lvcreate -L 1GiB -n vol_home extra_storage
...output omitted...
```

2. Formatee el volumen lógico vol_home con el tipo de sistema de archivos XFS y móntelo en /user-homes de manera persistente.

- 2.1. Cree el directorio /user-homes.

```
[root@serverb ~]# mkdir /user-homes
```

- 2.2. Formatee la partición /dev/extra_storage/vol_home con el tipo de sistema de archivos XFS.

```
[root@serverb ~]# mkfs -t xfs /dev/extra_storage/vol_home
...output omitted...
```

- 2.3. Monte la partición /dev/extra_storage/vol_home de manera persistente en el directorio /user-homes. Use el UUID de la partición para la entrada del archivo /etc/fstab.

```
[root@serverb ~]# lsblk -o UUID /dev/extra_storage/vol_home
UUID
988cf149-0667-4733-abca-f80c6ec50ab6
[root@serverb ~]# echo "UUID=988c...0ab6 /user-homes xfs defaults 0 0" \
>> /etc/fstab
[root@serverb ~]# mount /user-homes
```

3. En `serverb`, monte de forma persistente el sistema de archivos de red `/share` que exporta `servera` en el directorio `/local-share`. La máquina `servera` exporta la ruta `servera.lab.example.com:/share`.

- 3.1. Cree el directorio `/local-share`.

```
[root@serverb ~]# mkdir /local-share
```

- 3.2. Adjunte la entrada correspondiente al archivo `/etc/fstab` para que monte de manera persistente el sistema de archivos de red `servera.lab.example.com:/share`.

```
[root@serverb ~]# echo "servera.lab.example.com:/share /local-share \
nfs rw, sync 0 0" >> /etc/fstab
```

- 3.3. Monte el sistema de archivos de red en el directorio `/local-share`.

```
[root@serverb ~]# mount /local-share
```

4. En `serverb`, cree una partición de intercambio (swap) de 512 MiB en el disco `/dev/vdc`. Active y monte de forma persistente la partición de intercambio (swap).

- 4.1. Cree una partición de 512 MiB en el disco `/dev/vdc`.

```
[root@serverb ~]# parted /dev/vdc mklabel msdos
...output omitted...
[root@serverb ~]# parted /dev/vdc mkpart primary linux-swaps 1MiB 513MiB
...output omitted...
```

- 4.2. Cree el espacio de intercambio (swap) en la partición `/dev/vdc1`.

```
[root@serverb ~]# mkswap /dev/vdc1
...output omitted...
```

- 4.3. Cree una entrada en el archivo `/etc/fstab` para montar de manera persistente el espacio de intercambio (swap). Use el UUID de la partición para crear la entrada del archivo `/etc/fstab`. Active el espacio de intercambio (swap).

```
[root@serverb ~]# lsblk -o UUID /dev/vdc1
UUID
cc18ccb6-bd29-48a5-8554-546bf3471b69
[root@serverb ~]# echo "UUID=cc18...1b69 swap swap defaults 0 0" >> /etc/fstab
[root@serverb ~]# swapon -a
```

5. Cree el grupo de usuarios `production`. A continuación, cree los usuarios `production1`, `production2`, `production3` y `production4` con el grupo `production` como grupo adicional.

```
[root@serverb ~]# groupadd production
[root@serverb ~]# for i in 1 2 3 4; do useradd -G production production$i; done
```

6. En `serverb`, configure el directorio `/run/volatile` para almacenar archivos temporales. Si no se accede a los archivos de este directorio durante más de 30 segundos, el sistema los elimina automáticamente. Defina `0700` como los permisos octales para el directorio. Use el archivo `/etc/tmpfiles.d/volatile.conf` para configurar la eliminación basada en el tiempo de los archivos en el directorio `/run/volatile`.

- 6.1. Cree el archivo `/etc/tmpfiles.d/volatile.conf` con el siguiente contenido:

```
d /run/volatile 0700 root root 30s
```

- 6.2. Use el comando `systemd-tmpfiles --create` para crear el directorio `/run/volatile` si no existe.

```
[root@serverb ~]# systemd-tmpfiles --create /etc/tmpfiles.d/volatile.conf
```

- 6.3. Regrese a la máquina `workstation` como el usuario `student`.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview2
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview2
```

Esto concluye la sección.

► Trabajo de laboratorio

Configuración y administración de seguridad del servidor



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, configurará la autenticación basada en claves de SSH, cambiará la configuración del firewall, ajustará el modo de SELinux y un booleano de SELinux, y solucionará problemas de SELinux.

Resultados

- Configurar la autenticación basada en claves de SSH.
- Configurar parámetros de firewall.
- Ajustar el modo de SELinux y los booleanos de SELinux.
- Solucionar problemas de SELinux.

Antes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview3
```

Especificaciones

- En `serverb`, genere un par de claves SSH para el usuario `student`. No proteja la clave privada con una frase de contraseña.
- Configure el usuario `student` en `servera` para que acepte la autenticación de inicio de sesión con el par de claves de SSH que generó en la máquina `serverb`. El usuario `student` en `serverb` debe ser capaz de iniciar sesión en `servera` con SSH sin ingresar una contraseña.
- En `servera`, verifique los permisos del directorio `/user-homes/production5`. A continuación, configure SELinux para que se ejecute en el modo `permissive` de manera predeterminada.

- En `serverb`, verifique que el directorio `/localhome` no existe. A continuación, configure el directorio de inicio del usuario `production5` para montar el sistema de archivos de red `/user-homes/production5`. La máquina `servera.lab.example.com` exporta el sistema de archivos como el recurso compartido NFS `servera.lab.example.com:/user-homes/production5`. Use el servicio `autofs` para montar el recurso compartido de red. Verifique que el servicio `autofs` cree el directorio `/localhome/production5` con los mismos permisos que en `servera`.
- En `serverb`, ajuste el booleano correspondiente de SELinux para que el usuario `production5` pueda usar el directorio de inicio montado en NFS después de la autenticación basada en claves de SSH. Si se le solicita, use `redhat` como la contraseña para el usuario `production5`.
- En `serverb`, ajuste la configuración del firewall para rechazar todas las solicitudes de conexión de la máquina `servera`. Use la dirección IPv4 `servera (172.25.250.10)` para configurar la regla de firewall.
- En `serverb`, investigue y corrija el problema de falla del servicio web Apache, que escucha en el puerto `30080/TCP` para las conexiones. Ajuste la configuración del firewall según corresponda para que el puerto `30080/TCP` esté abierto para conexiones entrantes.

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview3
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview3
```

Esto concluye la sección.

► Solución

Configuración y administración de seguridad del servidor



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

En esta revisión, configurará la autenticación basada en claves de SSH, cambiará la configuración del firewall, ajustará el modo de SELinux y un booleano de SELinux, y solucionará problemas de SELinux.

Resultados

- Configurar la autenticación basada en claves de SSH.
- Configurar parámetros de firewall.
- Ajustar el modo de SELinux y los booleanos de SELinux.
- Solucionar problemas de SELinux.

Andes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview3
```

1. En `serverb`, genere un par de claves SSH para el usuario `student`. No proteja la clave privada con una frase de contraseña.

- 1.1. Inicie sesión en `serverb` con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
```

- 1.2. Use el comando `ssh-keygen` para generar un par de claves de SSH. No proteja la clave privada con una frase de contraseña.


```
[student@serverb ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): Enter
Created directory '/home/student/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:+ijpGqjEQSGBR80RNchiRTHw/URQksVDHjsHqVBXeYI student@serverb.lab.example.com
The key's randomart image is:
+---[RSA 3072]-----+
|+BBX+O*+o..=+.. |
|+.0.0000 .oE+o . |
|. + . . . . + .0 |
|. . . . . o . o |
|. . . . . S |
|... . |
|.o. .. |
|o .o o |
|. .o... . |
+-----[SHA256]-----+
```

2. Configure el usuario `student` en `servera` para que acepte la autenticación de inicio de sesión con el par de claves de SSH que generó en la máquina `serverb`. El usuario `student` en `serverb` debe ser capaz de iniciar sesión en `servera` con SSH sin ingresar una contraseña.

- 2.1. Envíe la clave pública del par de claves SSH recientemente generado al usuario `student` en la máquina `servera`.

```
[student@serverb ~]$ ssh-copy-id student@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/student/.ssh/
id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be established.
ED25519 key fingerprint is SHA256:shYfoFG0Nnv42pv7j+HG+FISmCAm4Bh5jfjwwSMJbrw.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
student@servera's password: student

Number of key(s) added: 1

Now try logging in to the machine, with: "ssh 'student@servera'"
and check to make sure that only the key(s) you wanted were added.
```

- 2.2. Verifique que el usuario `student` pueda iniciar sesión en `servera` desde `serverb` sin ingresar una contraseña. No cierre la conexión.

```
[student@serverb ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

3. En `servera`, verifique los permisos del directorio `/user-homes/production5`. A continuación, configure SELinux para que se ejecute en el modo `permissive` de manera predeterminada.

3.1. Verifique los permisos del directorio `/user-homes/production5`.

```
[student@servera ~]$ ls -ld /user-homes/production5
drwx-----. 2 production5 production5 62 May  6 05:27 /user-homes/production5
```

3.2. Edite el archivo `/etc/sysconfig/selinux` para establecer el valor del parámetro SELINUX en `permissive`.

```
[student@servera ~]$ sudo vi /etc/sysconfig/selinux
...output omitted...
#SELINUX=enforcing
SELINUX=permissive
...output omitted...
```

3.3. Reinicie el sistema.

```
[student@servera ~]$ sudo systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@serverb ~]$
```

4. En `serverb`, verifique que el directorio `/localhome` no existe. A continuación, configure el directorio de inicio del usuario `production5` para montar el sistema de archivos de red `/user-homes/production5`. La máquina `servera.lab.example.com` exporta el sistema de archivos como el recurso compartido NFS `servera.lab.example.com:/user-homes/production5`. Use el servicio `autofs` para montar el recurso compartido de red. Verifique que el servicio `autofs` cree el directorio `/localhome/production5` con los mismos permisos que en `servera`.

4.1. Verifique que el directorio `/localhome` no existe.

```
[student@serverb ~]$ ls -ld /localhome
ls: cannot access '/localhome': No such file or directory
```

4.2. En `serverb`, cambie al usuario `root`.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

4.3. Instale el paquete `autofs`.

```
[root@serverb ~]# dnf install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
Installed:
  autofs-1:5.1.7-27.el9.x86_64      libsss_autofs-2.6.2-2.el9.x86_64

Complete!
```

- 4.4. Cree el archivo de asignación `/etc/auto.master.d/production5.autofs` con el siguiente contenido:

```
/- /etc/auto.production5
```

- 4.5. Determine al directorio de inicio del usuario `production5`.

```
[root@serverb ~]# getent passwd production5
production5:x:5001:5001::/localhome/production5:/bin/bash
```

- 4.6. Cree el archivo `/etc/auto.production5` con el siguiente contenido:

```
/localhome/production5 -rw servera.lab.example.com:/user-homes/production5
```

- 4.7. Reinicie el servicio `autofs`.

```
[root@serverb ~]# systemctl restart autofs
```

- 4.8. Verifique que el servicio `autofs` cree el directorio `/localhome/production5` con los mismos permisos que en `servera`.

```
[root@serverb ~]# ls -ld /localhome/production5
drwx----- 2 production5 production5 62 May  6 05:52 /localhome/production5
```

5. En `serverb`, ajuste el booleano correspondiente de SELinux para que el usuario `production5` pueda usar el directorio de inicio montado en NFS después de la autenticación basada en claves de SSH. Si se le solicita, use `redhat` como la contraseña para el usuario `production5`.

- 5.1. Abra una nueva ventana de terminal y verifique desde `servera` que el usuario `production5` no pueda iniciar sesión en `serverb` con la autenticación basada en claves SSH. Un booleano de SELinux impide que el usuario inicie sesión. En `workstation`, abra un nuevo terminal e inicie sesión en `servera` con el usuario `student`.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 5.2. Cambie al usuario `production5`. Cuando se le solicite, use la contraseña `redhat` del usuario `production5`.

```
[student@servera ~]$ su - production5
Password: redhat
[production5@servera ~]$
```

5.3. Genere un par de claves SSH.

```
[production5@servera ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/production5/.ssh/id_rsa): Enter
Created directory '/home/production5/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/production5/.ssh/id_rsa.
Your public key has been saved in /home/production5/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:AbUcIBXneyiGIhr4wS1xzs3WqDvbTP+eZuSRn9HQ/cw
production5@servera.lab.example.com
The key's randomart image is:
+---[RSA 3072]-----+
|      ..=++      |
|      . = 0      |
|      . . = . .  |
|.. * + 0 + . . . |
|+ = = B S .. 0 0.|
|. + + + . . + . . E|
|. . . . 0 0 0    |
|      . = . +.0   |
|      000 . = +   |
+-----[SHA256]-----+
```

5.4. Transfiera la clave pública del par de claves SSH al usuario `production5` en la máquina `serverb`. Cuando se le solicite, use la contraseña `redhat` del usuario `production5`.

```
[production5@servera ~]$ ssh-copy-id production5@serverb
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
production5/.ssh/id_rsa.pub"
The authenticity of host 'serverb (172.25.250.11)' can't be established.
ECDSA key fingerprint is SHA256:ciCkaRWF4g6eR9nSdPxQ7KL8czpViXal6BousK544TY.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted
now it is to install the new keys
production5@serverb's password: redhat

Number of key(s) added: 1

Now try logging in to the machine, with:  "ssh 'production5@serverb'"
and check to make sure that only the key(s) you wanted were added.
```

- 5.5. Use la autenticación basada en clave pública SSH en lugar de la autenticación basada en contraseña para iniciar sesión en `serverb` con el usuario `production5`. Este comando debería fallar.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
production5@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-
mic,password).
```

- 5.6. En el terminal que está conectado a `serverb` con el usuario `root`, defina el booleano de SELinux `use_nfs_home_dirs` en `true`.

```
[root@serverb ~]# setsebool -P use_nfs_home_dirs true
```

- 5.7. Vuelva al terminal conectado a `servera` con el usuario `production5` y use la autenticación basada en clave pública SSH en lugar de la autenticación basada en contraseña para iniciar sesión en `serverb` con el usuario `production5`. Este comando debe arrojar resultados satisfactorios.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
...output omitted...
[production5@serverb ~]$
```

- 5.8. Salga y cierre el terminal que está conectado a `serverb` con el usuario `production5`. Mantenga abierto el terminal que está conectado a `serverb` con el usuario `root`.
6. En `serverb`, ajuste la configuración del firewall para rechazar todas las solicitudes de conexión que se originan de la máquina `servera`. Use la dirección IPv4 `servera` (`172.25.250.10`) para configurar la regla de firewall.

- 6.1. Agregue la dirección IPv4 de `servera` a la zona `block`.

```
[root@serverb ~]# firewall-cmd --add-source=172.25.250.10/32 \
--zone=block --permanent
success
```

- 6.2. Vuelva a cargar los cambios en la configuración del firewall.

```
[root@serverb ~]# firewall-cmd --reload
success
```

7. En `serverb`, investigue y corrija el problema de falla del servicio web Apache, que escucha en el puerto `30080/TCP` para las conexiones. Ajuste la configuración del firewall según corresponda para que el puerto `30080/TCP` esté abierto para conexiones entrantes.

- 7.1. Reinicie el servicio `httpd`. Este comando no puede reiniciar el servicio.

```
[root@serverb ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with error code.
See "systemctl status httpd.service" and "journalctl -xeu httpd.service" for
details.
```

- 7.2. Investigue por qué el servicio `httpd` está fallando. Observe el error de permiso que indica que el daemon `httpd` no pudo vincularse al puerto `30080/TCP` en el inicio. Las políticas de SELinux pueden evitar que una aplicación se vincule a un puerto no estándar. Presione `q` para salir del comando.

```
[root@serverb ~]# systemctl status httpd.service
x httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor
   preset: disabled)
   Active: failed (Result: exit-code) since Mon 2022-05-02 13:20:46 EDT; 29s ago
     Docs: man:httpd.service(8)
   Process: 2322 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited,
   status=1/FAILURE)
   Main PID: 2322 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."
       CPU: 30ms

May 02 13:20:46 serverb.lab.example.com systemd[1]: Starting The Apache HTTP
   Server...
May 02 13:20:46 serverb.lab.example.com httpd[2322]: (13)Permission denied:
AH00072: make_sock: could not bind to address [::]:30080
May 02 13:20:46 serverb.lab.example.com httpd[2322]: (13)Permission denied:
AH00072: make_sock: could not bind to address 0.0.0.0:30080
May 02 13:20:46 serverb.lab.example.com httpd[2322]: no listening sockets
available, shutting down
...output omitted...
```

- 7.3. Determine si una política de SELinux está impidiendo que el servicio `httpd` se vincule con el puerto `30080/TCP`. El mensaje de registro revela que el puerto `30080/TCP` no tiene el contexto apropiado de SELinux `http_port_t`, por lo que SELinux impide que el servicio `httpd` se vincule con el puerto. El mensaje de registro también produce la sintaxis del comando `semanage port` para que pueda corregir fácilmente el problema.

```
[root@serverb ~]# sealert -a /var/log/audit/audit.log
...output omitted...
SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port
30080.

**** Plugin bind_ports (92.2 confidence) suggests ****

If you want to allow /usr/sbin/httpd to bind to network port 30080
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 30080
   where PORT_TYPE is one of the following: http_cache_port_t, http_port_t,
   jboss_management_port_t, jboss_messaging_port_t, ntop_port_t, puppet_port_t.
...output omitted...
```

- 7.4. Defina el contexto apropiado de SELinux en el puerto `30080/TCP` para que el servicio `httpd` se vincule con él.

```
[root@serverb ~]# semanage port -a -t http_port_t -p tcp 30080
```

7.5. Reinicie el servicio `httpd`. Este comando debe reiniciar el servicio correctamente.

```
[root@serverb ~]# systemctl restart httpd
```

7.6. Agregue el puerto 30080/TCP a la zona `public` predeterminada.

```
[root@serverb ~]# firewall-cmd --add-port=30080/tcp --permanent
success
[root@serverb ~]# firewall-cmd --reload
success
```

7.7. Regrese a la máquina `workstation` como el usuario `student`.

```
[root@serverb ~]# exit
logout
[student@serverb ~]$ exit
logout
Connection to serverb closed.
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview3
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview3
```

Esto concluye la sección.

► Trabajo de laboratorio

Ejecución de contenedores



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

Resultados

- Crear contenedores rootless independientes.
- Configurar la asignación de puertos y el almacenamiento persistente.
- Configurar `systemd` para que un contenedor pueda administrarlo con los comandos `systemctl`.

Antes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview4
```

Especificaciones

- En `serverb`, configure el usuario `podmgr` con la contraseña `redhat` y configure las herramientas adecuadas para que el usuario `podmgr` administre los contenedores para esta revisión integral. Configure `registry.lab.example.com` como el registro remoto. Use el usuario `admin` y la contraseña `redhat321` para autenticarse. Puede usar el archivo `/tmp/review4/registry.conf` para configurar el registro.
- El directorio `/tmp/review4/container-dev` contiene dos directorios con archivos de desarrollo para los contenedores en esta revisión integral. Copie los dos directorios del directorio `/tmp/review4/container-dev` en el directorio de inicio `podmgr`. Configure el subdirectorio `/home/podmgr/storage/database` para poder usarlo como almacenamiento persistente para un contenedor.
- Cree la red de contenedor habilitada para DNS `production`. Use la subred `10.81.0.0/16` y `10.81.0.1` como puerta de enlace. Use esta red de contenedores para los contenedores que cree en esta revisión integral.

- Cree el contenedor separado `db-app01` basado en la imagen de contenedor `registry.lab.example.com/rhel8/mariadb-103` con el número de etiqueta más bajo en la red `production`. Use el directorio `/home/podmgr/storage/database` como almacenamiento persistente para el directorio `/var/lib/mysql/data` del contenedor `db-app01`. Asigne el puerto 13306 en la máquina local al puerto 3306 en el contenedor. Use los valores de la siguiente tabla para establecer las variables de entorno para crear la base de datos contenerizada.

Variable	Valor
MYSQL_USER	developer
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

- Cree archivos de servicio `systemd` para administrar el contenedor `db-app01`. Configure el servicio `systemd` para que cuando inicie el servicio, el daemon `systemd` mantenga el contenedor original. Inicie y habilite el contenedor como un servicio `systemd`. Configure el contenedor `db-app01` para que se inicien en el arranque del sistema
- Copie el script `/home/podmgr/db-dev/inventory.sql` en el directorio `/tmp` del contenedor `db-app01` y ejecútelo dentro del contenedor. Si ejecutó el script de manera local, debería usar el comando `mysql -u root inventory < /tmp/inventory.sql`.
- Use el archivo de contenedor en el directorio `/home/podmgr/http-dev` para crear el contenedor separado `http-app01` en la red `production`. El nombre de la imagen del contenedor debe ser `http-client` con la etiqueta `9.0`. Asigne el puerto 8080 en la máquina local al puerto 8080 en el contenedor.
- Use el comando `curl` para consultar el contenido del contenedor `http-app01`. Verifique que la salida del comando muestre el nombre del contenedor del cliente y que el estado de la base de datos sea `up` (activo).

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview4
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview4
```

Esto concluye la sección.

► Solución

Ejecución de contenedores



nota

Si planea realizar el examen RHCSA, use el siguiente enfoque para aprovechar al máximo las ventajas de esta revisión integral: intente realizar cada trabajo de laboratorio sin ver los botones de soluciones ni consultar el contenido del curso. Use los scripts de calificación para evaluar su progreso a medida que completa cada trabajo de laboratorio.

Resultados

- Crear contenedores rootless independientes.
- Configurar la asignación de puertos y el almacenamiento persistente.
- Configurar `systemd` para que un contenedor pueda administrarlo con los comandos `systemctl`.

Antes De Comenzar

Si no restableció las máquinas `workstation` y `server` al final del último capítulo, guarde el trabajo que desea mantener de ejercicios anteriores de esas máquinas y restablézcalas ahora.

Con el usuario `student` en la máquina `workstation`, use el comando `lab` para preparar el sistema para este ejercicio.

Este comando prepara su entorno y garantiza que estén disponibles todos los recursos requeridos.

```
[student@workstation ~]$ lab start rhcsa-compreview4
```

1. En `serverb`, configure el usuario `podmgr` con la contraseña `redhat` y configure las herramientas adecuadas para que el usuario `podmgr` administre los contenedores para esta revisión integral. Configure `registry.lab.example.com` como el registro remoto. Use el usuario `admin` y la contraseña `redhat321` para autenticarse. Puede usar el archivo `/tmp/review4/registry.conf` para configurar el registro.

- 1.1. Inicie sesión en `serverb` con el usuario `student`.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Instale el metapaquete `container-tools`.

```
[student@serverb ~]$ sudo dnf install container-tools
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 1.3. Cree el usuario `podmgr` y defina `redhat` como contraseña para el usuario.

```
[student@serverb ~]$ sudo useradd podmgr
[student@serverb ~]$ sudo passwd podmgr
Changing password for user podmgr.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

- 1.4. Salga de la sesión de usuario `student`. Inicie sesión en la máquina `serverb` como el usuario `podmgr`. Si se le solicita, use `redhat` como la contraseña.

```
[student@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$ ssh podmgr@serverb
...output omitted...
[podmgr@serverb ~]$
```

- 1.5. Cree el directorio `~/.config/containers`.

```
[podmgr@serverb ~]$ mkdir -p ~/.config/containers
```

- 1.6. Copie el archivo `/tmp/review4/registries.conf` en el directorio de configuración del contenedor en el directorio de inicio.

```
[podmgr@serverb ~]$ cp /tmp/review4/registries.conf ~/.config/containers/
```

- 1.7. Inicie sesión en el registro y compruebe la configuración.

```
[podmgr@serverb ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

2. El directorio `/tmp/review4/container-dev` contiene dos directorios con archivos de desarrollo para los contenedores en esta revisión integral. Copie los dos directorios del directorio `/tmp/review4/container-dev` en el directorio de inicio `podmgr`. Configure el subdirectorio `/home/podmgr/storage/database` para poder usarlo como almacenamiento persistente para un contenedor.
 - 2.1. Copie el contenido del directorio `/tmp/review4/container-dev` para el directorio de inicio `podmgr`.

```
[podmgr@serverb ~]$ cp -r /tmp/review4/container-dev/* .
[podmgr@serverb ~]$ ls -l
total 0
drwxr-xr-x. 2 podmgr podmgr 27 May 10 21:52 db-dev
drwxr-xr-x. 2 podmgr podmgr 44 May 10 21:52 http-dev
```

- 2.2. Cree el directorio `/home/podmgr/storage/database` en el directorio de inicio `podmgr`. Defina los permisos adecuados en el directorio para que el contenedor lo monte como almacenamiento persistente.

```
[podmgr@serverb ~]$ mkdir -p storage/database
[podmgr@serverb ~]$ chmod 0777 storage/database
[podmgr@serverb ~]$ ls -l storage/
total 0
drwxrwxrwx. 2 podmgr podmgr 6 May 10 21:55 database
```

3. Cree la red de contenedor habilitada para DNS `production`. Use la subred `10.81.0.0/16` y `10.81.0.1` como puerta de enlace. Use esta red de contenedores para los contenedores que cree en esta revisión integral.

- 3.1. Cree la red de contenedor habilitada para DNS `production`. Use la subred `10.81.0.0/16` y `10.81.0.1` como puerta de enlace.

```
[podmgr@serverb ~]$ podman network create --gateway 10.81.0.1 \
--subnet 10.81.0.0/16 production
production
```

- 3.2. Verifique que la función DNS esté habilitada en la red `production`.

```
[podmgr@serverb ~]$ podman network inspect production
[
  {
    "name": "production",
    ...output omitted...
    "subnets": [
      {
        "subnet": "10.81.0.0/16",
        "gateway": "10.81.0.1"
      }
    ],
    ...output omitted...
    "dns_enabled": true,
    ...output omitted...
```

4. Cree el contenedor separado `db-app01` basado en la imagen de contenedor `registry.lab.example.com/rhel8/mariadb-103` con el número de etiqueta más bajo en la red `production`. Use el directorio `/home/podmgr/storage/database` como almacenamiento persistente para el directorio `/var/lib/mysql/data` del contenedor `db-app01`. Asigne el puerto 13306 en la máquina local al puerto 3306 en el contenedor. Use los valores de la siguiente tabla para establecer las variables de entorno para crear la base de datos contenerizada.

Variable	Valor
MYSQL_USER	developer
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

- 4.1. Busque el número de etiqueta de versión más antiguo de la imagen de contenedor `registry.lab.example.com/rhel8/mariadb`.

```
[podmgr@serverb ~]$ skopeo inspect \
docker://registry.lab.example.com/rhel8/mariadb-103
{
  "Name": "registry.lab.example.com/rhel8/mariadb-103",
  "Digest":
    "sha256:a95b678e52bb9f4305cb696e45c91a38c19a7c2c5c360ba6c681b10717394816",
  "RepoTags": [
    "1-86",
    "1-102",
    "latest"
  ]
  ...output omitted...
```

- 4.2. Use el número de etiqueta de versión más antiguo de la salida del paso anterior para crear el contenedor separado `db-app01` en la red `production`. Use el directorio `/home/podmgr/storage/database` como almacenamiento persistente para el contenedor. Asigne el puerto 13306 al puerto del contenedor 3306. Use los datos que se proporcionan en la tabla para definir las variables de entorno para el contenedor.

```
[podmgr@serverb ~]$ podman run -d --name db-app01 \
-e MYSQL_USER=developer \
-e MYSQL_PASSWORD=redhat \
-e MYSQL_DATABASE=inventory \
-e MYSQL_ROOT_PASSWORD=redhat \
--network production -p 13306:3306 \
-v /home/podmgr/storage/database:/var/lib/mysql/data:Z \
registry.lab.example.com/rhel8/mariadb-103:1-86
...output omitted...
ba398d080e00ba1d52b1cf4f5959c477681cce343c11cc7fc39e4ce5f1cf2384
[podmgr@serverb ~]$ podman ps -a
```

CONTAINER ID	IMAGE	STATUS	PORTS	NAMES	COMMAND	CREATED
ba398d080e00	registry.lab.example.com/rhel8/mariadb-103:1-86	Up 20 seconds ago	0.0.0.0:13306->3306/tcp	db-app01	run-mysqld	20 seconds ago

5. Cree archivos de servicio `systemd` para administrar el contenedor `db-app01`. Configure el servicio `systemd` para que cuando inicie el servicio, el daemon `systemd` mantenga el contenedor original. Inicie y habilite el contenedor como un servicio `systemd`. Configure el contenedor `db-app01` para que se inicien en el arranque del sistema

- 5.1. Cree el directorio `~/ .config/systemd/user/` para el archivo de unidad del contenedor.

```
[podmgr@serverb ~]$ mkdir -p ~/.config/systemd/user/
```

- 5.2. Cree el archivo de unidad `systemd` para el contenedor `db-app01` y mueva el archivo de unidad al directorio `~/ .config/systemd/user/`.

```
[podmgr@serverb ~]$ podman generate systemd --name db-app01 --files
/home/podmgr/container-db-app01.service
[podmgr@serverb ~]$ mv container-db-app01.service ~/.config/systemd/user/
```

- 5.3. Detenga el contenedor `db-app01`.

```
[podmgr@serverb ~]$ podman stop db-app01
db-app01
[podmgr@serverb ~]$ podman ps -a
```

CONTAINER ID	IMAGE	STATUS	PORTS	COMMAND NAMES	CREATED
ba398d080e00	registry.lab.example.com/rhel8/mariadb-103:1-86	Exited (0) 3 seconds ago	0.0.0.0:13306->3306/tcp	run-mysqld	About an hour ago

- 5.4. Vuelva a cargar el servicio del usuario `systemd` para usar la nueva unidad de servicio.

```
[podmgr@serverb ~]$ systemctl --user daemon-reload
```

- 5.5. Inicie y habilite la unidad `systemd` para el contenedor `db-app01`.

```
[podmgr@serverb ~]$ systemctl --user enable --now container-db-app01
Created symlink /home/podmgr/.config/systemd/user/default.target.wants/container-db-app01.service → /home/podmgr/.config/systemd/user/container-db-app01.service.
[podmgr@serverb ~]$ systemctl --user status container-db-app01
```

- container-db-app01.service - Podman container-db-app01.service
 - Loaded: loaded (/home/podmgr/.config/systemd/user/container-db-app01.service; disabled; vendor preset: disabled)
 - Active: active (running) since Tue 2022-05-10 22:16:23 EDT; 7s ago
 - ...output omitted...

```
[podmgr@serverb ~]$ podman ps -a
```

CONTAINER ID	IMAGE	STATUS	PORTS	COMMAND NAMES	CREATED
ba398d080e00	registry.lab.example.com/rhel8/mariadb-103:1-86	Up About a minute ago	0.0.0.0:13306->3306/tcp	run-mysqld	59 seconds ago

- 5.6. Use el comando `loginctl` para configurar el contenedor `db-app01` para que se inicie en el arranque del sistema.

```
[podmgr@serverb ~]$ loginctl enable-linger
```

6. Copie el script `/home/podmgr/db-dev/inventory.sql` en el directorio `/tmp` del contenedor `db-app01` y ejecútelo dentro del contenedor. Si ejecutó el script de manera local, debería usar el comando `mysql -u root inventory < /tmp/inventory.sql`.

- 6.1. Copie el script `/home/podmgr/db-dev/inventory.sql` en el directorio `/tmp` del contenedor `db-app01`.

```
[podmgr@serverb ~]$ podman cp /home/podmgr/db-dev/inventory.sql \
db-app01:/tmp/inventory.sql
```

- 6.2. Ejecute el script `inventory.sql` en el contenedor `db-app01`.

```
[podmgr@serverb ~]$ podman exec -it db-app01 sh -c 'mysql -u root inventory
< /tmp/inventory.sql'
```

7. Use el archivo de contenedor en el directorio `/home/podmgr/http-dev` para crear el contenedor separado `http-app01` en la red `production`. El nombre de la imagen del contenedor debe ser `http-client` con la etiqueta `9.0`. Asigne el puerto `8080` en la máquina local al puerto `8080` en el contenedor.

- 7.1. Cree la imagen `http-client:9.0` con el archivo del contenedor en el directorio `/home/podmgr/http-dev`.

```
[podmgr@serverb ~]$ podman build -t http-client:9.0 http-dev/
STEP 1/7: FROM registry.lab.example.com/rhel8/php-74:1-63
...output omitted...
```

- 7.2. Cree el contenedor independiente `http-app01` en la red `production`. Asigne el puerto `8080` en la máquina local al puerto `8080` en el contenedor.

```
[podmgr@serverb ~]$ podman run -d --name http-app01 \
--network production -p 8080:8080 localhost/http-client:9.0
[podmgr@serverb ~]$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED
ba398d080e00	registry.lab.example.com/rhel8/mariadb-103:1-86	run-mysqld	20 minutes ago
ee424df19621	localhost/http-client:9.0	/bin/sh -c	4 seconds ago

8. Consulte el contenido del contenedor `http-app01`. Verifique que muestre el nombre del contenedor del cliente y que el estado de la base de datos sea `up` (activo).

- 8.1. Verifique que el contenedor `http-app01` responda a solicitudes `http`.

```
[podmgr@serverb ~]$ curl 127.0.0.1:8080
This is the server http-app01 and the database is up
```

9. Regrese a la máquina `workstation` como el usuario `student`.

```
[podmgr@serverb ~]$ exit
logout
Connection to serverb closed.
[student@workstation ~]$
```

Evaluación

Inicie sesión con el usuario `student` en la máquina `workstation` y use el comando `lab` para calificar su trabajo. Corrija los errores informados y vuelva a ejecutar el comando hasta obtener un resultado satisfactorio.

```
[student@workstation ~]$ lab grade rhcsa-compreview4
```

Finalizar

En la máquina `workstation`, cambie al directorio de inicio del usuario `student` y use el comando `lab` para completar este ejercicio. Este paso es importante para garantizar que los recursos de ejercicios anteriores no impacten en los siguientes.

```
[student@workstation ~]$ lab finish rhcsa-compreview4
```

Esto concluye la sección.