



Red Hat System Administration II



Red Hat Enterprise Linux 8.2 RH134
Red Hat System Administration II
Ausgabe 120200928
Veröffentlicht 20200928

Autoren: Fiona Allen, Adrian Andrade, Hervé Quatremain, Victor Costea,
Snehangshu Karmakar, Marc Kesler, Ed Parenti, Saumik Paul,
Dallas Spohn
Editor: Steven Bonneville, Philip Sweany, Ralph Rodriguez, David Sacco, Nicole
Muller, Seth Kenlon, Heather Charles

Copyright © 2020 Red Hat, Inc.

The contents of this course and all its modules and related materials, including handouts to audience members, are
Copyright © 2020 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, Hibernate, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered 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 and/or its affiliates.

XFS® is a registered trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

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.

Mitwirkende: Artur Glogowski, Fernando Lozano, Latha Murthy, Samik Sanyal, Chetan Tiwary, Achyut Madhusudan, Rob Locke, Rudolf Kastl, Prashant Rastogi, Heider Souza, Michael Phillips

Dokumentkonventionen	ix
Einführung	xi
Red Hat System Administration II	xi
Informationen zur Kursumgebung	xii
Landessprachliche Anpassung	xvi
1. Steigern der Produktivität in der Befehlszeile	1
Erstellen einfacher Bash-Skripte	2
Angeleitete Übung: Erstellen einfacher Bash-Skripte	6
Effizienteres Ausführen von Befehlen mit Schleifen	9
Angeleitete Übung: Effizienteres Ausführen von Befehlen mit Schleifen	15
Suchen von übereinstimmendem Text in der Befehlsausgabe mit regulären Ausdrücken	17
Angeleitete Übung: Suchen von übereinstimmendem Text in der Befehlsausgabe mit regulären Ausdrücken	26
Praktische Übung: Steigern der Produktivität in der Befehlszeile	29
Zusammenfassung	35
2. Terminieren zukünftiger Tasks	37
Terminieren eines verschobenen Benutzerjobs	38
Angeleitete Übung: Terminieren eines verschobenen Benutzerjobs	40
Terminieren wiederkehrender Benutzerjobs	44
Angeleitete Übung: Terminieren wiederkehrender Benutzerjobs	47
Terminieren wiederkehrender Systemjobs	50
Angeleitete Übung: Terminieren wiederkehrender Systemjobs	54
Verwalten temporärer Dateien	58
Angeleitete Übung: Verwalten temporärer Dateien	62
Quiz: Terminieren zukünftiger Tasks	66
Zusammenfassung	70
3. Tuning der Systemleistung	71
Anpassen von Tuning-Profilen	72
Angeleitete Übung: Anpassen von Tuning-Profilen	77
Beeinflussen der Prozessplanung	80
Angeleitete Übung: Beeinflussen der Prozessplanung	84
Praktische Übung: Tuning der Systemleistung	88
Zusammenfassung	92
4. Steuern des Dateizugriffs mit ACLs	93
Interpretieren von Datei-ACLs	94
Quiz: Interpretieren von Datei-ACLs	102
Sichern von Dateien mit ACLs	105
Angeleitete Übung: Sichern von Dateien mit ACLs	110
Praktische Übung: Steuern des Dateizugriffs mit ACLs	115
Zusammenfassung	124
5. Verwalten der SELinux-Sicherheit	125
Ändern des SELinux-Enforcement-Modus	126
Angeleitete Übung: Ändern des SELinux-Enforcement-Modus	130
Steuern von SELinux-Dateikontexten	133
Angeleitete Übung: Steuern von SELinux-Dateikontexten	137
Anpassen der SELinux-Richtlinie mit booleschen Werten	140
Angeleitete Übung: Anpassen der SELinux-Richtlinie mit booleschen Werten	142
Untersuchen und Beheben von SELinux-Problemen	145
Angeleitete Übung: Untersuchen und Beheben von SELinux-Problemen	150
Praktische Übung: Verwalten der SELinux-Sicherheit	154
Zusammenfassung	160

6. Verwalten von Basisspeicher	161
Hinzufügen von Partitionen, Dateisystemen und dauerhaften Mounts	162
Angeleitete Übung: Hinzufügen von Partitionen, Dateisystemen und dauerhaften Mounts	173
Verwalten des Swap-Speichers	177
Angeleitete Übung: Verwalten des Swap-Speichers	181
Praktische Übung: Verwalten von Basisspeicher	185
Zusammenfassung	194
7. Verwalten logischer Volumes	195
Erstellen logischer Volumes	196
Angeleitete Übung: Erstellen logischer Volumes	204
Erweitern logischer Volumes	209
Angeleitete Übung: Erweitern logischer Volumes	215
Praktische Übung: Verwalten logischer Volumes	219
Zusammenfassung	225
8. Implementieren erweiterter Storage-Features	227
Verwalten von Storage-Schichten mit Stratis	228
Angeleitete Übung: Verwalten von Storage-Schichten mit Stratis	233
Komprimieren und Deduplizieren von Storage mit VDO	239
Angeleitete Übung: Komprimieren und Deduplizieren von Storage mit VDO	242
Praktische Übung: Implementieren erweiterter Storage-Features	246
Zusammenfassung	256
9. Zugreifen auf Network-Attached Storage	257
Mounten von Network-Attached Storage mit NFS	258
Angeleitete Übung: Verwalten des NAS mit NFS	261
Automatisches Mounten von Network-Attached Storage	265
Angeleitete Übung: Automatisches Mounten von Network-Attached Storage	269
Praktische Übung: Zugreifen auf Network-Attached Storage	275
Zusammenfassung	282
10. Steuern des Boot-Vorgangs	283
Auswahl des Boot-Ziels	284
Angeleitete Übung: Auswahl des Boot-Ziels	289
Zurücksetzen des Root-Passworts	292
Angeleitete Übung: Zurücksetzen des Root-Passworts	296
Beheben von Dateisystemproblemen während des Boot-Vorgangs	298
Angeleitete Übung: Beheben von Dateisystemproblemen während des Boot-Vorgangs ..	300
Praktische Übung: Steuern des Boot-Vorgangs	303
Zusammenfassung	309
11. Verwalten der Netzwerksicherheit	311
Verwalten von Server-Firewalls	312
Angeleitete Übung: Verwalten von Server-Firewalls	321
Steuern der SELinux-Portbezeichnung	325
Angeleitete Übung: Steuern der SELinux-Portbezeichnung	328
Praktische Übung: Verwalten der Netzwerksicherheit	332
Zusammenfassung	340
12. Installation von Red Hat Enterprise Linux	341
Installieren von Red Hat Enterprise Linux	342
Angeleitete Übung: Installieren von Red Hat Enterprise Linux	347
Automatisieren der Installation mit Kickstart	350
Angeleitete Übung: Automatisieren der Installation mit Kickstart	359
Installieren und Konfigurieren virtueller Rechner	362
Quiz: Installieren und Konfigurieren virtueller Rechner	367

Praktische Übung: Installation von Red Hat Enterprise Linux	369
Zusammenfassung	375
13. Ausführen von Containern	377
Einführung in Container	379
Quiz: Einführung in Container	384
Ausführen eines einfachen Containers	386
Angeleitete Übung: Ausführen eines einfachen Containers	391
Suchen und Verwalten von Container-Images	394
Angeleitete Übung: Suchen und Verwalten von Container-Images	400
Durchführen des erweiterten Container-Managements	405
Angeleitete Übung: Durchführen des erweiterten Container-Managements	411
Zuordnen von persistentem Storage zu einem Container	417
Angeleitete Übung: Zuordnen von persistentem Storage zu einem Container	419
Verwalten von Containern als Services	423
Angeleitete Übung: Verwalten von Containern als Services	430
Praktische Übung: Ausführen von Containern	436
Zusammenfassung	443
14. Ausführliche Wiederholung	445
Ausführliche Wiederholung	446
Praktische Übung: Beheben von Startproblemen und Warten von Servern	450
Praktische Übung: Konfigurieren und Verwalten von Dateisystemen und Storage	457
Praktische Übung: Konfigurieren und Verwalten der Serversicherheit	465
Praktische Übung: Ausführen von Containern	476

Dokumentkonventionen



Literaturhinweise

„Verweise“ geben an, wo Sie weitere Informationen zu einem Thema in externen Dokumentationen finden können.



Anmerkung

„Hinweise“ sind Tipps, Tastenkombinationen oder alternative Ansätze für die vorliegende Aufgabe. Wenn Sie einen Hinweis ignorieren, hat dies normalerweise keine negativen Konsequenzen. Allerdings können Hinweise helfen, einen Vorgang zu optimieren.



Wichtig

In den Feldern „Wichtig“ werden Details hervorgehoben, die andernfalls leicht übersehen werden könnten: Konfigurationsänderungen, die nur die aktuelle Sitzung betreffen, oder Dienste, die neu gestartet werden müssen, bevor ein Update angewendet werden kann. Wenn Sie ein Feld mit der Bezeichnung „Wichtig“ ignorieren, führt dies nicht zu Datenverlust, kann jedoch Irritationen und Frustration hervorrufen.



Warnung

„Warnungen“ dürfen nicht ignoriert werden. Ignorierte Warnungen führen mit großer Wahrscheinlichkeit zu Datenverlust.

Einführung

Red Hat System Administration II

Dieser Kurs richtet sich insbesondere an Teilnehmer, die Red Hat System Administration I (RH124) erfolgreich abgeschlossen haben. In Red Hat System Administration II (RH134) werden die zentralen Aufgaben behandelt, die ein hauptberuflicher Linux-Administrator beherrschen muss. Diese Kenntnisse werden bei der Prüfung „Red Hat Certified System Administrator“ überprüft. Darüber hinaus wird in diesem Kurs die Enterprise Linux-Administration vertieft. Es werden u. a. Dateisysteme und Partitionierung, logische Volumes, SELinux, Firewallfunktionen und Fehlerbehebung behandelt.

Lerninhalte

- Fähigkeiten, die im Kurs „Red Hat System Administration I (RH124)“ erworben wurden, ausbauen
- Know-how erwerben, das ein RHCSA-zertifizierter Red Hat Enterprise Linux-Systemadministrator benötigt

Zielgruppe

- Dieser Kurs richtet sich insbesondere an Teilnehmer, die Red Hat System Administration I (RH124) erfolgreich abgeschlossen haben. RH134 eignet sich aufgrund seiner Themen nicht als Einstieg in das Red Hat-Schulungsangebot. Teilnehmer, die bisher nicht an einem Red Hat-Kurs teilgenommen haben, sollten System Administration I (RH124) belegen, falls sie noch nie mit Linux gearbeitet haben, oder den Kurs RHCSA Fast Track (RH200), wenn sie bereits Erfahrung mit der Enterprise Linux-Administration haben.

Voraussetzungen

- Erfolgreiche Teilnahme an Red Hat System Administration I (RH124) bzw. vergleichbare Kenntnisse

Informationen zur Kursumgebung

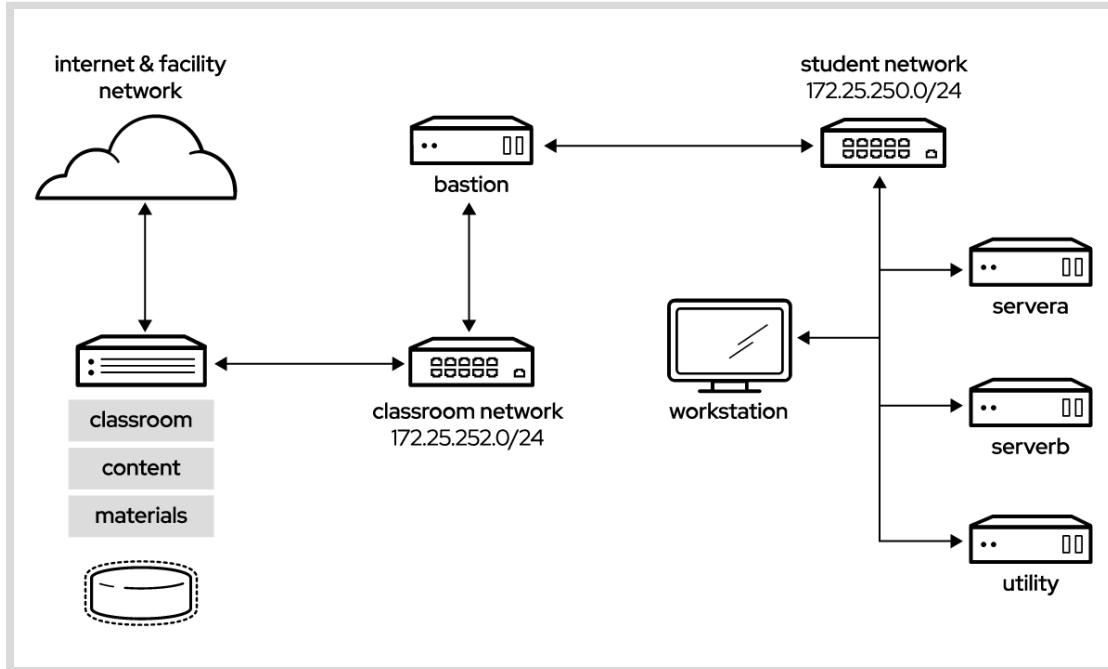


Abbildung 0.1: Kursumgebung

In diesem Kurs wird **workstation** als primäres Computersystem für praktische Übungen verwendet. Für diese Aktivitäten werden von den Teilnehmern zudem zwei weitere Rechner verwendet: **servera** und **serverb**. Alle drei Systeme befinden sich in der DNS Domain **lab.example.com**.

Alle Computersysteme für Teilnehmer verfügen über das standardmäßige Benutzerkonto **student** mit dem Passwort **student**. Das **root**-Passwort für alle Kursteilnehmer-Systeme lautet **redhat**.

Kursraum-Rechner

Rechnername	IP-Adressen	Rolle
bastion.lab.example.com	172.25.250.254	Gateway-System zum Verbinden des privaten Netzwerks des Teilnehmers mit dem Kursraumservern (muss immer ausgeführt werden)
workstation.lab.example.com	172.25.250.9	Grafische Workstation für die Systemadministration
servera.lab.example.com	172.25.250.10	Erster Server
serverb.lab.example.com	172.25.250.11	Zweiter Server

Rechnername	IP-Adressen	Rolle
utility.lab.example.com (registry.lab.example.com)	172.25.250.220	Container-Registry-Server

Die primäre Funktion von **bastion** ist, dass es als Router zwischen dem Kursraumnetzwerk und dem Netzwerk fungiert, das die Rechner der Kursteilnehmer verbindet. Wenn **bastion** außer Betrieb ist, können andere Kursteilnehmer-Rechner nur auf Systeme im individuellen Kursteilnehmer-Netzwerk zugreifen.

Im Kursraum stellen verschiedene Systeme unterstützende Services bereit. Der Host **classroom.example.com** stellt zwei Systeme bereit, **content.example.com** und **materials.example.com**, die als Quellen für Software- und Übungsmaterialien für praktische Übungen dienen. Informationen zur Verwendung dieser Server finden Sie in der Anleitung der jeweiligen Übungen. Der Host **utility.lab.example.com** stellt **registry.lab.example.com** bereit, auf dem ein Red Hat Quay-Container-Registry-Server ausgeführt wird, der Container-Images für Übungen im Kursraum bereitstellt.

Die Systeme **classroom**, **bastion** und **utility** müssen ausgeführt werden, damit die Kursumgebung ordnungsgemäß funktioniert.



Anmerkung

Bei der Anmeldung bei **servera** oder **serverb** wird möglicherweise eine Meldung hinsichtlich der Aktivierung von **cockpit** angezeigt. Die Meldung kann ignoriert werden.

```
[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 ~]$
```

Steuerung Ihrer Systeme

Ihnen werden Remote-Computer in einem Red Hat Online Learning-Kursraum zugewiesen. Der Zugriff darauf erfolgt über eine unter rol.redhat.com [<http://rol.redhat.com>] gehostete Webanwendung. Sie sollten sich mithilfe Ihrer Anmelddaten für das Red Hat Customer Portal auf dieser Website anmelden.

Steuern der virtuellen Rechner

Die virtuellen Rechner in Ihrer Kursumgebung werden über eine Webseite gesteuert. Der Status jedes virtuellen Rechners im Kursraum wird auf der unter der Registerkarte **Online Lab** befindlichen Seite angezeigt.

Rechnerstatus

VM-Status	Beschreibung
STARTING	Der virtuelle Rechner wird hochgefahren.

VM-Status	Beschreibung
STARTED	Der virtuelle Rechner wird ausgeführt und ist verfügbar (oder, falls noch hochgefahren wird, wird es bald sein.)
STOPPING	Der virtuelle Rechner wird heruntergefahren.
STOPPED	Der virtuelle Rechner ist vollständig heruntergefahren. Beim Starten fährt der virtuelle Rechner in denselben Status hoch, in dem er sich vor dem Herunterfahren befand. (Die Disk wurde nicht gelöscht.)
PUBLISHING	Der virtuelle Rechner wird anfänglich erstellt.
WAITING_TO_START	Der virtuelle Rechner wartet auf den Start anderer virtueller Rechner.

In Abhängigkeit des Zustands eines Rechners steht eine Auswahl der folgenden Aktionen zur Verfügung.

Aktionen für Kursumgebung/Rechner

Schaltfläche oder Aktion	Beschreibung
PROVISION LAB	Erstellen Sie den ROL-Kursraum. Hiermit werden sämtliche für die Kursumgebung erforderlichen virtuellen Rechner erstellt und gestartet. Dies dauert ggf. mehrere Minuten.
DELETE LAB	Entfernen Sie den ROL-Kursraum. Hiermit werden alle virtuellen Rechner im Kursraum entfernt. Achtung: Alle auf den Disks gespeicherten Arbeiten gehen verloren.
START LAB	Starten Sie alle virtuellen Rechner im Kursraum.
SHUTDOWN LAB	Halten Sie alle virtuellen Rechner im Kursraum an.
OPEN CONSOLE	Öffnet eine neue Registerkarte im Browser und stellt eine Verbindung zwischen Konsole und virtuellem Rechner her. Sie können sich direkt beim virtuellen Rechner anmelden und Befehle ausführen. In den meisten Fällen sollten Sie sich beim virtuellen Rechner workstation anmelden und ssh verwenden, um mit anderen virtuellen Rechnern eine Verbindung herzustellen.
ACTION → Start	Startet den virtuellen Rechner (d. h. schaltet ihn ein).
ACTION → Shutdown	Fährt den virtuellen Rechner ordnungsgemäß herunter, damit die Disk-Inhalte nicht verloren gehen.
ACTION → Power Off	Erzwingt ein Herunterfahren des virtuellen Rechners und behält die Inhalte seiner Disk bei. Dies entspricht der Stromabschaltung bei einem physischen Rechner.
ACTION → Reset	Erzwingt das Herunterfahren des virtuellen Rechners und setzt die Disk in den Ursprungszustand zurück. Achtung: Sämtliche auf der Disk gespeicherte Arbeit geht verloren.

Einführung

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, einen einzelnen Knoten eines virtuellen Rechners zurückzusetzen, nur für den bestimmten virtuellen Rechner auf **ACTION** → **Reset**.

Klicken Sie zu Beginn einer Übung, sofern Sie angewiesen wurden, alle virtuellen Rechner zurückzusetzen, auf **ACTION** → **Reset**.

Wenn Sie die Kursumgebung auf ihren ursprünglichen Zustand beim Start des Kurses zurücksetzen möchten, können Sie auf **DELETE LAB** klicken, um die gesamte Kursumgebung zu entfernen.

Nach dem Löschen des Labs können Sie auf **PROVISION LAB** klicken, um einen neuen Satz von Kurssystemen bereitzustellen.



Warnung

Der Vorgang **DELETE LAB** kann nicht rückgängig gemacht werden. Die von Ihnen bis zu diesem Zeitpunkt in der Kursumgebung vorgenommene Arbeit geht verloren.

Der Autostop-Timer

Die Registrierung bei Red Hat Online Learning ermöglicht Ihnen eine bestimmte Menge Zeit am Computer. Für den sparsamen Umgang mit der vorgegebenen Computerzeit verfügt der ROL-Kursraum über einen verknüpften Zählvorgang, der die Kursumgebung herunterfährt, wenn der Timer abgelaufen ist.

Klicken Sie zum Anpassen des Timers auf **MODIFY**, damit das Dialogfeld **New Autostop Time** angezeigt wird. Legen Sie die Anzahl der Stunden fest, bis der Kursraum automatisch angehalten wird. Beachten Sie, dass die maximale Dauer zehn Stunden beträgt. Klicken Sie auf **ADJUST TIME**, um diese Änderung auf die Timer-Einstellungen anzuwenden.

Landessprachliche Anpassung

Sprachauswahl auf Benutzerbasis

Ihre Benutzer bevorzugen für ihre Desktop-Umgebung eventuell eine andere Sprache als die Standardsprache des Systems. Für ihr Benutzerkonto möchten sie möglicherweise auch ein anderes Tastaturlayout oder eine andere Eingabemethode verwenden.

Spracheinstellungen

In der GNOME-Desktopumgebung wird der Benutzer möglicherweise bei der ersten Anmeldung aufgefordert, seine bevorzugte Sprache und Eingabemethode einzustellen. Ist dies nicht der Fall, ist die Anwendung Region & Language für den einzelnen Benutzer der einfachste Weg, die bevorzugte Sprache und Eingabemethode anzupassen.

Sie können diese Anwendung auf zwei Arten starten. Sie können den Befehl **gnome-control-center region** über ein Terminal ausführen. Wählen Sie alternativ auf der oberen Leiste im Systemmenü in der rechten Ecke die Schaltfläche für die Einstellungen (das Symbol mit dem gekreuzten Schraubendreher und Schraubenschlüssel) im linken unteren Bereich des Menüs aus.

Wählen Sie in dem sich öffnenden Fenster Region & Language aus. Klicken Sie auf das Feld **Language**, und wählen Sie aus der daraufhin angezeigten Liste die bevorzugte Sprache aus. Dadurch wird auch die Einstellung für **Formats** auf die Standardwerte dieser Sprache aktualisiert. Bei Ihrer nächsten Anmeldung werden die Änderungen vollständig wirksam.

Diese Einstellungen wirken sich auf die GNOME-Desktop-Umgebung sowie auf alle Anwendungen, beispielsweise **gnome-terminal** aus, die innerhalb der Umgebung gestartet werden. Standardmäßig werden sie jedoch nicht auf dieses Benutzerkonto angewendet, wenn über eine **ssh**-Anmeldung von einem Remote-System oder über eine textbasierte Anmeldung von einer virtuellen Konsole (beispielsweise **tty5**) darauf zugegriffen wird.



Anmerkung

Sie können festlegen, dass Ihre Shell-Umgebung dieselbe **LANG**-Einstellung wie Ihre grafische Umgebung verwendet, selbst wenn Sie sich über eine textbasierte virtuelle Konsole oder über **ssh** anmelden. Hierzu kann beispielsweise Code ähnlich dem folgenden in Ihrer `~/.bashrc`-Datei platziert werden. Der Code im folgenden Beispiel stellt die Sprache für eine Textanmeldung so ein, dass sie mit der zurzeit für die GNOME Desktop-Umgebung des Benutzers konfigurierten Sprache übereinstimmt:

```
i=$(grep 'Language=' /var/lib/AccountsService/users/${USER} \
| sed 's/Language=//')
if [ "$i" != "" ]; then
    export LANG=$i
fi
```

Japanisch, Koreanisch, Chinesisch und andere Sprachen mit nicht lateinischem Zeichensatz werden in textbasierten virtuellen Konsolen eventuell nicht einwandfrei angezeigt.

Durch Festlegen der Variablen **LANG** in der Befehlszeile kann für einzelne Befehle die Verwendung einer anderen Sprache festgelegt werden:

```
[user@host ~]$ LANG=fr_FR.utf8 date  
jeu. avril 25 17:55:01 CET 2019
```

Bei den nachfolgenden Befehlen erfolgt die Ausgabe wieder in der Standardsprache des Systems. Der Befehl **locale** kann verwendet werden, um den aktuellen Wert von **LANG** und andere relevante Umgebungsvariablen zu bestimmen.

Einstellungen der Eingabemethode

GNOME 3 in Red Hat Enterprise Linux 7 oder höher verwendet automatisch das Auswahlsystem der Eingabemethode IBus, wodurch sich die Tastaturlayouts und Eingabemethoden schnell wechseln lassen.

Die Anwendung Region & Language kann auch zum Aktivieren alternativer Eingabemethoden verwendet werden. Im Anwendungsfenster von Region & Language zeigt das Feld **Input Sources** an, welche Eingabemethoden derzeit verfügbar sind. Standardmäßig ist eventuell **English (US)** die einzige verfügbare Methode. Markieren Sie **English (US)**, und klicken Sie auf das **Tastatur**-Symbol, um das aktuelle Tastaturlayout anzuzeigen.

Um eine weitere Eingabemethode hinzuzufügen, klicken Sie auf die Schaltfläche **+** im unteren linken Bereich des Fensters **Input Sources**. Das Fenster **Add an Input Source** wird geöffnet. Wählen Sie Ihre Sprache und anschließend die bevorzugte Eingabemethode oder das Tastaturlayout aus.

Wenn mehr als eine Eingabemethode konfiguriert ist, kann der Benutzer rasch zwischen diesen wechseln, indem er **Super+Space** (manchmal auch **Windows+Space**) eingibt. Auf der oberen GNOME-Leiste wird ein **Statusindikator** angezeigt, der über zwei Funktionen verfügt: Er gibt an, welche Ausgabemethode aktiv ist, und fungiert als Menü, das zum Wechseln zwischen Eingabemethoden oder zur Auswahl von erweiterten komplexeren Eingabemethoden verwendet werden kann.

Einige der Methoden sind mit Zahnräder gekennzeichnet. Diese verfügen über erweiterte Konfigurationsoptionen und Funktionen. Beispielsweise kann der Benutzer mit der japanischen Eingabemethode **Japanese (Kana Kanji)** Text im lateinischen Zeichensatz vorab bearbeiten und mit den Tasten **Pfeil nach unten** und **Pfeil nach oben** die zu verwendenden Zeichen auswählen.

Englischsprachige Benutzer in den USA finden dies ggf. ebenfalls hilfreich. Bei der Eingabemethode **English (United States)** lautet beispielsweise das Tastaturlayout **English (international AltGr dead keys)**, das die Taste **AltGr** (oder die rechte **Alt**-Taste) auf einer PC-Tastatur mit 104/105 Tasten als Zusatztaste in Form einer „zweiten Umschalttaste“ sowie als Aktivierungstaste für nicht belegte Tasten zur Eingabe zusätzlicher Zeichen behandelt. Zur Verfügung stehen auch Dvorak und andere Tastaturlayouts.



Anmerkung

In der GNOME Desktop-Umgebung können alle Unicode-Zeichen eingegeben werden, sofern Sie den Unicode-Codepoint oder Unicode-Zahlenwert des Zeichens kennen. Drücken Sie auf **Strg+Umschalt+U** und dann auf den Codepoint. Nach dem Drücken von **Strg+Umschalt+U** erscheint ein unterstrichenes **u**, das angibt, dass das System auf die Eingabe des Unicode-Codepunkts wartet.

Beispielsweise hat der kleine griechische Buchstabe Lambda den Codepoint U +03BB und wird durch Eingabe von **Strg+Umschalt+U**, anschließend **03BB** und Drücken der **Eingabetaste** eingegeben.

Einstellungen der Standardsprache des Systems

Die Standardsprache des Systems ist US-Englisch mit den Unicode-Zeichen der UTF-8-Codierung (**en_US.utf8**). Dies lässt sich aber während oder nach der Installation ändern.

Der **root**-Benutzer kann in der Befehlszeile die systemweiten Ländereinstellungen mit dem Befehl **localectl** ändern. Wenn **localectl** ohne Argumente ausgeführt wird, werden die aktuellen Ländereinstellungen des Systems angezeigt.

Führen Sie zum Einstellen der systemweiten Standardsprache den Befehl **localectl set-locale LANG=locale** aus, wobei *locale* der entsprechende Wert für die **LANG**-Umgebungsvariable aus der Tabelle „Sprachcodereferenz“ in diesem Kapitel ist. Die Änderung wird für die Benutzer bei der nächsten Anmeldung wirksam und wird in der Datei **/etc/locale.conf** gespeichert.

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

In GNOME kann ein Administrator diese Einstellung ändern, indem er in Region & Language in der rechten oberen Ecke des Fensters auf die Schaltfläche **Login Screen** klickt. Das Ändern der **Sprache** des grafischen Anmeldebildschirms wirkt sich auch auf die Einstellungen der Standardsprache des Systems aus, die in der Konfigurationsdatei **/etc/locale.conf** gespeichert sind.



Wichtig

Textbasierte virtuelle Konsolen wie **tty4** sind hinsichtlich der Schriftarten, die sie anzeigen können, eingeschränkter als Terminals in einer virtuellen Konsole, die in einer grafischen Umgebung ausgeführt wird, oder als Pseudoterminals für **ssh**-Sitzungen. Japanische, koreanische und chinesische Zeichen beispielsweise werden in einer textbasierten virtuellen Konsole eventuell nicht richtig angezeigt. Daher sollten Sie in Betracht ziehen, Englisch oder eine andere Sprache mit einem lateinischen Zeichensatz als systemweite Standardeinstellung zu verwenden.

Gleichermaßen sind textbasierte virtuelle Konsolen bei den von ihnen unterstützten Eingabemethoden eingeschränkt. Dies wird separat über die grafische Desktopumgebung verwaltet. Die verfügbaren globalen Eingabeeinstellungen werden sowohl für textbasierte virtuelle Konsolen als auch für die grafische Umgebung anhand von **localectl** konfiguriert. Weitere Informationen finden Sie auf den Manpages **localectl(1)** und **vconsole.conf(5)**.

Sprachpakete

Mit speziellen RPM-Paketen, die als *langpacks* bezeichnet werden, werden Sprachpakete installiert, die Unterstützung für bestimmte Sprachen hinzufügen. Diese Sprachpakete nutzen Abhängigkeiten, um zusätzliche RPM-Pakete, die Lokalisierungen, Verzeichnisse und Übersetzungen für andere Softwarepakete enthalten, automatisch auf Ihrem System zu installieren.

Verwenden Sie **yum list langpacks-*** zum Auflisten der Sprachpakete, die installiert sind und möglicherweise installiert werden:

```
[root@host ~]# yum list langpacks-*
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Installed Packages
langpacks-en.noarch      1.0-12.el8        @AppStream
Available Packages
langpacks-af.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-am.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-ar.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-as.noarch       1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
langpacks-ast.noarch      1.0-12.el8        rhel-8-for-x86_64-appstream-rpms
...output omitted...
```

Um Sprachunterstützung hinzuzufügen, installieren Sie das entsprechende Sprachpaket. Mit dem folgenden Befehl wird beispielsweise Unterstützung für Französisch hinzugefügt:

```
[root@host ~]# yum install langpacks-fr
```

Mit **yum repoquery --whatsonplements** können Sie bestimmen, welche RPM-Pakete durch ein Sprachpaket installiert werden können:

```
[root@host ~]# yum repoquery --whatsonplements langpacks-fr
Updating Subscription Management repositories.
Updating Subscription Management repositories.
Last metadata expiration check: 0:01:33 ago on Wed 06 Feb 2019 10:47:24 AM CST.
glibc-langpack-fr-0:2.28-18.el8.x86_64
gnome-getting-started-docs-fr-0:3.28.2-1.el8.noarch
hunspell-fr-0:6.2-1.el8.noarch
hyphen-fr-0:3.0-1.el8.noarch
libreoffice-langpack-fr-1:6.0.6.1-9.el8.x86_64
man-pages-fr-0:3.70-16.el8.noarch
mythes-fr-0:2.3-10.el8.noarch
```



Wichtig

Sprachpakte nutzen schwache Abhängigkeiten für RPM, damit Zusatzpakte nur installiert werden, wenn das Kernpaket, das sie benötigt, ebenfalls installiert ist.

Wenn beispielsweise *langpacks-fr* wie in den vorherigen Beispielen gezeigt installiert wird, wird das Paket *mythes-fr* nur installiert, wenn auch der Thesaurus *mythes* auf dem System installiert ist.

Wenn *mythes* anschließend auf dem System installiert wird, wird das Paket *mythes-fr* aufgrund der schwachen Abhängigkeit des bereits installierten Pakets *langpacks-fr* ebenfalls automatisch installiert.



Literaturhinweise

Manpages **locale(7)**, **localectl(1)**, **locale.conf(5)**, **vconsole.conf(5)**, **unicode(7)** und **utf-8(7)**

Konvertierungen zwischen den Namen der X11-Layouts der grafischen Desktopumgebung und deren Namen in **localectl** befinden sich in der Datei **/usr/share/X11/xkb/rules/base.lst**.

Sprachcodereferenz



Anmerkung

Diese Tabelle enthält möglicherweise nicht alle auf Ihrem System verfügbaren Sprachpakte. Verwenden Sie **yum info langpacks-SUFFIX**, um weitere Informationen zum jeweiligen Sprachpaket abzurufen.

Sprachcodes

Sprache	Suffix für Sprachpakte	\$LANG-Wert
Englisch (US)	en	en_US.utf8
Assamesisch	as	as_IN.utf8
Bengalisch	bn	bn_IN.utf8
Chinesisch (vereinfacht)	zh_CN	zh_CN.utf8
Chinesisch (traditionell)	zh_TW	zh_TW.utf8
Französisch	fr	fr_FR.utf8
Deutsch	de	de_DE.utf8
Gujarati	gu	gu_IN.utf8
Hindi	hi	hi_IN.utf8

Sprache	Suffix für Sprachpakete	\$LANG-Wert
Italienisch	it	it_IT.utf8
Japanisch	ja	ja_JP.utf8
Kanaresisch	kn	kn_IN.utf8
Koreanisch	ko	ko_KR.utf8
Malayalam	ml	ml_IN.utf8
Marathisch	mr	mr_IN.utf8
Odia	oder	or_IN.utf8
Portugiesisch (Brasilien)	pt_BR	pt_BR.utf8
Punjabi	pa	pa_IN.utf8
Russisch	ru	ru_RU.utf8
Spanisch	es	es_ES.utf8
Tamilisch	ta	ta_IN.utf8
Telugu	te	te_IN.utf8

Kapitel 1

Steigern der Produktivität in der Befehlszeile

Ziel

Führen Sie die Befehle effizienter aus, indem Sie erweiterte Funktionen der Bash-Shell, Shell-Skripte und verschiedene von Red Hat Enterprise Linux bereitgestellte Dienstprogramme verwenden.

Ziele

- Befehlsfolgen durch Schreiben eines einfachen Shell-Skripts automatisieren
- Befehle für Listen von Elementen in einem Skript oder aus der Befehlszeile mit Schleifen und Bedingungen ausführen
- Mit dem Befehl **grep** und regulären Ausdrücken in Protokolldateien und der Befehlsausgabe nach Text suchen, der mit einem Muster übereinstimmt

Abschnitte

- Schreiben einfacher Bash-Skripte (und angeleitete Übung)
- Effizienteres Ausführen von Befehlen mit Schleifen (und angeleitete Übung)
- Suchen von übereinstimmendem Text in der Befehlsausgabe mit regulären Ausdrücken (und angeleitete Übung)

Praktische Übung

Steigern der Produktivität in der Befehlszeile

Erstellen einfacher Bash-Skripte

Ziele

Nachdem Sie diesen Abschnitt abgeschlossen haben, sollten Sie in der Lage sein, Befehlsfolgen zu automatisieren, indem Sie ein einfaches Shell-Skript schreiben.

Erstellen und Ausführen von Bash-Shell-Skripten

Viele einfache, allgemeine Aufgaben der Systemadministration werden mit Befehlszeilertools erledigt. Für komplexere Aufgaben müssen häufig mehrere Befehle, die Ergebnisse untereinander übergeben, verkettet werden. In der Bash-Shell-Umgebung und mit Skriptfunktionen können Linux-Befehle in *Shell-Skripte* kombiniert werden, um sich wiederholende und schwierige reale Probleme zu lösen.

In seiner einfachsten Form ist ein Bash-Shell-Skript eine ausführbare Datei, die eine Liste von Befehlen und möglicherweise eine Programmierlogik zum Steuern von Entscheidungen in der Gesamtaufgabe enthält. Ein gut geschriebenes Shell-Skript ist selbst ein leistungsfähiges Befehlszeilentool und kann von anderen Skripten genutzt werden.

Kenntnisse in der Erstellung von Shell-Skripten sind die Grundlage für eine erfolgreiche Systemadministration in jeder Art von Betriebsumgebung. Praktische Erfahrungen in der Shell-Skripterstellung sind besonders wichtig in Enterprise-Umgebungen, in denen Skripte entscheidend zur Verbesserung von Effizienz und Präzision bei der Erledigung von Routineaufgaben beitragen.

Sie erstellen ein Bash-Shell-Skript, indem Sie eine neue leere Datei in einem Texteditor öffnen. Im Grunde kann jeder beliebige Texteditor verwendet werden, aber erweiterte Editoren wie **vim** oder **emacs** verstehen die Bash-Shell-Syntax und können so **farbcodierte** Hervorhebungen bereitstellen. Diese Hervorhebungen helfen dabei, häufig auftretende Fehler wie falsche Syntax, ungleiche Anzahl an Anführungszeichen, nicht geschlossene runde, geschweifte und eckige Klammern und vieles mehr zu erkennen.

Festlegen des Befehlsinterpreters

Die erste Zeile eines Skripts beginnt mit der Notation '#!', im Allgemeinen als **sh-bang** oder **she-bang** bezeichnet, was von den englischen Namen dieser beiden Zeichen, **sharp** und **bang**, abgeleitet ist. Diese spezifische Notation mit den zwei Byte großen **magischen Zahl** gibt ein interpretierendes Skript an. Die folgende Notation ist der vollständig qualifizierte Dateiname für den richtigen **Befehlsinterpret**, der zur Verarbeitung der Zeilen dieses Skripts benötigt wird. Auf den Manpages **file(1)** und **magic(5)** finden Sie eine Erläuterung dazu, wie in Linux **magische Zahlen** Dateitypen angeben. Bei Skriptdateien in der Bash-Skriptsyntax beginnt die erste Zeile eines Shell-Skripts wie folgt:

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

Ausführen eines Bash-Shell-Skripts

Ein abgeschlossenes Shell-Skript muss ausführbar sein, um als normaler Befehl ausgeführt zu werden. Mit dem Befehl **chmod** fügen Sie die Ausführungsberechtigung hinzu, wobei auch in

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

Verbindung mit dem Befehl **chown** der Dateieigentümer des Skripts geändert werden kann. Erteilen Sie eine Ausführungsberechtigung nur für vorgesehene Benutzer des Skripts.

Wenn Sie das Skript in einem der Verzeichnisse ablegen, das in der Umgebungsvariable **PATH** der Shell aufgeführt ist, dann können Sie das Shell-Skript wie jeden anderen Befehl allein mit dem Dateinamen aufrufen. Die Shell verwendet den ersten Befehl, der mit diesem Dateinamen gefunden wird. Vermeiden Sie die Verwendung vorhandener Befehlsnamen für den Namen der Shell-Skriptdatei. Alternativ können Sie ein Shell-Skript aufrufen, indem Sie in der Befehlszeile einen Pfadnamen für das Skript eingeben. Der Befehl **which**, gefolgt vom Dateinamen des ausführbaren Skripts, zeigt den Pfadnamen des Befehls an, der ausgeführt werden soll.

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

Angeben spezieller Zeichen

Zahlreiche Zeichen und Wörter haben in der Bash-Shell eine spezielle Bedeutung. Gelegentlich müssen Sie diese Zeichen jedoch als buchstäbliche Werte, anstatt ihrer speziellen Bedeutung verwenden. Verwenden Sie dazu eines der drei Tools zum Entfernen (oder *Maskieren*) der besonderen Bedeutung: der umgekehrte Schrägstrich (\), einfache Anführungszeichen ("") oder doppelte Anführungszeichen ("").

Das Escape-Zeichen „Umgekehrter Schrägstrich“ entfernt die spezielle Bedeutung des einzelnen Zeichens, das direkt auf ihn folgt. Um beispielsweise die buchstäbliche Zeichenfolge # **kein Kommentar** mit dem Befehl **echo** anzuzeigen, darf das Zeichen # der Bash(-Shell) nicht mit seiner speziellen Bedeutung interpretiert werden. Fügen Sie den umgekehrten Schrägstrich vor dem Zeichen # ein.

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

Wenn mehrere Zeichen in einer Textzeichenfolge maskiert werden müssen, kann entweder der umgekehrte Schrägstrich mehrfach gesetzt oder einfache Anführungszeichen ("") verwendet werden. Mit einfachen Anführungszeichen wird die ursprüngliche Bedeutung aller Zeichen gesichert, die von diesen Anführungszeichen eingeschlossen sind. Beachten Sie das Escape-Zeichen und die einfachen Anführungszeichen:

```
[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
```

Verwenden Sie doppelte Anführungszeichen zur Unterdrückung von Globbing und von Shell-Erweiterung, ohne gleichzeitig auch Befehls- und Variablensubstitution zu unterdrücken. Das

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

Konzept der Variablensubstitution unterscheidet sich nicht von dem der Befehlssubstitution, es kann jedoch eine optionale Klammersyntax verwendet werden. Beachten Sie die Beispiele für verschiedene Formen der Verwendung von Anführungszeichen.

Verwenden Sie einfache Anführungszeichen zur buchstäblichen Interpretation des gesamten Textes. Zusätzlich zur Unterdrückung von Globbing und Shell-Erweiterung weisen Anführungszeichen die Shell an, Befehls- und Variablensubstitution zu unterdrücken. Das Fragezeichen (?) ist ein **Metazeichen**, das ebenfalls vor Erweiterung geschützt werden muss.

```
[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"
```

Bereitstellen der Ausgabe eines Shell-Skripts

Mit dem Befehl **echo** kann beliebiger Text ausgegeben werden, indem dieser Text als Argument an den Befehl übergeben wird. Standardmäßig wird der Text in der Standardausgabe *standard output (STDOUT)* angezeigt, kann aber mittels Ausgabeumleitung auch an *standard error (STDERR)* umgeleitet werden. In dem folgenden einfachen Bash-Skript wird mit dem Befehl **echo** die Meldung „Hello, world“ an STDOUT angezeigt.

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

echo "Hello, world"

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



Anmerkung

Dieser Benutzer kann **hello** einfach an der Eingabeaufforderung ausführen, weil das Verzeichnis **~/bin** (**/home/user/bin**) in der Variable **PATH** des Benutzers enthalten und das Skript **hello** darin ausführbar ist. Die Shell findet das Skript dort automatisch, sofern keine andere ausführbare Datei mit dem Namen **hello** in einem der vor **/home/user/bin** aufgeführten Verzeichnisse in **PATH** vorhanden ist.

Der Befehl **echo** wird häufig in Shell-Skripten verwendet, um Informationen oder Fehlermeldungen anzuzeigen. Diese Meldungen können ein hilfreicher Indikator für den Verarbeitungsstand eines Skripts sein und können entweder an die Standardausgabe **STDOUT** oder an die Fehlerausgabe **STDERR** weitergegeben oder zum Archivieren in eine Protokolldatei umgeleitet werden. Bei der Anzeige von Fehlermeldungen hat es sich bewährt, dass diese an

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

STDERR übergeben werden, um Fehlermeldungen besser von normalen Statusmeldungen unterscheiden zu können.

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

Der Befehl **echo** kann auch sehr hilfreich beim Debuggen eines problematischen Shell-Skripts sein. Durch Hinzufügen von **echo**-Anweisungen zu dem Teil des Skripts, der sich nicht wie erwartet verhält, werden die ausgeführten Befehle sowie die Werte der aufgerufenen Variablen besser verständlich.



Literaturhinweise

Manpages **bash(1)**, **magic(5)**, **echo(1)** und **echo(1p)**.

► Angeleitete Übung

Erstellen einfacher Bash-Skripte

In dieser Übung schreiben Sie ein einfaches Bash-Skript mit einer Befehlsfolge und führen es von der Befehlszeile aus.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ein einfaches Bash-Skript schreiben und ausführen
- Die Ausgabe eines einfachen Bash-Skripts in eine Datei umleiten

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab console-write start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Das Skript benachrichtigt Sie, wenn der Rechner nicht verfügbar ist. Es installiert zudem das Paket **vim-enhanced**, falls erforderlich.

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

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Erstellen Sie ein einfaches Bash-Skript und führen Sie es aus.

- 2.1. Verwenden Sie den Texteditor **vim**, um eine neue Textdatei in Ihrem Benutzerverzeichnis zu erstellen, und nennen Sie sie **firstscript.sh**.

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

- 2.2. Fügen Sie den folgenden Text ein und speichern Sie die Datei. Beachten Sie, dass die Anzahl der Hash-Zeichen (#) beliebig ist.

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

- 2.3. Führen Sie mit dem Befehl **sh** das Skript aus.

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

- 2.4. Überprüfen Sie die vom Skript generierte Ausgabedatei.

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

#####
```

- 3. Fügen Sie dem Skript **firstscript.sh** weitere Befehle hinzu, führen Sie es aus und überprüfen Sie die Ausgabe.

- 3.1. Bearbeiten Sie die Datei **firstscript.sh** im **vim**-Texteditor.

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

- 3.2. Hängen Sie die folgenden Zeilen in Fettschrift an die Datei **firstscript.sh** an.

```
#!/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
```

- 3.3. Wandeln Sie die Datei **firstscript.sh** mit dem Befehl **chmod** in eine ausführbare Datei um.

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

- 3.4. Führen Sie das **firstscript.sh**-Skript aus.

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

- 3.5. Überprüfen Sie die vom Skript generierte Ausgabedatei.

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

#####
LIST BLOCK DEVICES
```

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

```
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0      11:0    1 1024M  0 rom
vda     252:0    0   10G  0 disk
└─vda1  252:1    0   10G  0 part /
vdb     252:16   0    5G  0 disk

#####
FILESYSTEM FREE SPACE STATUS

Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        892M   0  892M  0% /dev
tmpfs          915M   0  915M  0% /dev/shm
tmpfs          915M   17M 899M  2% /run
tmpfs          915M   0  915M  0% /sys/fs/cgroup
/dev/vda1       10G  1.5G 8.6G 15% /
tmpfs         183M   0  183M  0% /run/user/1000
#####
```

- 4. Entfernen Sie die Übungsdateien und melden Sie sich von **servera** ab.

- 4.1. Löschen Sie die Skriptdatei **firstscript.sh** und die Ausgabedatei **output.txt**.

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

- 4.2. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab console-write finish** aus, um diese Übung zu beenden.

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

Hiermit ist die angeleitete Übung beendet.

Effizienteres Ausführen von Befehlen mit Schleifen

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Listen mit **for**-Schleifen durchlaufen
- Exit-Codes von Befehlen und Skripten bewerten
- Tests mit Operatoren durchführen
- Bedingte Strukturen mit **if**-Anweisungen erstellen

Verwenden von Schleifen zum Durchlaufen von Befehlen

Systemadministratoren haben es bei ihrer täglichen Arbeit häufig mit sich wiederholenden Aktivitäten zu tun. Eine sich wiederholende Aufgabe kann dabei die mehrmalige Ausführung derselben Aktion auf einem Zielgerät bedeuten, wie beispielsweise innerhalb von 10 Minuten das minütliche Prüfen eines Prozesses auf erfolgreiche Fertigstellung. Aufgabenwiederholung kann aber auch die einmalige Ausführung derselben Aktion auf mehreren Zielgeräten bedeuten, wie beispielsweise die Erstellung eines Datenbank-Backups von jeder Datenbank eines Systems. Die **for**-Schleife ist eines der von der Bash bereitgestellten Konstrukte für Shell-Schleifen und kann für Aufgaben-Iterationen verwendet werden.

Verarbeiten von Elementen über die Befehlszeile

Das **for**-Schleifenkonstrukt der Bash hat folgende Syntax.

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

Die Schleife verarbeitet die in *LIST* bereitgestellten Zeichenfolgen einzeln nacheinander. Die Schleifenverarbeitung endet nach Verarbeitung der letzten Zeichenfolge der Liste. Jede Zeichenfolge in der Liste wird während der Verarbeitung des Befehlsblocks der **for**-Schleife temporär als Wert von *VARIABLE* gespeichert. Der Name der Variable ist beliebig. Typischerweise wird der Variablenwert von Befehlen im Befehlsblock referenziert.

Die Liste der in einer **for**-Schleife zu verarbeitenden Zeichenfolgen kann auf verschiedene Weise bereitgestellt werden. Die Zeichenfolgen können einer Liste direkt vom Benutzer hinzugefügt werden oder sie werden von verschiedenen Typen der Shell-Erweiterung generiert wie Variablen-, Klammern- und Dateinamenerweiterung oder Befehlssubstitution. Im Folgenden sind einige Beispiele für die unterschiedlichen Möglichkeiten aufgeführt, mit denen Zeichenfolgen für **for**-Schleifen bereitgestellt werden können.

```
[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*; do ls $FILE; done
filea
fileb
filec
[user@host ~]$ 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" $PACKAGE))"; done
abrt-addon-kerneloops-2.1.11-12.el7.x86_64 was installed on Tue Apr 22 00:09:07
EDT 2014
kernel-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:27:52 EDT 2014
kernel-tools-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:28:01 EDT 2014
kernel-tools-libs-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:26:22 EDT
2014
[user@host ~]$ for EVEN in $(seq 2 2 10); do echo "$EVEN"; done
2
4
6
8
10
```

Verwenden von Exit-Codes in einem Skript

Nachdem ein Skript seinen gesamten Inhalt verarbeitet hat, wird die Steuerung an den aufrufenden Prozess übergeben. Manchmal ist es aber auch wünschenswert, die Ausführung eines Skripts vorher zu beenden, zum Beispiel wenn eine Fehlerbedingung festgestellt wurde. Dies kann durch den Befehl **exit** im Skript erreicht werden. Wenn ein Skript auf einen **exit**-Befehl stößt, wird es sofort beendet und das restliche Skript wird nicht verarbeitet.

Der Befehl **exit** kann mit einem optionalen ganzzahligen Argument zwischen **0** und **255** ausgeführt werden, das den Exit-Code angibt. Ein Exit-Code ist ein Code, der nach Abschluss eines Prozesses zurückgegeben wird. Ein Exit-Code-Wert von **0** zeigt an, dass es keinen Fehler gab. Alle anderen Werte ungleich null geben einen Fehler-Exit-Code an. Sie können Werte ungleich null verwenden, um verschiedene Arten von aufgetretenen Fehlern zu unterscheiden. Dieser Exit-Code wird an den übergeordneten Prozess zurückgegeben, der ihn in der Variable **\$?** speichert und auf den mit **\$?** wie in den folgenden Beispielen zugegriffen werden kann.

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

[user@host bin]$ ./hello
```

```
Hello, world  
[user@host bin]$ echo $?  
0
```

Wenn der Befehl **exit** ohne Argument aufgerufen wird, dann wird das Skript beendet und der Beendigungsstatus des zuletzt ausgeführten Befehls an den übergeordneten Prozess übergeben.

Testen von Skripteingaben

Um zu verhindern, dass Skripte durch unerwartete Bedingungen abgebrochen werden, sollte nicht mit Vermutungen hinsichtlich Eingaben, wie Befehlszeilenargumenten, Benutzereingaben, Befehlssubstitutionen, Variableneinweiterungen und Dateinameneinweiterungen, gearbeitet werden. Mit dem Bash-Befehl **test** kann eine Integritätsprüfung vorgenommen werden.

Wie alle anderen Befehle gibt auch der Befehl **test** nach seiner Ausführung einen Exit-Code zurück, der als der Wert **\$?** gespeichert wird. Die Ausgabe des Testergebnisses erfolgt unmittelbar nach Ausführung des Befehls **test** durch Anzeige des Wertes von **\$?**. Auch hier gibt ein Beendigungsstatuswert von **0** an, dass der Test erfolgreich war, während ein Wert ungleich null auf einen fehlgeschlagenen Test verweist.

Tests können mit einer Vielzahl von Operatoren durchgeführt werden. Mit Operatoren können Sie ermitteln, ob eine Zahl größer als, größer oder gleich, kleiner als, kleiner oder gleich oder gleich einer anderen Zahl ist. Operatoren können verwendet werden, um zu testen, ob eine Textzeichenfolge mit einer anderen Textzeichenfolge identisch ist oder nicht. Mit Operatoren kann auch ausgewertet werden, ob eine Variable einen Wert hat oder nicht.



Anmerkung

Neben den hier vorgestellten Vergleichsoperatoren werden beim Erstellen von Shell-Skripten viele Arten von Operatoren verwendet. Die Manpage für **test(1)** enthält die wichtigen Operatoren für Bedingungsausdrücke mit Beschreibungen.

Auf der Manpage **bash(1)** sind auch Verwendung und Auswertung von Operatoren erläutert, dies ist für Anfänger jedoch sehr schwierig. Es wird empfohlen, dass die Teilnehmer weitergehende Anforderungen an die Erstellung von Shell-Skripten durch Bücher und Kurse zur Shell-Programmierung erlernen.

Die folgenden Beispiele demonstrieren die Verwendung des Befehls **test** mit numerischen Vergleichsoperatoren der Bash.

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

Tests können mit der Testbefehlssyntax der Bash, [**<TESTEXPRESSION>**], durchgeführt werden. Es kann aber auch die neuere erweiterte Testbefehlssyntax [[**<TESTEXPRESSION>**]] verwendet werden, die seit Bash-Version 2.02 verfügbar ist und Funktionen wie Glob-Mustervergleich und Musterabgleich regulärer Ausdrücke bereitstellt.

Die folgenden Beispiele demonstrieren die Verwendung der Testbefehlssyntax sowie der numerischen Vergleichsoperatoren der 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
```

Die folgenden Beispiele demonstrieren die Verwendung der Bash-Vergleichsoperatoren für Zeichenfolgen.

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

Die folgenden Beispiele demonstrieren die Verwendung der unären Zeichenfolgenoperatoren der Bash.

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



Anmerkung

Die Leerzeichen in den Testklammern sind obligatorisch, da sie die Wörter und Elemente im Testausdruck trennen. Die Befehlsanalyseroutine der Shell zerlegt alle Befehlszeilen anhand integrierter Analyseregeln in Wörter und Operatoren, indem Leerzeichen und andere Metazeichen erkannt werden. Auf der Manpage **getopt(3)** finden Sie eine vollständige Beschreibung dieses fortgeschrittenen Konzepts. Die linke eckige Klammer (`[]`) ist selbst ein integrierter Alias für den Befehl **test**. Shell-Wörter, wie Befehle, Sub-Befehle, Optionen, Argumente oder andere Tokenelemente, werden immer durch Leerzeichen begrenzt.

Bedingungsstrukturen

Einfache Shell-Skripte stellen eine Sammlung von Befehlen dar, die vom Anfang bis zum Ende ausgeführt werden. Mit Bedingungsstrukturen können Benutzer Entscheidungen in das Shell-Skript einfügen, sodass bestimmte Teile des Skripts nur dann ausgeführt werden, wenn bestimmte Bedingungen erfüllt sind.

Verwenden des if/then-Konstrukts

Die einfachsten Bedingungsstrukturen in der Bash sind if/then-Konstrukte, die die folgende Syntax aufweisen.

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

Wenn bei diesem Konstrukt eine bestimmte Bedingung erfüllt ist, werden eine oder mehrere Aktionen ausgeführt. Wenn die Bedingung nicht erfüllt ist, wird keine Aktion ausgeführt. Die zuvor gezeigten numerischen, Zeichenfolgen- und Dateitests werden häufig zum Testen der Bedingungen in **if/then**-Anweisungen verwendet. Die Anweisung **fi** am Ende schließt das **if/then**-Konstrukt. Das folgende Codebeispiel zeigt die Verwendung eines **if/then**-Konstrukts zum Starten des **psacct**-Service, falls dieser nicht aktiv ist.

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

Verwenden des if/then/else-Konstrukts

Das **if/then**-Konstrukt kann so erweitert werden, dass unterschiedliche Aktionssätze ausgeführt werden, je nachdem, ob die Bedingung erfüllt ist oder nicht. Dies kann mit dem **if/then/else**-Konstrukt erreicht werden.

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

Das folgende Codebeispiel zeigt die Verwendung einer **if/then/else**-Anweisung zum Starten des **psacct**-Service, falls dieser nicht aktiv ist, und zum Stoppen des Service, falls er aktiv ist.

```
[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
```

Verwenden des **if/then/elif/then/else**-Konstrukts

Schließlich kann das **if/then/else**-Konstrukt so erweitert werden, dass mehr als eine Bedingung geprüft wird und unterschiedliche Aktionssätze ausgeführt werden, je nachdem, ob eine Bedingung erfüllt ist. Das Konstrukt dafür wird im folgenden Beispiel gezeigt:

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

Bei dieser Bedingungsstruktur prüft die Bash die Bedingungen in der angegebenen Reihenfolge. Wenn eine wahre Bedingung gefunden wird, führt die Bash die mit dieser Bedingung verbundenen Aktionen aus und überspringt den Rest der Bedingungsstruktur. Wenn keine der Bedingungen wahr ist, führt die Bash die unter der **else**-Klausel aufgeführten Aktionen aus.

Im folgenden Codebeispiel wird die Verwendung einer **if/then/elif/then/else**-Anweisung demonstriert: Wenn der Service **mariadb** aktiv ist, wird der Client **mysql** ausgeführt. Wenn der Service **postgresql** aktiv ist, wird der Client **psql** ausgeführt. Wenn weder der Service **mariadb** noch der Service **postgresql** aktiv ist, wird der Client **sqlite3** ausgeführt.

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



Literaturhinweise

Manpage (1)**bash**

► Angeleitete Übung

Effizienteres Ausführen von Befehlen mit Schleifen

In dieser Übung verwenden Sie Schleifen, um die Hostnamen mehrerer Server effizient auszugeben.

Ergebnisse

Sie sollten in der Lage sein, eine **for**-Schleife zu erstellen, um eine Liste von Elementen in der Befehlszeile und in einem Shell-Skript zu durchlaufen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab console-commands start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob die Hosts **servera** und **serverb** im Netzwerk erreichbar sind. Das Skript benachrichtigt Sie, wenn die Server nicht verfügbar sind.

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

- 1. Geben Sie mit den Befehlen **ssh** und **hostname** den Hostnamen von **servera** und **serverb** in der Standardausgabe aus.

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

- 2. Erstellen Sie eine **for**-Schleife, um dieselbe Aufgabe effizienter auszuführen.

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

- 3. Erstellen Sie ein Shell-Skript, um dieselbe **for**-Schleife auszuführen.

- 3.1. Erstellen Sie das Verzeichnis **/home/student/bin** zur Aufnahme des Shell-Skripts.

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

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

- 3.2. Überprüfen Sie, ob das neu erstellte Verzeichnis in der Umgebungsvariable **PATH** enthalten ist.

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

- 3.3. Erstellen Sie ein Shell-Skript in **/home/student/bin/printhostname.sh**, um die **for**-Schleife auszuführen. Überprüfen Sie mit dem Befehl **cat** den Inhalt von **printhostname.sh**.

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

- 3.4. Stellen Sie sicher, dass das neu erstellte Skript ausführbar ist.

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

- 3.5. Führen Sie das Skript aus Ihrem Benutzerverzeichnis aus.

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

- 3.6. Überprüfen Sie, ob der Exit-Code Ihres Skripts 0 ist.

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

Beenden

Führen Sie auf **workstation** das Skript **lab console-commands finish** aus, um diese Übung abzuschließen.

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

Hiermit ist die angeleitete Übung beendet.

Suchen von übereinstimmendem Text in der Befehlsausgabe mit regulären Ausdrücken

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Reguläre Ausdrücke erstellen, die den gesuchten Daten entsprechen
- Reguläre Ausdrücke auf Textdateien mit dem Befehl **grep** anwenden
- Dateien und Daten von per Pipe gesendeten Befehlen mit **grep** suchen

Schreiben regulärer Ausdrücke

Reguläre Ausdrücke bieten einen Mechanismus für den Mustervergleich, der das Auffinden bestimmter Inhalte erleichtert. Mit den Befehlen **vim**, **grep** und **less** können reguläre Ausdrücke verwendet werden. Programmiersprachen, wie Perl, Python und C, können reguläre Ausdrücke für mustervergleichende Kriterien verwenden.

Reguläre Ausdrücke sind eine Sprache für sich; das bedeutet, sie haben eine eigene Syntax und eigene Regeln. In diesem Abschnitt wird die Syntax für das Erstellen regulärer Ausdrücke behandelt sowie Beispiele für reguläre Ausdrücke dargestellt.

Beschreibung eines einfachen regulären Ausdrucks

Der einfachste reguläre Ausdruck ist eine genaue Entsprechung. Eine genaue Entsprechung liegt dann vor, wenn die Zeichen im regulären Ausdruck dem Typ und der Anordnung in den gesuchten Daten entsprechen.

Angenommen, ein Benutzer durchsucht die folgende Datei nach allen Vorkommen des Musters **cat**:

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

cat ist die genaue Entsprechung eines **c**, gefolgt von einem **a**, gefolgt von einem **t** ohne andere Zeichen dazwischen. Die Verwendung von **cat** als regulärem Ausdruck für die Suche nach der vorherigen Datei gibt folgende Treffer zurück:

```
cat
concatenate
category
educated
vindication
```

Abgleichen des Anfangs und des Endes einer Zeile

Im vorherigen Abschnitt wurde ein exakt passender regulärer Ausdruck für eine Datei angewendet. Beachten Sie, dass der reguläre Ausdruck der Suchzeichenfolge entsprechen würde, unabhängig davon, an welcher Stelle sie sich befindet: am Anfang, am Ende oder in der Mitte eines Wortes oder einer Zeile. Mit einem **Zeilenanker** können Sie die Position festlegen, an der der reguläre Ausdruck nach einer Übereinstimmung sucht.

Verwenden Sie das Caret-Zeichen (^), um am Anfang einer Zeile zu suchen. Verwenden Sie das Dollarzeichen (\$), um am Ende einer Zeile zu suchen.

Bei Verwendung derselben Datei wie oben würde der reguläre Ausdruck **^cat** zwei Wörtern entsprechen. Der reguläre Ausdruck **\$cat** würde keine übereinstimmenden Wörter finden.

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

Um nur Zeilen in der Datei zu finden, die mit **dog** enden, erstellen Sie mit genau diesem Ausdruck und einem Zeilenende-Anker den regulären Ausdruck **dog\$**. Die Anwendung von **dog\$** auf die Datei liefert zwei Entsprechungen:

```
dog
chilidog
```

Um das einzige Wort in einer Zeile zu finden, verwenden Sie sowohl den Zeilenanfang- als auch den Zeilenende-Anker. Um beispielsweise das Wort **cat** zu finden, wenn es das einzige Wort in einer Zeile ist, verwenden Sie **^cat\$**.

```
cat dog rabbit
cat
horse cat cow
cat pig
```

Hinzufügen von Platzhaltern und Multiplikatoren zu regulären Ausdrücken

In regulären Ausdrücken entspricht ein Punkt (.) einem beliebigen einzelnen Zeichen mit Ausnahme des Zeilenumbruchs. Mit dem regulären Ausdruck **c . t** wird nach einer Zeichenfolge

gesucht, die ein **c**, gefolgt von einem beliebigen einzelnen Zeichen, gefolgt von einem **t** enthält. Beispiele für Übereinstimmungen sind **cat**, **concatenate**, **vindication**, **c5t** und **c\$t**.

Mit einem uneingeschränkten Platzhalter können Sie das Zeichen nicht vorhersagen, das mit dem Platzhalter übereinstimmen würde. Um bestimmte Zeichen zu finden, ersetzen Sie den uneingeschränkten Platzhalter durch zulässige Zeichen. Wenn der reguläre Ausdruck in **c[aou]t** geändert wird, stimmt er mit einem Muster überein, das mit einem **c** beginnt und danach entweder ein **a**, **o** oder **u**, gefolgt von einem **t** steht.

Multiplikatoren sind ein Mechanismus, der oft mit Platzhaltern verwendet wird. Multiplikatoren werden auf das vorherige Zeichen im regulären Ausdruck angewendet. Einer der am häufigsten verwendeten Multiplikatoren ist das Sternchen (*). Bei Verwendung in einem regulären Ausdruck bedeutet dieser Multiplikator, dass null oder mehr des vorherigen Ausdrucks übereinstimmen. Sie können * mit Ausdrücken, nicht nur mit Zeichen verwenden. Zum Beispiel: **c[aou]*t**. Der reguläre Ausdruck **c.*t** stimmt mit **cat**, **coat**, **culvert** und sogar mit **ct** (keine Zeichen zwischen dem **c** und dem **t**) überein. Alle Daten, die mit einem **c** beginnen, dann keine oder mehrere Zeichen enthalten und mit einem **t** enden.

Ein anderer Multiplikatortyp gibt die Anzahl der vorhergehenden, im Muster erwünschten Zeichen an. Ein Beispiel für die Verwendung eines expliziten Multiplikators ist '**c.\{2\}t**'. Dieser reguläre Ausdruck stimmt mit jedem Wort überein, das mit einem **c** beginnt, gefolgt von genau zwei beliebigen Zeichen und mit einem **t** endet. '**c.\{2\}t**' würde im folgenden Beispiel auf zwei Wörter zutreffen:

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



Anmerkung

Es ist empfohlene Praxis, reguläre Ausdrücke in einfache Anführungszeichen zu setzen, da sie häufig Shell-Metazeichen (wie \$, * und {}) enthalten. Dadurch wird sichergestellt, dass die Zeichen vom Befehl und nicht von der Shell interpretiert werden.



Anmerkung

In diesem Kurs wurden zwei unterschiedliche Textanalysesysteme für Metazeichen vorgestellt: **Shell-Mustervergleich** (auch Datei-Globbing oder Dateinamenerweiterung genannt) und **reguläre Ausdrücke**. Weil beide Systeme ähnliche Metazeichen, wie z. B. das Sternchen (*), verwenden, aber über unterschiedliche Interpretationen und Regeln für Metazeichen verfügen, können die beiden Systeme verwirrend sein, bis jedes ausreichend beherrscht wird.

Mustervergleich ist eine Analysetechnik in der Befehlszeile zur einfachen Angabe vieler Dateinamen und wird in erster Linie nur für die Darstellung von Dateinamensmustern in der Befehlszeile unterstützt. Mit regulären Ausdrücken wird jede Form oder jedes Muster in Textzeichenfolgen dargestellt, unabhängig davon, wie komplex diese sind. Reguläre Ausdrücke werden intern durch zahlreiche Textverarbeitungsbefehle unterstützt, wie **grep**, **sed**, **awk**, **python**, **perl** und viele Anwendungen, mit minimalen befehlszeilenabhängigen Variationen der Interpretationsregeln.

Reguläre Ausdrücke

Option	Beschreibung
.	Der Punkt (.) entspricht einem beliebigen einzelnen Zeichen.
?	Das vorhergehende Element ist optional und darf höchstens einmal vorkommen.
*	Das vorhergehende Element muss nie oder mehrmals vorkommen.
+	Das vorhergehende Element muss einmal oder mehrmals vorkommen.
{n}	Das vorhergehende Element muss genau n-mal vorkommen.
{n,}	Das vorhergehende Element muss n-mal oder öfter vorkommen.
{,m}	Das vorhergehende Element darf höchstens m-mal vorkommen.
{n,m}	Das vorhergehende Element muss mindestens n-mal, darf aber höchstens m-mal vorkommen.
[:alnum:]	Alphanumerische Zeichen: '[:alpha:]' und '[:digit:]'; im C-Gebietsschema und in der ASCII-Zeichencodierung ist dies identisch mit '[0-9A-Za-z]'.
[:alpha:]	Alphabetische Zeichen: '[:lower:]' und '[:upper:]'; im C-Gebietsschema und in der ASCII-Zeichencodierung ist dies identisch mit '[A-Za-z]'.
[:blank:]	Leerzeichen: Leerraum und Tabulator
[:cntrl:]	Steuerzeichen In ASCII haben diese Zeichen die Oktalcodes 000 bis 037 und 177 (DEL). In anderen Zeichensätzen sind dies ggf. die entsprechenden Zeichen.
[:digit:]	Ziffern: 0 1 2 3 4 5 6 7 8 9
[:graph:]	Grafische Zeichen: '[:alnum:]' und '[:punct:]'
[:lower:]	Kleinbuchstaben; im C-Gebietsschema und in der ASCII-Zeichencodierung entspricht das 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.
[:print:]	Druckbare Zeichen: '[:alnum:]', '[:punct:]' und Leerzeichen
[:punct:]	Satzzeichen; im C-Gebietsschema und in der ASCII-Zeichencodierung entspricht dies ! " # \$ % & ' () * + , - . / ; < = > ? @ [\] ^ _ ' { } ~. In anderen Zeichensätzen sind dies ggf. die entsprechenden Zeichen.
[:space:]	Leerzeichen: Im C-Gebietsschema sind dies Tabulator, Zeilenumbruch, vertikaler Tabulator, Seitenverschub, Wagenrücklauf und Leerzeichen.
[:upper:]	Großbuchstaben: In 'C'-Gebietsschema und in der ASCII-Zeichencodierung entspricht dies 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.
[:xdigit:]	Hexadezimalziffern: 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.
\b	Übereinstimmung mit leerer Zeichenfolge am Wortanfang oder am Wortende.

Option	Beschreibung
\B	Übereinstimmung mit leerer Zeichenfolge, die nicht am Anfang oder am Ende eines Wortes steht.
\^	Übereinstimmung mit leerer Zeichenfolge am Wortanfang.
\\$	Übereinstimmung mit leerer Zeichenfolge am Wortende.
\w	Übereinstimmung mit Wortbestandteil. Synonym für '[_[:alnum:]]'.
\W	Übereinstimmung mit Nicht-Wortbestandteil. Synonym für '[^_[:alnum:]]'.
\s	Übereinstimmung mit Leerraum. Synonym für '[:space:]'.
\S	Übereinstimmung mit Nicht-Leerraum. Synonym für '[^[:space:]]'.

Abgleichen mit regulären Ausdrücken mit Grep

Der Befehl **grep**, der in der Distribution enthalten ist, verwendet reguläre Ausdrücke, um übereinstimmende Daten zu isolieren.

Isolieren von Daten mit dem Befehl grep

Mit dem Befehl **grep** werden ein regulärer Ausdruck und eine Datei angegeben, in der Übereinstimmungen mit dem regulären Ausdruck gesucht werden sollen.

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



Anmerkung

Es ist empfohlene Praxis, reguläre Ausdrücke in einfache Anführungszeichen zu setzen, da sie häufig Shell-Metazeichen (wie \$, * und {}) enthalten. Dadurch wird sichergestellt, dass die Zeichen von **grep** und nicht von der Shell interpretiert werden.

Der Befehl **grep** kann mit einem Pipe-Operator (|) in Verbindung mit anderen Befehlen verwendet werden. Zum Beispiel:

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

grep-Optionen

Für den Befehl **grep** sind viele nützliche Optionen verfügbar, um anzupassen, wie der Befehl den angegebenen regulären Ausdruck mit Daten verwendet.

Tabelle mit allgemeinen grep-Optionen

Option	Funktion
-i	Den angegebenen regulären Ausdruck verwenden, aber keine Berücksichtigung der Groß-/Kleinschreibung erzwingen (ohne Unterscheidung zwischen Groß-/Kleinschreibung ausführen)
-v	Nur Zeilen anzeigen, die <i>keine</i> Übereinstimmungen mit dem regulären Ausdruck enthalten
-r	Suche nach Daten, die dem regulären Ausdruck entsprechen, rekursiv auf eine Gruppe von Dateien oder Verzeichnissen anwenden
-A NUMBER	ANZAHL (Number) der Zeilen nach der Übereinstimmung mit dem regulären Ausdruck anzeigen
-B NUMBER	ANZAHL (Number) der Zeilen vor der Übereinstimmung mit dem regulären Ausdruck anzeigen
-e	Mit mehreren -e -Optionen können mehrere reguläre Ausdrücke angegeben werden, nach denen dann, verknüpft durch ein logisches ODER, zusammen gesucht wird.

Es gibt viele weitere Optionen für **grep**. Auf der Seite **man** finden Sie die Beschreibungen dieser Optionen.

grep-Beispiele

In den nächsten Beispielen werden verschiedene Konfigurationsdateien und Protokolldateien verwendet.

Reguläre Ausdrücke berücksichtigen standardmäßig die Groß-/Kleinschreibung. Verwenden Sie die Option **-i** mit **grep**, um eine Suche ohne Berücksichtigung der Groß-/Kleinschreibung durchzuführen. Das folgende Beispiel sucht nach dem Muster **serverroot**.

```
[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 specific IP addresses as shown below to
```

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

```
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
Listen 80  
...output omitted...
```

```
[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"
```

Für Fälle, bei denen Sie wissen, nach was Sie *nicht* suchen, ist die Option **-v** sehr nützlich. Die Option **-v** zeigt nur Zeilen an, die nicht mit dem regulären Ausdruck übereinstimmen. Im folgenden Beispiel werden alle Zeilen unabhängig von der Groß-/Kleinschreibung zurückgegeben, die den regulären Ausdruck **server** nicht enthalten.

```
[user@host ~]$ cat /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
172.25.254.254 classroom.example.com classroom  
172.25.254.254 content.example.com content  
172.25.254.254 materials.example.com materials  
172.25.250.254 workstation.lab.example.com workstation  
### rht-vm-hosts file listing the entries to be appended to /etc/hosts  
  
172.25.250.10 servera.lab.example.com servera  
172.25.250.11 serverb.lab.example.com serverb  
172.25.250.254 workstation.lab.example.com workstation
```

```
[user@host ~]$ grep -v -i server /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
172.25.254.254 classroom.example.com classroom  
172.25.254.254 content.example.com content  
172.25.254.254 materials.example.com materials  
172.25.250.254 workstation.lab.example.com workstation  
### rht-vm-hosts file listing the entries to be appended to /etc/hosts  
  
172.25.250.254 workstation.lab.example.com workstation
```

Mit der Option **-v** durchsuchen Sie eine Datei ohne Berücksichtigung von Kommentarzeilen. Im folgenden Beispiel stimmt der reguläre Ausdruck mit allen Zeilen überein, die mit # oder ; beginnen (typische Zeichen, die als Kommentar zu interpretierende Zeilen markieren). Diese Zeilen werden dann in der Ausgabe weggelassen.

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

```
[user@host ~]$ cat /etc/ethertypes
#
# Ethernet frame types
#      This file describes some of the various Ethernet
#      protocol types that are used on Ethernet networks.
#
# This list could be found on:
#          http://www.iana.org/assignments/ethernet-numbers
#          http://www.iana.org/assignments/ieee-802-numbers
#
# <name>    <hexnumber> <alias1>...<alias35> #Comment
#
IPv4        0800     ip ip4      # Internet IP (IPv4)
X25        0805
ARP         0806     ether-arp   #
FR_ARP      0808           # Frame Relay ARP          [RFC1701]
...output omitted...
```

```
[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]
```

Mit dem Befehl **grep** und der Option **-e** können Sie nach mehreren regulären Ausdrücken gleichzeitig suchen. Das folgende Beispiel sucht mit einer Kombination aus **less** und **grep** in der Protokolldatei **/var/log/secure** alle Vorkommen von **pam_unix**, **user root** und **Accepted publickey**.

```
[root@host ~]# cat /var/log/secure | grep -e 'pam_unix' \
-e 'user root' -e 'Accepted publickey' | less
Mar 19 08:04:46 host sshd[6141]: pam_unix(sshd:session): session opened for user
root by (uid=0)
Mar 19 08:04:50 host sshd[6144]: Disconnected from user root 172.25.250.254 port
41170
Mar 19 08:04:50 host sshd[6141]: pam_unix(sshd:session): session closed for user
root
Mar 19 08:04:53 host sshd[6168]: Accepted publickey for student from
172.25.250.254 port 41172 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCT
+wowZLNzNlBT0
```

Um nach Text in einer Datei zu suchen, die mit **vim** oder **less** geöffnet wurde, verwenden Sie den Schrägstrich (/) und geben das zu suchende Muster ein. Drücken Sie die **Eingabetaste**, um die Suche zu starten. Drücken Sie **N**, um nach der nächsten Übereinstimmung zu suchen.

```
[root@host ~]# vim /var/log/boot.log
...output omitted...
[[0;32m OK ^[[0m Reached target Initrd Default Target.^M
Starting dracut pre-pivot and cleanup hook...^M
[[0;32m OK ^[[0m Started dracut pre-pivot and cleanup hook.^M
Starting Cleaning Up and Shutting Down Daemons...^M
Starting Plymouth switch root service...^M
```

```
Starting Setup Virtual Console...^M
[^[[[0;32m OK ^[[[0m] Stopped target Timers.^M
[^[[[0;32m OK ^[[[0m] Stopped dracut pre-pivot and cleanup hook.^M
[^[[[0;32m OK ^[[[0m] Stopped target Initrd Default Target.^M
/Daemons
```

```
[root@host ~]# less /var/log/messages
...output omitted...
Feb 26 15:51:07 host NetworkManager[689]: <info> [1551214267.8584] Loaded device
plugin: NMTeamFactory (/usr/lib64/NetworkManager/1.14.0-14.el8/libnm-device-
plugin-team.so)
Feb 26 15:51:07 host NetworkManager[689]: <info> [1551214267.8599] device (lo):
carrier: link connected
Feb 26 15:51:07 host NetworkManager[689]: <info> [1551214267.8600] manager: (lo):
new Generic device (/org/freedesktop/NetworkManager/Devices/1)
Feb 26 15:51:07 host NetworkManager[689]: <info> [1551214267.8623] manager:
(ens3): new Ethernet device (/org/freedesktop/NetworkManager/Devices/2)
Feb 26 15:51:07 host NetworkManager[689]: <info> [1551214267.8653] device (ens3):
state change: unmanaged -> unavailable (reason 'managed', sys-iface-state:
'external')
/device
```



Literaturhinweise

Manpages **regex(7)** und **grep(1)**

► Angeleitete Übung

Suchen von übereinstimmendem Text in der Befehlsausgabe mit regulären Ausdrücken

In dieser Übung suchen Sie nach Text in den Systemprotokollen und in der Ausgabe von Befehlen, um Informationen effizienter zu finden.

Ergebnisse

Sie sollten in der Lage sein, in Protokolldateien und Konfigurationsdateien effizient nach Text zu suchen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab console-regex start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Es installiert zudem das **postfix**-Paket.

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

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Das **postfix**-Paket wurde heute vom **Start**-Skript installiert. Suchen Sie mit dem Befehl **grep** die GID und die UID für die Gruppen und Benutzer von **postfix** und **postdrop**. Zeigen Sie alle Protokolle mit einer bestimmten **Startzeit** an, um die Ausgabe des Befehls **grep** zu reduzieren.

- 3.1. Ermitteln Sie mit dem Befehl **date** die aktuelle Uhrzeit.

```
[root@servera ~]# date
Fri Mar 22 08:23:56 CET 2019
```

- 3.2. Verwenden Sie den Befehl **grep** mit Datum, Startzeit und GID-Optionen, um die **postfix**- und **postdrop**-GID und -UID des Benutzers zu suchen. Das Übungs-Setup-Skript lief einige Minuten vor der aktuellen Uhrzeit. Berücksichtigen Sie dies bei der Suche nach der Protokolldatei **/var/log/secure**.

```
[root@servera ~]# grep '^Mar 22 08:2.*GID' /var/log/secure
Mar 22 08:20:04 servera groupadd[2514]: group added to /etc/group: name=postdrop,
GID=90
Mar 22 08:20:04 servera groupadd[2514]: new group: name=postdrop, GID=90
Mar 22 08:20:04 servera groupadd[2520]: group added to /etc/group: name=postfix,
GID=89
Mar 22 08:20:04 servera groupadd[2520]: new group: name=postfix, GID=89
Mar 22 08:20:04 servera useradd[2527]: new user: name=postfix, UID=89, GID=89,
home=/var/spool/postfix, shell=/sbin/nologin
```

- 4. Ändern Sie Ihren regulären Ausdruck, um die erste Meldung in der Datei **/var/log/maillog** zu suchen. Beachten Sie, dass bei dieser Suche kein Caret-Zeichen (^) verwendet wird, da Sie nicht nach dem ersten Zeichen in einer Zeile suchen.

```
[root@servera ~]# grep 'postfix' /var/log/maillog | head -n 2
Mar 22 08:21:02 servera postfix/postfix-script[3879]: starting the Postfix mail
system
Mar 22 08:21:02 servera postfix/master[3881]: daemon started -- version 3.3.1,
configuration /etc/postfix
```

- 5. Sie müssen den Namen des **queue**-Verzeichnisses für den **Postfix**-Server suchen. Durchsuchen Sie die Konfigurationsdatei **/etc/postfix/main.cf** nach allen Informationen über Warteschlangen. Verwenden Sie die Option **-i**, um die Groß-/Kleinschreibung zu ignorieren.

```
[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
```

- 6. Überprüfen Sie, ob **postfix** Nachrichten in **/var/log/messages** schreibt. Verwenden Sie den Befehl **less** und dann den Schrägstrich (/), um die Datei zu durchsuchen. Drücken Sie **n**, um zum nächsten Eintrag zu gelangen, der der Suche entspricht. Beenden Sie den Befehl **less** mit der Taste **q**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 22 07:58:04 servera systemd[1]: Started Postfix Mail Transport Agent.
...output omitted...
Mar 22 08:12:26 servera systemd[1]: Stopping Postfix Mail Transport Agent...
Mar 22 08:12:26 servera systemd[1]: Stopped Postfix Mail Transport Agent.
...output omitted...
/Postfix
```

- 7. Überprüfen Sie mit dem Befehl **ps aux**, ob der **postfix**-Server aktuell ausgeführt wird. Reduzieren Sie die Ausgabe von **ps aux**, indem Sie ihn mit dem Befehl **grep** kombinieren.

```
[root@servera ~]# ps aux | grep postfix
root      3881  0.0  0.2 121664  5364 ?          Ss   08:21   0:00 /usr/
libexec/postfix/master -w
postfix    3882  0.0  0.4 147284  9088 ?          S     08:21   0:00 pickup -l -t unix
-u
postfix    3883  0.0  0.4 147336  9124 ?          S     08:21   0:00 qmgr -l -t unix -
u
```

- 8. Überprüfen Sie, ob die Warteschlangen **qmgr**, **cleanup** und **pickup** korrekt konfiguriert sind. Verwenden Sie den Befehl **grep** mit der Option **-e**, um mehrere Einträge in derselben Datei abzuleichen. Die Konfigurationsdatei ist **/etc/postfix/master.cf**.

```
[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
```

- 9. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab console-regex finish** aus, um diese Übung zu beenden.

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

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Steigern der Produktivität in der Befehlszeile

Leistungscheckliste

In dieser Übung erstellen Sie ein Bash-Skript, mit dem relevante Informationen von verschiedenen Hosts gefiltert werden können.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ein Bash-Skript erstellen und die Ausgabe in eine Datei umleiten
- Code mit Schleifen vereinfachen
- Relevanten Inhalt mit **grep** und regulären Ausdrücken filtern

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab console-review start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob die Rechner **workstation**, **servera** und **serverb** im Netzwerk erreichbar sind. Das Skript benachrichtigt Sie, wenn die Server nicht verfügbar sind. Es installiert auch, falls erforderlich, die Pakete *vim-enhanced* und *util-linux*, konfiguriert **sudo** und bereitet den Inhalt von **/var/log/secure** auf **servera** und **serverb** vor.

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

1. Erstellen Sie die Skriptdatei **/home/student/bin/bash-lab** auf **workstation**.
2. Bearbeiten Sie Ihre neu erstellte Skriptdatei so, dass sie den folgenden angeforderten Informationen von den Hosts **servera** und **serverb** entspricht. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

Befehl oder Datei	Angeforderter Inhalt
<code>hostname -f</code>	Die gesamte Ausgabe abrufen
<code>echo "#####"</code>	Die gesamte Ausgabe abrufen
<code>lscpu</code>	Nur die Zeilen abrufen, die mit der Zeichenfolge CPU beginnen
<code>echo "#####"</code>	Die gesamte Ausgabe abrufen
<code>/etc/selinux/config</code>	Leere Zeilen ignorieren. Zeilen, die mit # beginnen, ignorieren
<code>echo "#####"</code>	Die gesamte Ausgabe abrufen
<code>/var/log/secure</code>	Alle Einträge, die „Failed password“ enthalten, abrufen
<code>echo "#####"</code>	Die gesamte Ausgabe abrufen

Speichern Sie die erforderlichen Informationen in den neuen Dateien **/home/student/output-servera** und **/home/student/output-serverb**.



Anmerkung

Sie können **sudo** auf den Hosts **servera** und **serverb** ohne Passwort verwenden. Denken Sie daran, eine Schleife zu verwenden, um Ihr Skript zu vereinfachen. Sie können auch mehrere **grep**-Befehle verwenden, die mit dem Pipe-Zeichen (|) verkettet sind.

- Führen Sie das Skript **/home/student/bin/bash-lab** aus und überprüfen Sie den Inhalt der Ausgabe auf **workstation**.

Bewertung

Führen Sie auf **workstation** den Befehl **lab console-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab console-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Steigern der Produktivität in der Befehlszeile

Leistungscheckliste

In dieser Übung erstellen Sie ein Bash-Skript, mit dem relevante Informationen von verschiedenen Hosts gefiltert werden können.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ein Bash-Skript erstellen und die Ausgabe in eine Datei umleiten
- Code mit Schleifen vereinfachen
- Relevanten Inhalt mit **grep** und regulären Ausdrücken filtern

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab console-review start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob die Rechner **workstation**, **servera** und **serverb** im Netzwerk erreichbar sind. Das Skript benachrichtigt Sie, wenn die Server nicht verfügbar sind. Es installiert auch, falls erforderlich, die Pakete *vim-enhanced* und *util-linux*, konfiguriert **sudo** und bereitet den Inhalt von **/var/log/secure** auf **servera** und **serverb** vor.

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

1. Erstellen Sie die Skriptdatei **/home/student/bin/bash-lab** auf **workstation**.

- 1.1. Erstellen Sie auf **workstation** den Ordner **/home/student/bin/**, falls erforderlich.

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

- 1.2. Erstellen und bearbeiten Sie die Skriptdatei **/home/student/bin/bash-lab** mit **vim**.

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

- 1.3. Fügen Sie den folgenden Text ein und speichern Sie die Datei.

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

- 1.4. Wandeln Sie das Skript in eine ausführbare Datei um.

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

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

2. Bearbeiten Sie Ihre neu erstellte Skriptdatei so, dass sie den folgenden angeforderten Informationen von den Hosts **servera** und **serverb** entspricht. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

Befehl oder Datei	Angeforderter Inhalt
hostname -f	Die gesamte Ausgabe abrufen
echo "#####"	Die gesamte Ausgabe abrufen
lscpu	Nur die Zeilen abrufen, die mit der Zeichenfolge CPU beginnen
echo "#####"	Die gesamte Ausgabe abrufen
/etc/selinux/config	Leere Zeilen ignorieren. Zeilen, die mit # beginnen, ignorieren
echo "#####"	Die gesamte Ausgabe abrufen
/var/log/secure	Alle Einträge, die „Failed password“ enthalten, abrufen
echo "#####"	Die gesamte Ausgabe abrufen

Speichern Sie die erforderlichen Informationen in den neuen Dateien **/home/student/output-servera** und **/home/student/output-serverb**.

**Anmerkung**

Sie können **sudo** auf den Hosts **servera** und **serverb** ohne Passwort verwenden. Denken Sie daran, eine Schleife zu verwenden, um Ihr Skript zu vereinfachen. Sie können auch mehrere **grep**-Befehle verwenden, die mit dem Pipe-Zeichen (|) verkettet sind.

- 2.1. Öffnen und bearbeiten Sie die Skriptdatei **/home/student/bin/bash-lab** mit **vim**.

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

- 2.2. Hängen Sie die folgenden Zeilen in Fettschrift an die Skriptdatei **/home/student/bin/bash-lab** an.

**Anmerkung**

Das folgende Beispiel zeigt, wie Sie das angeforderte Skript erstellen können. Für die Bash-Skripterstellung können Sie verschiedene Ansätze verwenden und dasselbe Ergebnis erzielen.

```
#!/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. Führen Sie das Skript **/home/student/bin/bash-lab** aus und überprüfen Sie den Inhalt der Ausgabe auf **workstation**.

- 3.1. Führen Sie auf **workstation** das Skript **/home/student/bin/bash-lab** aus.

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

- 3.2. Prüfen Sie den Inhalt von **/home/student/output-servera** auf **/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:            21
CPU MHz:              2294.670
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:28 servera sshd[3939]: Failed password for invalid user operator1
from 172.25.250.9 port 58382 ssh2
Mar 21 22:30:31 servera sshd[3951]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 58384 ssh2
Mar 21 22:30:34 servera sshd[3953]: Failed password for invalid user manager1 from
172.25.250.9 port 58386 ssh2
#####
```

```
[student@workstation ~]$ cat /home/student/output-serverb
serverb.lab.example.com
#####
CPU op-mode(s):      32-bit, 64-bit
CPU(s):                2
```

Kapitel 1 | Steigern der Produktivität in der Befehlszeile

```
CPU family:          6
CPU MHz:           2294.664
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:37 serverb sshd[3883]: Failed password for invalid user operator1
from 172.25.250.9 port 39008 ssh2
Mar 21 22:30:39 serverb sshd[3891]: Failed password for invalid user sysadmin1
from 172.25.250.9 port 39010 ssh2
Mar 21 22:30:43 serverb sshd[3893]: Failed password for invalid user manager1 from
172.25.250.9 port 39012 ssh2
#####
```

Bewertung

Führen Sie auf **workstation** den Befehl **lab console-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab console-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Einfache Bash-Skripte erstellen und ausführen
- Schleifen zum Durchlaufen einer Liste von Elementen in der Befehlszeile und in einem Shell-Skript verwenden
- Text in Protokolldateien und Konfigurationsdateien mit regulären Ausdrücken und **grep** suchen

Kapitel 2

Terminieren zukünftiger Tasks

Ziel

Terminieren von Tasks zur automatischen Ausführung in der Zukunft

Ziele

- Einen Befehl einrichten, der zu einem späteren Zeitpunkt einmal ausgeführt wird
- Die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei eines Benutzers terminieren
- Die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei des Systems terminieren
- systemd-Timer aktivieren und deaktivieren sowie einen Timer zum Verwalten temporärer Dateien konfigurieren

Abschnitte

- Terminieren eines verschobenen Benutzerjobs (und angeleitete Übung)
- Terminieren wiederkehrender Benutzerjobs (und angeleitete Übung)
- Terminieren wiederkehrender Systemjobs (und angeleitete Übung)
- Verwalten von temporären Dateien (und angeleitete Übung)

Praktische Übung

Terminieren zukünftiger Tasks

Terminieren eines verschobenen Benutzerjobs

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, einen Befehl einzurichten, der zu einem späteren Zeitpunkt einmal ausgeführt wird.

Beschreibung verschobener Benutzer-Tasks

Manchmal müssen Sie möglicherweise einen Befehl oder eine Befehlsgruppe zu einem bestimmten Zeitpunkt in der Zukunft ausführen. Beispiele: Mitarbeiter, die den Versand einer E-Mail an ihren Chef terminieren möchten, oder ein Systemadministrator, der an einer Firewall-Konfiguration arbeitet und einen „Sicherheitsjob“ einrichtet, durch den die Firewall-Einstellungen alle 10 Minuten zurückgesetzt werden (es sei denn, der Administrator deaktiviert den Job vor diesem Zeitpunkt).

Diese terminierten Befehle werden häufig *Tasks* oder *Jobs* genannt und der Begriff *verschoben* gibt an, dass diese Tasks oder Jobs in der Zukunft ausgeführt werden.

at ist eine der für Red Hat Enterprise Linux-Benutzer verfügbaren Lösungen zum Terminieren verschobener Tasks. Das **at**-Paket stellt den System-Daemon (**atd**) zusammen mit einer Reihe von Befehlszeilertools für die Interaktion mit dem Daemon zur Verfügung (**at**, **atq** und weitere). In einer Standardinstallation von Red Hat Enterprise Linux wird der Daemon **atd** automatisch installiert und aktiviert.

Benutzer (einschließlich **root**) können mit dem Befehl **at** Jobs für den Daemon **atd** in die Warteschlange stellen. Der Daemon **atd** bietet 26 Warteschlangen von **a** bis **z**. Jobs in Warteschlangen, die in der alphabetischen Reihenfolge weiter hinten stehen, erhalten eine niedrigere Systempriorität (höhere *nice*-Werte, dies wird in einem späteren Kapitel behandelt).

Terminieren verschobener Benutzer-Tasks

Verwenden Sie den Befehl **at TIMESPEC**, um einen neuen Job zu terminieren. Der Befehl **at** liest dann die auszuführenden Befehle aus dem **stdin**-Kanal. Bei der manuellen Eingabe von Befehlen können Sie einen Befehl mit **Strg+D** abschließen. Bei komplexeren Befehlen, die für Tippfehler anfällig sind, ist es häufig einfacher, die Eingabeumleitung aus einer Skriptdatei zu verwenden, z. B. **at now +5min < myscript**, anstatt alle Befehle manuell in ein Terminalfenster einzugeben.

Das Argument **TIMESPEC** mit dem Befehl **at** akzeptiert viele leistungsstarke Kombinationen, sodass Benutzer genau beschreiben können, wann ein Job ausgeführt werden soll. Typischerweise beginnen sie mit einer Zeitangabe, z. B. **02:00pm**, **15:59** oder sogar **teatime**, gefolgt von einem optionalen Datum oder einer optionalen Anzahl von Tagen. Im Folgenden finden Sie einige Beispielkombinationen, die verwendet werden können.

- **now+5min**
- **teatime tomorrow** (Teatime ist um **16:00** Uhr)
- **noon +4 days**
- **5pm august 3 2021**

Eine vollständige Liste der gültigen Zeitangaben finden Sie in den Referenzen zur **timespec**-Definition.

Überprüfen und Verwalten verschobener Benutzerjobs

Mit dem Befehl **atq** oder den **at -l**-Befehlen erhalten Sie einen Überblick über die ausstehenden Jobs für den aktuellen Benutzer.

```
[user@host ~]$ atq
❶ 28 ❷ Mon Feb  2 05:13:00 2015 ❸ a ❹ user
29 Mon Feb  3 16:00:00 2014 h user
27 Tue Feb  4 12:00:00 2014 a user
```

In der obigen Ausgabe stellt jede Zeile einen anderen Job dar, dessen Ausführung für die Zukunft terminiert ist.

- ❶ Die eindeutige Jobnummer für diesen Job.
- ❷ Datum und Uhrzeit der Ausführung des terminierten Jobs.
- ❸ Gibt an, dass der Job mit der Standardwarteschlange **a** terminiert ist. Verschiedene Jobs können mit unterschiedlichen Warteschlangen terminiert werden.
- ❹ Der Eigentümer des Jobs (und der Benutzer, als der der Job ausgeführt werden soll).



Wichtig

Unprivilegierte Benutzer können nur ihre eigenen Jobs anzeigen und verwalten. Der Benutzer **root** kann alle Jobs anzeigen und verwalten.

Mit dem Befehl **at -c *JOBNUMBER*** können Sie die tatsächlich bei der Ausführung eines Jobs ausgeführten Befehle überprüfen. Dieser Befehl zeigt zuerst die *Umgebung* für den einzurichtenden Job an, um die Umgebung des Benutzers, der den Job erstellt hat, zum Zeitpunkt der Erstellung darzustellen, gefolgt von den auszuführenden Befehlen.

Entfernen von Jobs

Mit dem Befehl **atrm *JOBNUMBER*** wird ein terminierter Job entfernt. Entfernen Sie einen terminierten Job, wenn er nicht mehr nötig ist. Zum Beispiel, wenn eine Remote-Firewall-Konfiguration erfolgreich war und nicht zurückgesetzt werden muss.



Literaturhinweise

Manpages **at(1)** und **atd(8)**

/usr/share/doc/at/timespec

► Angeleitete Übung

Terminieren eines verschobenen Benutzerjobs

In dieser Übung terminieren Sie mit dem Befehl **at** mehrere Befehle so, dass sie zu bestimmten, zukünftigen Zeitpunkten ausgeführt werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Die Ausführung eines Jobs zu einem bestimmten, zukünftigen Zeitpunkt terminieren
- Befehle überprüfen, die ein terminierter Job ausführt
- Terminierte Jobs löschen

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** das Skript **lab scheduling-at start** aus, um diese Übung zu beginnen. Dieses Skript stellt sicher, dass die Umgebung bereinigt und richtig eingerichtet ist.

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

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Terminieren Sie mit dem Befehl **at** die Ausführung eines Jobs in drei Minuten ab jetzt. Der Job muss die Ausgabe des Befehls **date** in **/home/student/myjob.txt** speichern.

- 2.1. Übergeben Sie mit dem Befehl **echo** die Zeichenfolge **date >> /home/student/myjob.txt** als Eingabe für den Befehl **at**, damit der Job in drei Minuten ab jetzt ausgeführt wird.

```
[student@servera ~]$ echo "date >> /home/student/myjob.txt" | at now +3min
warning: commands will be executed using /bin/sh
job 1 at Thu Mar 21 12:30:00 2019
```

- 2.2. Listen Sie mit dem Befehl **atq** die terminierten Jobs auf.

```
[student@servera ~]$ atq
1 Thu Mar 21 12:30:00 2019 a student
```

Kapitel 2 | Terminieren zukünftiger Tasks

- 2.3. Überwachen Sie mit dem Befehl **watch atq** die Warteschlange der verschobenen Jobs in Echtzeit. Der Job wird nach seiner Ausführung aus der Warteschlange entfernt.

```
[student@servera ~]$ watch atq
Every 2.0s: atq      servera.lab.example.com: Thu Mar 21 12:30:00 2019
1 Thu Mar 21 12:30:00 2019 a student
```

Der vorherige Befehl **watch** aktualisiert standardmäßig alle zwei Sekunden die Ausgabe von **atq**. Nachdem der verschobene Job aus der Warteschlange entfernt wurde, drücken Sie **Strg+c**, um **watch** zu beenden und zur Shell-Eingabeaufforderung zurückzukehren.

- 2.4. Überprüfen Sie mit dem Befehl **cat**, ob der Inhalt von **/home/student/myjob.txt** mit der Ausgabe des Befehls **date** übereinstimmt.

```
[student@servera ~]$ cat myjob.txt
Thu Mar 21 12:30:00 IST 2019
```

Die vorherige Ausgabe stimmt mit der Ausgabe des Befehls **date** überein, was bestätigt, dass der terminierte Job erfolgreich ausgeführt wurde.

- 3. Terminieren Sie mit dem Befehl **at** einen Job interaktiv mit der Warteschlange **g**, die zur **teatime** (16:00 Uhr) ausgeführt wird. Der Job sollte einen Befehl ausführen, der die Meldung **It's teatime** in **/home/student/tea.txt** ausgibt. Die neuen Meldungen sollten an die Datei **/home/student/tea.txt** angehängt werden.

```
[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 Thu Mar 21 16:00:00 2019
```

- 4. Terminieren Sie mit dem Befehl **at** einen weiteren Job interaktiv mit der Warteschlange **b**, die um **16:05** ausgeführt wird. Der Job sollte einen Befehl ausführen, der die Meldung **The cookies are good** in **/home/student/cookies.txt** ausgibt. Die neuen Meldungen sollten an die Datei **/home/student/cookies.txt** angehängt werden.

```
[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 Thu Mar 21 16:05:00 2019
```

- 5. Überprüfen Sie die Befehle in den ausstehenden Jobs.

- 5.1. Zeigen Sie mit dem Befehl **atq** die Jobnummern der ausstehenden Jobs an.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
```

Kapitel 2 | Terminieren zukünftiger Tasks

Achten Sie auf die Jobnummern in der vorherigen Ausgabe. Diese Jobnummern können auf Ihrem System abweichen.

- 5.2. Zeigen Sie mit dem Befehl **at** die Befehle im ausstehenden Job mit der Nummer 2 an.

```
[student@servera ~]$ at -c 2
...output omitted...
echo "It's teatime" >> /home/student/tea.txt
marcinDELIMITER28d54caa
```

Beachten Sie, dass der vorherige terminierte Job einen **echo**-Befehl ausführt, der die Meldung **It's teatime** an **/home/student/tea.txt** anhängt.

- 5.3. Zeigen Sie mit dem Befehl **at** die Befehle im ausstehenden Job mit der Nummer 3 an.

```
[student@servera ~]$ at -c 3
...output omitted...
echo "The cookies are good" >> /home/student/cookies.txt
marcinDELIMITER1d2b47e9
```

Beachten Sie, dass der vorherige terminierte Job einen **echo**-Befehl ausführt, der die Meldung **The cookies are good** an **/home/student/cookies.txt** anhängt.

- 6. Zeigen Sie mit dem Befehl **atq** die Jobnummer eines Jobs an, der zur **teatime** (16:00 Uhr) ausgeführt wird, und entfernen Sie ihn mit dem Befehl **atrm**.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
[student@servera ~]$ atrm 2
```

- 7. Vergewissern Sie sich, dass der Job mit der terminierten Ausführungszeit **teatime** (16:00 Uhr) nicht mehr vorhanden ist.

- 7.1. Verwenden Sie den Befehl **atq**, um die Liste der ausstehenden Jobs anzuzeigen und zu überprüfen, ob der Job mit der terminierten Ausführungszeit **teatime** (16:00 Uhr) nicht mehr vorhanden ist.

```
[student@servera ~]$ atq
3 Thu Mar 21 16:05:00 2019 b student
```

- 7.2. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab scheduling-at finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit ist die angeleitete Übung beendet.

Terminieren wiederkehrender Benutzerjobs

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei eines Benutzers zu terminieren.

Beschreibung wiederkehrender Benutzerjobs

Jobs, deren Ausführung wiederholt werden soll, werden *wiederkehrende Jobs* genannt. Red Hat Enterprise Linux-Systeme werden mit dem Daemon **crond** aus dem cronie-Paket ausgeliefert, der standardmäßig speziell für wiederkehrende Jobs aktiviert und gestartet ist. Der Daemon **crond** liest mehrere Konfigurationsdateien: eine pro Benutzer (die mit dem Befehl **crontab** bearbeitet werden) und eine Reihe systemweiter Dateien. Diese Konfigurationsdateien ermöglichen Benutzern und Administratoren detaillierte Kontrolle darüber, wann die wiederkehrenden Jobs ausgeführt werden sollen.

Wenn ein terminierter Befehl eine Ausgabe oder einen Fehler erzeugt, die nicht umgeleitet werden, versucht der Daemon **crond** diese Ausgabe oder diesen Fehler per E-Mail über den auf dem System konfigurierten Mail-Server an den Benutzer zu senden, der Eigentümer dieses Jobs ist (soweit nicht überschrieben). Je nach Umgebung ist dafür eine zusätzliche Konfiguration erforderlich. Die Ausgabe oder der Fehler des terminierten Befehls kann in verschiedene Dateien umgeleitet werden.

Terminieren wiederkehrender Benutzerjobs

Normale Benutzer können mit dem Befehl **crontab** ihre Jobs verwalten. Dieser Befehl kann auf vier verschiedene Arten aufgerufen werden:

Crontab-Beispiele

Befehl	Verwendungszweck
crontab -l	Auflisten der Jobs für den aktuellen Benutzer
crontab -r	Entfernen aller Jobs für den aktuellen Benutzer
crontab -e	Bearbeiten von Jobs für den aktuellen Benutzer
crontab filename	Entfernen aller Jobs und Ersetzen der aus <i>filename</i> gelesenen Jobs. Wenn keine Datei angegeben ist, wird stdin verwendet.



Anmerkung

Der Superuser kann mit der Option **-u** für den Befehl **crontab** Jobs eines anderen Benutzers verwalten. Sie sollten mit dem Befehl **crontab** keine Systemjobs verwalten. Verwenden Sie stattdessen die im nächsten Abschnitt beschriebenen Methoden.

Beschreibung des Benutzerjobformats

Der Befehl **crontab -e** ruft standardmäßig Vim auf, es sei denn, in der Umgebungsvariable **EDITOR** wurde etwas anderes festgelegt. Geben Sie einen Job pro Zeile ein. Andere gültige Einträge umfassen: Leerzeilen, normalerweise zum leichteren Lesen, Kommentare, gekennzeichnet durch Zeilen, die mit dem Nummernzeichen (#) beginnen, und UmgebungsvARIABLEn im Format **NAME=WERT**, die sich auf alle Zeilen unterhalb der Zeile auswirken, in der sie deklariert sind. Zu den allgemeinen Variableneinstellungen gehören die Variable **SHELL**, die deklariert, mit welcher Shell die verbleibenden Zeilen der Crontab-Datei interpretiert werden sollen, und die Variable **MAILTO**, die festlegt, wer eine E-Mail-Ausgabe erhalten soll.



Wichtig

Das Senden von E-Mails erfordert unter Umständen die zusätzliche Konfiguration des lokalen Mail-Servers oder des SMTP-Relays auf einem System.

Felder in der Datei **crontab** sind in der folgenden Reihenfolge angeordnet:

- Minuten
- Stunden
- Tag des Monats
- Monat
- Wochentag
- Befehl



Wichtig

Wenn die Felder **Tag des Monats** und **Wochentag** beide kein * enthalten, wird der Befehl ausgeführt, wenn eines dieser Felder einen Wert enthält. Um beispielsweise einen Befehl an jedem 15. eines Monats und an jedem Freitag um **12:15** Uhr auszuführen, verwenden Sie das folgende Jobformat:

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

Für die ersten fünf Felder werden die gleichen Syntaxregeln verwendet:

- * für „egal“/immer
- Eine Zahl, die die Anzahl an Stunden oder Minuten, ein Datum oder einen Wochentag angibt. Bei Wochentagen steht **0** für Sonntag, **1** für Montag, **2** für Dienstag usw. **7** steht ebenfalls für Sonntag.
- **x-y** für einen Bereich, **x** bis einschließlich **y**
- **x, y** für Listen. Listen können ebenfalls Bereiche enthalten, zum Beispiel **5, 10-13, 17** in der Spalte **Minuten**. Auf diese Weise wird festgelegt, dass ein Job 5, 10, 11, 12, 13 und 17 Minuten nach der vollen Stunde ausgeführt werden soll.
- **/x** zum Festlegen eines Intervalls von **x**. Zum Beispiel wird bei der Angabe von ***/7** in der Spalte **Minuten** ein Job alle sieben Minuten ausgeführt.

Kapitel 2 | Terminieren zukünftiger Tasks

Außerdem können die dreibuchstabigen englischen Abkürzungen für Monate und Wochentage verwendet werden. Zum Beispiel: Jan, Feb und Mon, Tue.

Das letzte Feld enthält den in der Standard-Shell auszuführenden Befehl. In der Umgebungsvariable **SHELL** kann die Shell für den terminierten Befehl geändert werden. Wenn der Befehl ein nicht maskiertes Prozentzeichen (%) enthält, dann wird das Prozentzeichen als Zeichen für den Zeilenvorschub behandelt und alles nach dem Prozentzeichen wird an den Befehl in **stdin** übergeben.

Beispiele für wiederkehrende Benutzerjobs

In diesem Abschnitt werden einige Beispiele für wiederkehrende Jobs beschrieben.

- Der folgende Job führt jährlich am 2. Februar um genau 9 Uhr den Befehl **/usr/local/bin/yearly_backup** aus.

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

- Der folgende Job sendet an jedem Freitag im Juli alle fünf Minuten zwischen 9 Uhr und 17 Uhr eine E-Mail mit dem Wort **Chime** an den Eigentümer dieses Jobs.

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

Der vorherige Stundenbereich **9-16** bedeutet, dass der Job-Timer um neun Uhr (09:00) beginnt und bis zum Ende der sechzehnten Stunde (16:59) fortgesetzt wird. Der Job beginnt um **09:00** und die letzte Ausführung erfolgt um **16:55**, weil fünf Minuten ab **16:55 17:00** ergäbe, was außerhalb des festgelegten Zeitbereichs liegt.

- Der folgende Job führt jede Woche zwei Minuten vor Mitternacht den Befehl **/usr/local/bin/daily_report** aus.

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

- Der folgende Job führt an jedem Werktag (Montag bis Freitag) um 9 Uhr den Befehl **mutt** zum Senden der E-Mail-Nachricht **Checking in** an den Empfänger **boss@example.com** aus.

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



Literaturhinweise

Manpages **cron(8)**, **crontab(1)** und **crontab(5)**

► Angeleitete Übung

Terminieren wiederkehrender Benutzerjobs

In dieser Übung terminieren Sie als unprivilegierter Benutzer Befehle mit dem Befehl **crontab** so, dass sie nach einem sich wiederholenden Zeitplan ausgeführt werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Wiederkehrende Jobs als unprivilegierter Benutzer terminieren
- Befehle überprüfen, die ein terminierter wiederkehrender Job ausführt
- Terminierte wiederkehrende Jobs entfernen

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** das Skript **lab scheduling-cron start** aus, um diese Übung zu beginnen. Dieses Skript stellt sicher, dass die Umgebung bereinigt und richtig eingerichtet ist.

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

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Terminieren Sie als **student** einen wiederkehrenden Job, der alle zwei Minuten zwischen 8 Uhr und 21 Uhr das aktuelle Datum und die Uhrzeit an **/home/student/my_first_cron_job.txt** anhängt. Der Job darf nur von Montag bis Freitag ausgeführt werden, nicht am Samstag oder Sonntag.



Wichtig

Wenn Sie an dieser praktischen Übung außerhalb des Tages und der Uhrzeit arbeiten, die in der obigen Anleitung angegeben sind, sollten Sie die Systemzeit und/oder das Datum entsprechend anpassen, damit der Job ausgeführt wird, während Sie arbeiten.

- 2.1. Öffnen Sie mit dem Befehl **crontab -e** die Crontab im Standardtexteditor.

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

Kapitel 2 | Terminieren zukünftiger Tasks

- 2.2. Fügen Sie die folgenden Zeilen ein.

```
* /2 08-20 * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

- 2.3. Drücken Sie im Texteditor **Esc** und geben Sie **:wq** ein, um die Änderungen zu speichern und den Editor zu verlassen. Wenn der Editor beendet ist, sollte die folgende Ausgabe angezeigt werden:

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

Die vorherige Ausgabe bestätigt, dass der Job erfolgreich terminiert wurde.

- 3. Listen Sie mit dem Befehl **crontab -l** die geplanten wiederkehrenden Jobs auf. Überprüfen Sie den Befehl, den Sie im vorherigen Schritt für die Ausführung als wiederkehrenden Job terminiert haben.

```
[student@servera ~]$ crontab -l
* /2 08-20 * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

Beachten Sie, dass der vorherige terminierte Job den Befehl **/usr/bin/date** ausführt und die Ausgabe an **/home/student/my_first_cron_job.txt** anhängt.

- 4. Verwenden Sie den Befehl **while**, damit Ihre Shell-Eingabeaufforderung im Ruhezustand verbleibt, bis die Datei **/home/student/my_first_cron_job.txt** als Ergebnis der erfolgreichen Ausführung des von Ihnen terminierten wiederkehrenden Jobs erstellt wurde. Warten Sie, bis die Shell-Eingabeaufforderung angezeigt wird.

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

Der vorherige Befehl **while** verwendet **! test -f**, um die Ausführung einer Schleife von **sleep 1s**-Befehlen fortzusetzen, bis die Datei **my_first_cron_job.txt** im Verzeichnis **/home/student** erstellt wurde.

- 5. Überprüfen Sie mit dem Befehl **cat**, ob der Inhalt von **/home/student/my_first_cron_job.txt** mit der Ausgabe des Befehls **date** übereinstimmt.

```
[student@servera ~]$ cat my_first_cron_job.txt
Fri Mar 22 13:56:01 IST 2019
```

Die vorherige Ausgabe kann auf Ihrem System abweichen.

- 6. Entfernen Sie alle wiederkehrenden Jobs, deren Ausführung als **student** terminiert ist.

- 6.1. Entfernen Sie mit dem Befehl **crontab -r** alle terminierten wiederkehrenden Jobs für **student**.

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

- 6.2. Vergewissern Sie sich mit dem Befehl **crontab -l**, dass keine wiederkehrenden Jobs für **student** vorhanden sind.

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

6.3. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab scheduling-cron finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit ist die angeleitete Übung beendet.

Terminieren wiederkehrender Systemjobs

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei des Systems und Verzeichnissen zu terminieren.

Beschreibung wiederkehrender Systemjobs

Systemadministratoren müssen häufig wiederkehrende Jobs ausführen. Die beste Vorgehensweise besteht darin, diese Jobs über Systemkonten anstatt über Benutzerkonten auszuführen. Planen Sie also die Ausführung dieser Jobs nicht mit dem Befehl **crontab**, sondern verwenden Sie stattdessen systemweite Crontab-Dateien. Jobeinträge in den systemweiten Crontab-Dateien ähneln denen der Crontab-Einträge von Benutzern, mit der Ausnahme, dass die systemweiten Crontab-Dateien über ein zusätzliches Feld vor dem Befehlsfeld verfügen: ein Feld für den Benutzer, unter dem der Befehl ausgeführt werden soll.

Die Datei **/etc/crontab** enthält ein hilfreiches Syntaxdiagramm in den Kommentaren.

```
# 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 ...
# | | | | |
# * * * * * user-name command to be executed
```

Wiederkehrende Systemjobs werden an zwei Stellen definiert: in der Datei **/etc/crontab** und in Dateien im Verzeichnis **/etc/cron.d/**. Sie sollten immer Ihre benutzerdefinierten Crontab-Dateien im Verzeichnis **/etc/cron.d** erstellen, um wiederkehrende Systemjobs zu terminieren. Legen Sie die benutzerdefinierte Crontab-Datei in **/etc/cron.d** ab, um zu verhindern, dass sie bei einer Paketaktualisierung beim **/etc/crontab**-Hersteller überschrieben wird, die den vorhandenen Inhalt in **/etc/crontab** überschreiben könnte. Pakete, die wiederkehrende Systemjobs erfordern, legen ihre Crontab-Dateien in **/etc/cron.d/** ab, die die Jobeinträge enthält. Administratoren verwenden diesen Speicherort ebenfalls, um zusammengehörige Jobs in einer einzigen Datei zu gruppieren.

Das Crontab-System enthält außerdem Repositorys für Skripte, die stündlich, täglich, wöchentlich und monatlich ausgeführt werden müssen. Diese Repositorys sind die Verzeichnisse **/etc/cron.hourly/**, **/etc/cron.daily/**, **/etc/cron.weekly/** und **/etc/cron.monthly/**. Diese Verzeichnisse enthalten ausführbare Shell-Skripte, keine Crontab-Dateien.



Wichtig

Denken Sie daran, alle Skripte, die Sie in diesen Verzeichnissen ablegen, in ausführbare Skripte umzuwandeln. Wenn ein Skript nicht ausführbar ist, wird es nicht ausgeführt. Mit dem Befehl **chmod +x script_name** wandeln Sie ein Skript in ein ausführbares Skript um.

Der aus der Datei **/etc/cron.d/0hourly** aufgerufene Befehl **run-parts** führt die **/etc/cron.hourly/***-Skripte aus. Der Befehl **run-parts** führt die täglichen, wöchentlichen und monatlichen Jobs aus, er wird aber aus einer anderen Konfigurationsdatei, **/etc/anacrontab**, aufgerufen.



Anmerkung

In der Vergangenheit wurde mit dem separaten Service **anacron** die Datei **/etc/anacrontab** verarbeitet, aber in Red Hat Enterprise Linux 7 und später analysiert der reguläre Service **crond** diese Datei.

Der Zweck von **/etc/anacrontab** ist es sicherzustellen, dass wichtige Jobs stets ausgeführt und nicht versehentlich übersprungen werden, weil das System ausgeschaltet oder im Standby-Modus war, als die Jobs ausgeführt werden sollten. Wenn beispielsweise ein Systemjob, der täglich ausgeführt wird, am letzten Fälligkeitstermin aufgrund eines Reboots des Systems nicht ausgeführt wurde, wird der Job ausgeführt, wenn das System wieder bereit ist. Es kann jedoch eine Verzögerung von einigen Minuten beim Starten des Jobs auftreten, abhängig vom Wert des Parameters **Delay in minutes**, der für den Job in **/etc/anacrontab** angegeben ist.

/var/spool/anacron/ enthält verschiedene Dateien für jeden der täglichen, wöchentlichen und monatlichen Jobs, um festzustellen, ob ein bestimmter Job ausgeführt wurde. Wenn **crond** einen Job aus **/etc/anacrontab** startet, aktualisiert es die Zeitstempel dieser Dateien. Anhand dieses Zeitstempels wird ermittelt, wann ein Job zuletzt ausgeführt wurde. Die Syntax von **/etc/anacrontab** weicht von der regulärer **crontab**-Konfigurationsdateien ab. Sie umfasst genau die vier folgenden Felder pro Zeile:

- **Period in days**

Das Intervall in Tagen für den Job, der nach einem sich wiederholenden Zeitplan ausgeführt wird. Dieses Feld akzeptiert eine Ganzzahl oder ein Makro als Wert. Zum Beispiel entspricht das Makro **@daily** der Ganzzahl **1**. Dies bedeutet, dass der Job täglich ausgeführt wird. Genauso entspricht das Makro **@weekly** der Ganzzahl **7**. Dies bedeutet, dass der Job wöchentlich ausgeführt wird.

- **Delay in minutes**

Die Dauer, für die der **crond**-Daemon warten sollte, bis er den Job startet.

- **Job identifier**

Der eindeutige Name, mit dem der Job in den Protokollmeldungen identifiziert wird.

- **Befehl**

Der auszuführende Befehl.

Die Datei **/etc/anacrontab** enthält außerdem Deklarationen von Umgebungsvariablen mit der Syntax **NAME=Wert**. Die Variable **START_HOURS_RANGE** ist von besonderem Interesse, sie gibt

Kapitel 2 | Terminieren zukünftiger Tasks

das Zeitintervall für die Ausführung der Jobs an. Jobs werden nicht außerhalb dieses Intervalls gestartet. Wenn an einem bestimmten Tag ein Job nicht innerhalb dieses Zeitintervalls ausgeführt wird, muss der Job bis zum nächsten Tag auf die Ausführung warten.

Einführung in den Systemd-Timer

Mit der Einführung von **systemd** in Red Hat Enterprise Linux 7 ist nun eine neue Planungsfunktion verfügbar: **systemd-Timer-Units**. Eine **systemd**-Timer-Einheit aktiviert eine andere Einheit eines anderen Typs (z. B. einen Service). Der dieser Name stimmt mit dem Namen der Timer-Einheit überein. Die Timer-Einheit erlaubt die Timer-bezogene Aktivierung anderer Einheiten. Zur Vereinfachen des Debuggens protokolliert **systemd** Timer-Events in Systemjournalen.

Beispiel für eine Timer-Einheit

Das Paket `sysstat` enthält die **systemd**-Timer-Einheit `sysstat-collect.timer`, die alle 10 Minuten Systemstatistiken erfasst. Die folgende Ausgabe zeigt die Konfigurationszeilen von `/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
```

Der Parameter **OnCalendar=*:00/10** gibt an, dass diese Timer-Einheit die entsprechende Einheit (`sysstat-collect.service`) alle 10 Minuten aktiviert. Sie können aber auch komplexere Zeitintervalle angeben. Zum Beispiel bewirkt der Wert **2019-03-*12:35,37,39:16** für den Parameter **OnCalendar**, dass die Timer-Einheit die entsprechende Serviceeinheit während des gesamten Monats März 2019 täglich um **12:35:16, 12:37:16** und **12:39:16** aktiviert. Sie können auch relative Timer mit Parametern wie **OnUnitActiveSec** angeben. Zum Beispiel bewirkt die Option **OnUnitActiveSec=15min**, dass die Timer-Einheit die entsprechende Einheit 15 Minuten nach der letzten Aktivierung der Einheit auslöst.



Wichtig

Ändern Sie im Verzeichnis `/usr/lib/systemd/system` keine Konfigurationsdateien für Einheiten, da bei einer Aktualisierung des Herstellerpakets der Konfigurationsdatei möglicherweise die in dieser Datei vorgenommenen Konfigurationsänderungen überschrieben werden. Erstellen Sie deshalb im Verzeichnis `/etc/systemd/system` eine Kopie der zu ändernden Unit-Konfigurationsdatei und ändern Sie dann die Kopie, damit die Konfigurationsänderungen, die Sie für die Unit vornehmen, nicht durch eine Aktualisierung des Herstellerpakets überschrieben werden. Wenn in den Verzeichnissen `/usr/lib/systemd/system` und `/etc/systemd/system` zwei Dateien mit demselben Namen vorhanden sind, analysiert **systemd** die Datei im Verzeichnis `/etc/systemd/system`.

Stellen Sie nach der Änderung der Konfigurationsdatei der Timer-Einheit mit dem Befehl `systemctl daemon-reload` sicher, dass **systemd** über die Änderungen informiert wird. Dieser Befehl lädt die Manager-Konfiguration **systemd** erneut.

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

Aktivieren Sie nach dem Neuladen der **systemd**-Manager-Konfiguration die Timer-Einheit mit dem folgenden **systemctl**-Befehl.

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



Literaturhinweise

Manpages **crontab(5)**, **anacron(8)**, **anacrontab(5)**, **systemd.time(7)**, **systemd.timer(5)** und **crond(8)**

► Angeleitete Übung

Terminieren wiederkehrender Systemjobs

In dieser Übung terminieren Sie Befehle zur Ausführung nach verschiedenen Zeitplänen, indem Sie den crontab-Verzeichnissen des Systems Konfigurationsdateien hinzufügen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Einen wiederkehrenden Systemjob terminieren, um die Anzahl der aktiven Benutzer zu zählen
- Die **systemd**-Timer-Einheit aktualisieren, die Systemaktivitätsdaten erfasst

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** das Skript **lab scheduling-system start** aus, um diese Übung zu beginnen. Dieses Skript stellt sicher, dass die Umgebung bereinigt und richtig eingerichtet ist.

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

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum Konto des **root**-Benutzers zu wechseln.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Terminieren Sie einen wiederkehrenden Systemjob, der eine Protokollmeldung generiert, die die Anzahl der derzeit aktiven Benutzer im System angibt. Der Job muss täglich laufen. Mit dem Befehl **w -h | wc -l** können Sie die Anzahl der aktuell aktiven Benutzer im System abrufen. Verwenden Sie auch den Befehl **logger**, um die Protokollmeldung zu generieren.

- 3.1. Erstellen Sie eine Skriptdatei mit dem Namen **/etc/cron.daily/usercount** und dem folgenden Inhalt. Erstellen Sie die Skriptdatei mit dem Befehl **vi /etc/cron.daily/usercount**.

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

- 3.2. Aktivieren Sie mit dem Befehl **chmod** die Ausführungsberechtigung (**x**) auf **/etc/cron.daily/usercount**.

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

- 4. Das Paket **sysstat** stellt die **systemd**-Einheiten, **sysstat-collect.timer** und **sysstat-collect.service**, bereit. Die Timer-Einheit löst die Serviceeinheit alle 10 Minuten aus, um Systemaktivitätsdaten mit dem aufgerufenen Shell-Skript **/usr/lib64/sa/sa1** zu erfassen. Stellen Sie sicher, dass das Paket **sysstat** installiert ist, und ändern Sie die Konfigurationsdatei der Timer-Einheit, damit die Systemaktivitätsdaten alle zwei Minuten erfasst werden.

- 4.1. Installieren Sie mit dem Befehl **yum** das Paket **sysstat**.

```
[root@servera ~]# yum install sysstat
...output omitted...
Is this ok [y/N]: y
...output omitted...
Installed:
  sysstat-11.7.3-2.el8.x86_64           lm_sensors-
  libs-3.4.0-17.20180522git70f7e08.el8.x86_64

Complete!
```

- 4.2. Kopieren Sie **/usr/lib/systemd/system/sysstat-collect.timer** nach **/etc/systemd/system/sysstat-collect.timer**.

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



Wichtig

Sie sollten keine Dateien aus dem Verzeichnis **/usr/lib/systemd** bearbeiten. Mit **systemd** können Sie die Datei der Einheit in das Verzeichnis **/etc/systemd/system** kopieren und diese Kopie bearbeiten. Der **systemd**-Prozess analysiert Ihre angepasste Kopie anstelle der Datei im Verzeichnis **/usr/lib/systemd**.

- 4.3. Bearbeiten Sie **/etc/systemd/system/sysstat-collect.timer** so, dass die Timer-Einheit alle zwei Minuten ausgeführt wird. Ersetzen Sie außerdem in der Konfigurationsdatei der Einheit jedes Vorkommen der Zeichenfolge **10 minutes** durch **2 minutes**, einschließlich der Vorkommen in den Kommentarzeilen. Sie können den Befehl **vi /etc/systemd/system/sysstat-collect.timer** verwenden, um die Konfigurationsdatei anzupassen.

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

Kapitel 2 | Terminieren zukünftiger Tasks

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

[Timer]
OnCalendar=*:00/02

[Install]
WantedBy=sysstat.service
```

Die vorherigen Änderungen bewirken, dass die Einheit **sysstat-collect.timer** alle zwei Minuten die Einheit **sysstat-collect.service** auslöst, die **/usr/lib64/sa/sa1 1 1** ausführt. Bei der Ausführung von **/usr/lib64/sa/sa1 1 1** werden die Systemaktivitätsdaten in einer Binärdatei im Verzeichnis **/var/log/sa** erfasst.

- 4.4. Überprüfen Sie mit dem Befehl **systemctl daemon-reload**, ob **systemd** die Änderungen bekannt sind.

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

- 4.5. Aktivieren Sie mit dem Befehl **systemctl** die Timer-Einheit **sysstat-collect.timer**.

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

- 4.6. Verwenden Sie den Befehl **while**, um zu warten, bis die Binärdatei im Verzeichnis **/var/log/sa** erstellt wurde. Warten Sie, bis die Shell-Eingabeaufforderung angezeigt wird.

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

Im obigen **while**-Befehl gibt **ls /var/log/sa | wc -l** zurück, wenn die Datei nicht vorhanden ist, und **1**, wenn sie vorhanden ist. **while** überprüft, ob der Rückgabewert **0** ist, und tritt im positiven Fall in die Schleife ein, die eine Sekunde wartet. Wenn die Datei vorhanden ist, wird die **while**-Schleife beendet.

- 4.7. Überprüfen Sie mit dem **ls -l**, ob die Binärdatei im Verzeichnis **/var/log/sa** innerhalb der letzten zwei Minuten geändert wurde.

```
[root@servera ~]# ls -l /var/log/sa
total 8
-rw-r--r--. 1 root root 5156 Mar 25 12:34 sa25
[root@servera ~]# date
Mon Mar 25 12:35:32 +07 2019
```

Die Ausgabe des vorherigen Befehls kann auf Ihrem System abweichen.

- 4.8. Beenden Sie die Shell des Benutzers **root** und melden Sie sich bei **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab scheduling-system finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit ist die angeleitete Übung beendet.

Verwalten temporärer Dateien

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, **systemd**-Timer zu aktivieren und zu deaktivieren sowie einen Timer zu konfigurieren, der temporäre Dateien verwaltet.

Verwalten temporärer Dateien

Ein modernes System erfordert eine große Anzahl an temporären Dateien und Verzeichnissen. Einige Anwendungen (und Benutzer) verwenden das Verzeichnis **/tmp** für temporäre Daten, während andere einen eher Task-spezifischen Speicherort wie Daemon und benutzerspezifische flüchtige Verzeichnisse unter **/run** nutzen. „Temporär“ bedeutet in diesem Zusammenhang, dass das Dateisystem, das diese Dateien speichert, lediglich im Speicher existiert. Beim Reboot des Systems oder einem Stromausfall geht der Inhalt des temporären Speichers verloren.

Damit ein System ordnungsgemäß läuft, müssen diese Verzeichnisse und Dateien erstellt werden, sofern sie nicht vorhanden sind, und zwar weil Daemons und Skripte von diesen Dateien abhängig sein könnten, und um alte Dateien zu löschen, damit sie keinen Speicherplatz verbrauchen oder fehlerhafte Informationen bereitstellen.

Red Hat Enterprise Linux 7 und später enthalten das neue Tool **systemd-tmpfiles**, das eine strukturierte und konfigurierbare Methode zum Verwalten temporärer Verzeichnisse und Dateien ermöglicht.

Wenn **systemd** ein System startet, gehört **systemd-tmpfiles-setup** zu den ersten Serviceeinheiten, die aufgerufen werden. Dieser Service führt den Befehl **systemd-tmpfiles --create --remove** aus. Dieser Befehl liest die Konfigurationsdateien aus **/usr/lib/tmpfiles.d/* .conf**, **/run/tmpfiles.d/* .conf** und **/etc/tmpfiles.d/* .conf**. Alle Dateien und Verzeichnisse, die in diesen Konfigurationsdateien zum Löschen markiert sind, werden entfernt, und alle Dateien und Verzeichnisse, die zum Erstellen (oder Korrigieren von Berechtigungen) markiert sind, werden erstellt, gegebenenfalls mit den richtigen Berechtigungen.

Bereinigen temporärer Dateien mit einem Systemd-Timer

Um sicherzustellen, dass Systeme mit langer Ausführungszeit ihre Festplatten nicht mit veralteten Daten füllen, löst die **systemd**-Timer-Einheit **systemd-tmpfiles-clean.timer** in regelmäßigen Intervallen **systemd-tmpfiles-clean.service** aus, der den Befehl **systemd-tmpfiles --clean** ausführt.

Die **systemd**-Timer-Konfigurationsdateien enthalten den Abschnitt **[Timer]**, in dem angegeben ist, wie oft der Service mit diesem Namen gestartet werden soll.

Mit dem folgenden **systemctl**-Befehl zeigen Sie den Inhalt der Konfigurationsdatei **systemd-tmpfiles-clean.timer** der Einheit an.

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

Kapitel 2 | Terminieren zukünftiger Tasks

```

#
# 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)

[Timer]
OnBootSec=15min
OnUnitActiveSec=1d

```

In der vorherigen Konfiguration gibt der Parameter **OnBootSec=15min** an, dass die Serviceeinheit **systemd-tmpfiles-clean.service** 15 Minuten nach dem Booten des Systems ausgelöst wird. Der Parameter **OnUnitActiveSec=1d** gibt an, dass ein weiterer Auslöser für die Serviceeinheit **systemd-tmpfiles-clean.service** 24 Stunden nach der letzten Aktivierung der Serviceeinheit aufgerufen wird.

Je nach Anforderung können Sie die Parameter in der Konfigurationsdatei der Timer-Einheit **systemd-tmpfiles-clean.timer** ändern. Zum Beispiel löst der Wert **30min** für den Parameter **OnUnitActiveSec** die Serviceeinheit **systemd-tmpfiles-clean.service** 30 Minuten nach der letzten Aktivierung der Serviceeinheit aus. **systemd-tmpfiles-clean.service** wird daher alle 30 Minuten ausgelöst, nachdem die Änderungen wirksam wurden.

Stellen Sie nach der Änderung der Konfigurationsdatei der Timer-Einheit mit dem Befehl **systemctl daemon-reload** sicher, dass **systemd** über die Änderung informiert wird. Dieser Befehl lädt die **systemd**-Manager-Konfiguration

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

Aktivieren Sie nach dem Neuladen der **systemd**-Manager-Konfiguration mit dem folgenden **systemctl**-Befehl die Timer-Einheit **systemd-tmpfiles-clean.timer**.

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

Manuelles Bereinigen temporärer Dateien

Der Befehl **systemd-tmpfiles --clean** analysiert dieselben Konfigurationsdateien wie der Befehl **systemd-tmpfiles --create**, aber statt Dateien und Verzeichnisse zu erstellen, löscht dieser alle Dateien permanent, die innerhalb des maximalen, in der Konfigurationsdatei festgelegten Zeitraums nicht aufgerufen, geändert oder modifiziert wurden.

Das Format der Konfigurationsdateien für **systemd-tmpfiles** ist auf der Manpage **tmpfiles.d(5)** beschrieben. Die grundlegende Syntax besteht aus sieben Spalten: Type, Path, Mode, UID, GID, Age, und Argument. Type bezieht sich auf die Aktion, die **systemd-tmpfiles** durchführen soll, beispielsweise **d**, um ein Verzeichnis zu erstellen, wenn es noch nicht vorhanden ist, oder **Z**, um SELinux-Kontexte, Dateiberechtigungen und Eigentümerschaft rekursiv wiederherzustellen.

Im Folgenden finden Sie einige Beispiele mit Erklärungen.

Kapitel 2 | Terminieren zukünftiger Tasks

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

Sofern es noch nicht vorhanden ist, legen Sie beim Erstellen von Dateien und Verzeichnissen das Verzeichnis **/run/systemd/seats** an, dessen Eigentümer der Benutzer **root** und die Gruppe **root** sind, und legen Sie die Berechtigungen auf **rwxr-xr-x** fest. Dieses Verzeichnis wird nicht automatisch permanent gelöscht.

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

Erstellen Sie das Verzeichnis **/home/student**, falls es noch nicht vorhanden ist. Falls es bereits vorhanden ist, löschen Sie den gesamten Inhalt. Wenn **systemd-tmpfiles --clean** ausgeführt wird, entfernen Sie alle Dateien, auf die innerhalb eines Zeitraums von mehr als einem Tag kein Zugriff erfolgte und die weder modifiziert noch geändert wurden.

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

Erstellen Sie den symbolischen Link **/run/fstablink**, der auf **/etc/fstab** verweist. Lassen Sie diese Zeile niemals automatisch permanent löschen.

Rangordnung von Konfigurationsdateien

Konfigurationsdateien können an drei Stellen vorkommen:

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

Die Dateien in **/usr/lib/tmpfiles.d/** werden von den relevanten RPM-Paketen bereitgestellt und Sie sollten diese Dateien nicht bearbeiten. Die Dateien in **/run/tmpfiles.d/** sind selbst flüchtige Dateien, die normalerweise von Daemons zum Verwalten ihrer eigenen temporären Laufzeitdateien verwendet werden. Die Dateien in **/etc/tmpfiles.d/** sind für Administratoren vorgesehen, um benutzerdefinierte temporäre Speicherorte zu konfigurieren und vom Hersteller bereitgestellte Standardeinstellungen zu überschreiben.

Wenn eine Datei in **/run/tmpfiles.d/** denselben Dateinamen wie eine Datei in **/usr/lib/tmpfiles.d/** aufweist, dann wird die Datei in **/run/tmpfiles.d/** verwendet. Wenn eine Datei in **/etc/tmpfiles.d/** denselben Dateinamen wie eine Datei in **/run/tmpfiles.d/** oder **/usr/lib/tmpfiles.d/** aufweist, dann wird die Datei in **/etc/tmpfiles.d/** verwendet.

Mithilfe dieser Rangordnungsregeln können Sie die vom Hersteller bereitgestellten Einstellungen einfach durch Kopieren der relevanten Datei in das Verzeichnis **/etc/tmpfiles.d/** überschreiben und diese dann bearbeiten. Auf diese Weise wird sichergestellt, dass sich die vom Administrator bereitgestellten Einstellungen leicht über ein zentrales Konfigurationsverwaltungssystem verwalten lassen und beim Update eines Pakets nicht überschrieben werden.



Anmerkung

Beim Testen neuer oder modifizierter Konfigurationen ist es eventuell hilfreich, die Befehle aus einer Konfigurationsdatei anzuwenden. Geben Sie hierzu den Namen der Konfigurationsdatei in die Befehlszeile ein.



Literaturhinweise

Manpages **systemd-tmpfiles(8)**, **tmpfiles.d(5)**, **stat(1)**, **stat(2)** und
systemd.timer(5)

► Angeleitete Übung

Verwalten temporärer Dateien

In dieser Übung konfigurieren Sie **systemd-tmpfiles**, um zu ändern, wie schnell temporäre Dateien aus **/tmp** entfernt werden, und auch um Dateien aus einem anderen Verzeichnis regelmäßig zu löschen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- **systemd-tmpfiles** konfigurieren, um nicht verwendete temporäre Dateien aus **/tmp** zu entfernen
- **systemd-tmpfiles** konfigurieren, um Dateien regelmäßig aus einem anderen Verzeichnis zu löschen

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** den Befehl **lab scheduling-tempfiles start** aus, um die Übung zu starten. Dieses Skript erstellt die erforderlichen Dateien und stellt sicher, dass die Umgebung korrekt eingerichtet ist.

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

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Konfigurieren Sie **systemd-tmpfiles**, um das Verzeichnis **/tmp** zu bereinigen, damit es keine Dateien enthält, die in den letzten fünf Tagen nicht verwendet wurden. Stellen Sie sicher, dass die Konfiguration nicht durch Paketaktualisierungen überschrieben wird.

- 2.1. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- 2.2. Kopieren Sie **/usr/lib/tmpfiles.d/tmp.conf** in **/etc/tmpfiles.d/** **tmp.conf**.

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

- 2.3. Suchen Sie in **/etc/tmpfiles.d/tmp.conf** nach der Konfigurationszeile, die das Verzeichnis **/tmp** betrifft. Ersetzen Sie das vorhandene Alter der temporären Dateien in dieser Konfigurationszeile durch das neue Alter von **5** Tagen. Entfernen Sie alle anderen Zeilen aus der Datei, einschließlich der Kommentarzeilen. Sie können den Befehl **vim /etc/tmpfiles.d/tmp.conf** verwenden, um die Konfigurationsdatei zu bearbeiten. Die Datei **/etc/tmpfiles.d/tmp.conf** sollte wie folgt aussehen:

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

In der vorherigen Konfiguration ist der Typ **q** identisch mit **d** und weist **systemd-tmpfiles** an, das Verzeichnis **/tmp** zu erstellen, falls dieses nicht vorhanden ist. Die oktalen Berechtigungen für das Verzeichnis müssen auf **1777** festgelegt sein. Sowohl der Eigentümerbenutzer als auch die Eigentümergruppe von **/tmp** muss **root** sein. Das Verzeichnis **/tmp** darf keine temporären Dateien enthalten, die in den letzten fünf Tagen nicht verwendet wurden.

- 2.4. Überprüfen Sie mit dem Befehl **systemd-tmpfiles --clean**, ob die Datei **/etc/tmpfiles.d/tmp.conf** die korrekte Konfiguration enthält.

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

Da der vorherige Befehl keine Fehler zurückgegeben hat, wird bestätigt, dass die Konfigurationseinstellungen korrekt sind.

- 3. Fügen Sie eine neue Konfiguration hinzu, die sicherstellt, dass das Verzeichnis **/run/momentary** mit den auf **root** festgelegten Benutzer- und Gruppeneigentümern vorhanden ist. Die oktalen Berechtigungen für das Verzeichnis müssen **0700** lauten. Die Konfiguration sollte alle Dateien in diesem Verzeichnis löschen, die in den letzten 30 Sekunden nicht verwendet wurden.
- 3.1. Erstellen Sie eine Datei mit dem Namen **/etc/tmpfiles.d/momentary.conf** und folgendem Inhalt. Sie können den Befehl **vim /etc/tmpfiles.d/momentary.conf** verwenden, um die Konfigurationsdatei zu erstellen.

```
d /run/momentary 0700 root root 30s
```

Die vorherige Konfiguration bewirkt, dass **systemd-tmpfiles** sicherstellt, dass das Verzeichnis **/run/momentary** mit den auf **0700** gesetzten oktalen Berechtigungen vorhanden ist. Der Eigentümerbenutzer und die Eigentümergruppe von **/run/momentary** müssen **root** sein. Alle Dateien in diesem Verzeichnis, die in den letzten 30 Sekunden nicht verwendet wurden, müssen gelöscht werden.

- 3.2. Überprüfen Sie mit dem Befehl **systemd-tmpfiles --create**, ob die Datei **/etc/tmpfiles.d/momentary.conf** die entsprechende Konfiguration enthält. Der Befehl erstellt das Verzeichnis **/run/momentary**, falls es noch nicht vorhanden ist.

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

Da der vorherige Befehl keine Fehler zurückgegeben hat, wird bestätigt, dass die Konfigurationseinstellungen korrekt sind.

- 3.3. Überprüfen Sie mit dem Befehl **ls**, ob das Verzeichnis **/run/momentary** mit den entsprechenden Berechtigungen, dem Eigentümer und dem Gruppeneigentümer erstellt wurde.

Kapitel 2 | Terminieren zukünftiger Tasks

```
[root@servera ~]# ls -ld /run/momentary  
drwx----- 2 root root 40 Mar 21 16:39 /run/momentary
```

Beachten Sie, dass der oktale Berechtigungssatz von **/run/momentary 0700** lautet und dass die Benutzer- und die Gruppeneigentümerschaft auf **root** festgelegt sind.

- 4. Überprüfen Sie, ob alle Dateien im Verzeichnis **/run/momentary**, die in den letzten 30 Sekunden nicht verwendet wurden, entsprechend der Konfiguration **systemd-tmpfiles** für das Verzeichnis entfernt wurden.

- 4.1. Erstellen Sie mit dem Befehl **touch** eine Datei mit dem Namen **/run/momentary/testfile**.

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

- 4.2. Konfigurieren Sie mit dem Befehl **sleep** die Shell-Eingabeaufforderung so, dass sie 30 Sekunden nicht zurückkehrt.

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

- 4.3. Entfernen Sie nach Rückkehr der Shell-Eingabeaufforderung mit dem Befehl **systemd-tmpfiles --clean** entsprechend der in **/etc/tmpfiles.d/momentary.conf** angegebenen Regel veraltete Dateien aus **/run/momentary**.

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

Der vorherige Befehl entfernt **/run/momentary/testfile**, weil die Datei 30 Sekunden lang nicht verwendet wurde und aufgrund der in **/etc/tmpfiles.d/momentary.conf** angegebenen Regel entfernt werden sollte.

- 4.4. Vergewissern Sie sich mit dem Befehl **ls -l**, dass die Datei **/run/momentary/testfile** nicht vorhanden ist.

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

- 4.5. Beenden Sie die Shell des Benutzers **root** und melden Sie sich bei **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab_scheduling-tempfiles_finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit ist die angeleitete Übung beendet.

► Quiz

Terminieren zukünftiger Tasks

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus.

- ▶ 1. Welcher Befehl zeigt alle Benutzerjobs an, deren Ausführung als verschobene Jobs terminiert ist?
 - a. `atq`
 - b. `atrm`
 - c. `at -c`
 - d. `at --display`

- ▶ 2. Welcher Befehl entfernt den verschobenen Benutzerjob mit der Jobnummer 5?
 - a. `at -c 5`
 - b. `atrm 5`
 - c. `at 5`
 - d. `at --delete 5`

- ▶ 3. Welcher Befehl zeigt alle wiederkehrenden Benutzerjobs an, die für den aktuell angemeldeten Benutzer terminiert sind?
 - a. `crontab -r`
 - b. `crontab -l`
 - c. `crontab -u`
 - d. `crontab -v`

- ▶ 4. Welches Jobformat führt `/usr/local/bin/daily_backup` stündlich von 9 Uhr bis 18 Uhr an allen Tagen von Montag bis Freitag aus?
 - 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. In welchem Verzeichnis befinden sich die Shell-Skripte, die täglich ausgeführt werden sollen?
 - a. `/etc/cron.d`
 - b. `/etc/cron.hourly`
 - c. `/etc/cron.daily`
 - d. `/etc/cron.weekly`

- 6. In welcher Konfigurationsdatei sind die Einstellungen für die Systemjobs definiert, die täglich, wöchentlich und monatlich ausgeführt werden?
- a. `/etc/crontab`
 - b. `/etc/anacrontab`
 - c. `/etc/inittab`
 - d. `/etc/sysconfig/crond`
- 7. Welche `systemd`-Einheit löst regelmäßig die Bereinigung der temporären Dateien aus?
- a. `systemd-tmpfiles-clean.timer`
 - b. `systemd-tmpfiles-clean.service`
 - c. `dnf-makecache.timer`
 - d. `unbound-anchor.timer`

► Lösung

Terminieren zukünftiger Tasks

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus.

- ▶ 1. Welcher Befehl zeigt alle Benutzerjobs an, deren Ausführung als verschobene Jobs terminiert ist?
 - a. **atq**
 - b. **atrm**
 - c. **at -c**
 - d. **at --display**

- ▶ 2. Welcher Befehl entfernt den verschobenen Benutzerjob mit der Jobnummer 5?
 - a. **at -c 5**
 - b. **atrm 5**
 - c. **at 5**
 - d. **at --delete 5**

- ▶ 3. Welcher Befehl zeigt alle wiederkehrenden Benutzerjobs an, die für den aktuell angemeldeten Benutzer terminiert sind?
 - a. **crontab -r**
 - b. **crontab -l**
 - c. **crontab -u**
 - d. **crontab -v**

- ▶ 4. Welches Jobformat führt **/usr/local/bin/daily_backup** stündlich von 9 Uhr bis 18 Uhr an allen Tagen von Montag bis Freitag aus?
 - 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. In welchem Verzeichnis befinden sich die Shell-Skripte, die täglich ausgeführt werden sollen?
 - a. **/etc/cron.d**
 - b. **/etc/cron.hourly**
 - c. **/etc/cron.daily**
 - d. **/etc/cron.weekly**

- 6. In welcher Konfigurationsdatei sind die Einstellungen für die Systemjobs definiert, die täglich, wöchentlich und monatlich ausgeführt werden?
- a. /etc/crontab
 - b. /etc/anacrontab
 - c. /etc/inittab
 - d. /etc/sysconfig/crond
- 7. Welche **systemd**-Einheit löst regelmäßig die Bereinigung der temporären Dateien aus?
- a. **systemd-tmpfiles-clean.timer**
 - b. **systemd-tmpfiles-clean.service**
 - c. **dnf-makecache.timer**
 - d. **unbound-anchor.timer**

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Jobs, deren einmalige Ausführung terminiert ist, werden als verschobene Jobs oder Tasks bezeichnet.
- Wiederkehrende Benutzerjobs führen die Tasks des Benutzers nach einem sich wiederholenden Zeitplan aus.
- Wiederkehrende Systemjobs führen administrative Aufgaben nach einem sich wiederholenden Zeitplan mit systemweiter Auswirkung aus.
- Die **systemd**-Timer-Units können sowohl verschobene als auch wiederkehrende Jobs ausführen.

Kapitel 3

Tuning der Systemleistung

Ziel

Die Systemleistung durch Festlegen von Tuning-Parametern verbessern und die Planungspriorität von Prozessen festlegen

Ziele

- Die Systemleistung durch Auswahl eines vom Daemon „tuned“ verwalteten Tuning-Profil optimieren
- Die Priorität bestimmter Prozesse mit den Befehlen „nice“ und „renice“ festlegen bzw. aufheben

Abschnitte

- Anpassen von Tuning-Profilen (und angeleitete Übung)
- Beeinflussen der Prozessplanung (und angeleitete Übung)

Praktische Übung

Tuning der Systemleistung

Anpassen von Tuning-Profilen

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, die Systemleistung zu optimieren, indem Sie ein Tuning-Profil auswählen, das vom Daemon **tuned** verwaltet wird.

Systemoptimierung

Systemadministratoren können die Leistung eines Systems optimieren, indem sie verschiedene Geräteeinstellungen basierend auf einer Vielzahl von Anwendungsfall-Workloads anpassen. Der Daemon **tuned** wendet Tuning-Anpassungen sowohl statisch als auch dynamisch mit Tuning-Profilen an, die bestimmte Workload-Anforderungen berücksichtigen.

Konfigurieren von statischem Tuning

Der Daemon **tuned** wendet Systemeinstellungen an, wenn der Service startet oder ein neues Tuning-Profil ausgewählt wird. Statisches Tuning konfiguriert vordefinierte **Kernel**-Parameter in Profilen, die **tuned** zur Laufzeit anwendet. Beim statischen Tuning werden die Kernel-Parameter für die allgemeinen Leistungserwartungen festgelegt und nicht angepasst, wenn sich die Aktivitätsebenen ändern.

Konfigurieren von dynamischem Tuning

Beim dynamischen Tuning überwacht der Daemon **tuned** die Systemaktivität und passt die Einstellungen entsprechend den Änderungen des Laufzeitverhaltens an. Beim dynamischen Tuning wird das Tuning ausgehend von den Anfangseinstellungen, die im ausgewählten Tuning-Profil angegeben sind, kontinuierlich an den aktuellen Workload angepasst.

Beispielsweise werden Speichergeräte beim Starten und Anmelden häufig verwendet, sind jedoch nur minimal aktiv, wenn Benutzer-Workloads aus Webbrowsern und E-Mail-Clients bestehen. In ähnlicher Weise erhöht sich die Aktivität von CPU und Netzwerkgeräten bei Spitzenauslastungen während des gesamten Arbeitstages. Der Daemon **tuned** überwacht die Aktivität dieser Komponenten und passt die Parametereinstellungen an, um die Leistung während Zeiten mit hoher Aktivität zu maximieren und Einstellungen bei geringer Aktivität zu reduzieren. Der Daemon **tuned** verwendet Leistungsparameter aus den vordefinierten Tuning-Profilen.

Installieren und Aktivieren von Tuned

Eine minimale Installation von Red Hat Enterprise Linux 8 beinhaltet und aktiviert standardmäßig das Paket *tuned*. So installieren und aktivieren Sie das Paket manuell:

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

Auswählen eines Tuning-Profil

Die Anwendung Tuned umfasst Profile, die in die folgenden Kategorien unterteilt sind:

Kapitel 3 | Tuning der Systemleistung

- Profile zum Sparen von Energie
- Profile zur Steigerung der Leistung

Die Profile zur Steigerung der Leistung enthalten Profile für folgende Aspekte:

- Geringe Latenz für Speicher und Netzwerk
- Hoher Durchsatz für Speicher und Netzwerk
- Leistung des virtuellen Rechners
- Leistung des Virtualisierungshosts

Mit Red Hat Enterprise Linux 8 bereitgestellte Profile

Tuned-Profil	Zweck
ausgewogen	Ideal für Systeme, die einen Kompromiss zwischen Energieeinsparung und Leistung erfordern.
desktop	Abgeleitet vom Profil balanced . Ermöglicht eine schnellere Antwort interaktiver Anwendungen.
throughput-performance	Optimiert das System für maximalen Durchsatz.
latency-performance	Ideal für Serversysteme, die eine geringe Latenz erfordern, geht jedoch auf Kosten des Stromverbrauchs.
network-latency	Abgeleitet vom Profil latency-performance . Dieses Profil aktiviert zusätzliche Parameter für die Netzwerkoptimierung zur Bereitstellung einer geringen Netzwerklatenz.
network-throughput	Abgeleitet vom Profil throughput-performance . Für den maximalen Netzwerkdurchsatz werden zusätzliche Parameter zur Netzwerkoptimierung angewendet.
Energiespar	Optimiert das System für maximale Energieeinsparung.
oracle	Optimiert für Oracle-Datenbanklasten, basiert auf dem Profil throughput-performance .
virtual-guest	Optimiert das System für maximale Leistung, wenn es auf einem virtuellen Rechner ausgeführt wird.
virtual-host	Optimiert das System für maximale Leistung, wenn es als Host für virtuelle Rechner ausgeführt wird.

Verwalten von Profilen über die Befehlszeile

Mit dem Befehl **tuned-adm** werden Einstellungen des Daemon **tuned** geändert. Der Befehl **tuned-adm** kann aktuelle Einstellungen abfragen, verfügbare Profile auflisten, ein Tuning-Profil für das System empfehlen, Profile direkt ändern oder das Tuning deaktivieren.

Ein Systemadministrator fragt das aktuell aktive Tuning-Profil mit **tuned-adm active** ab.

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

Kapitel 3 | Tuning der Systemleistung

Der Befehl **tuned-adm list** listet alle verfügbaren Tuning-Profile auf, sowohl die integrierten Profile als auch benutzerdefinierte, von einem Systemadministrator erstellte Tuning-Profile.

```
[root@host ~]# tuned-adm list
Available profiles:
- balanced
- desktop
- latency-performance
- network-latency
- network-throughput
- powersave
- sap
- throughput-performance
- virtual-guest
- virtual-host
Current active profile: virtual-guest
```

Mit **tuned-adm profile *profilename*** stellen Sie das aktive Profil auf ein Profil um, das den aktuellen Tuning-Anforderungen des Systems besser entspricht.

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

Der Befehl **tuned-adm** kann ein Tuning-Profil für das System empfehlen. Dieser Mechanismus wird verwendet, um das Standardprofil eines Systems nach der Installation festzulegen.

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



Anmerkung

Die Ausgabe von **tuned-adm recommend** basiert auf verschiedenen Systemmerkmalen, einschließlich der Angabe, ob das System ein virtueller Rechner ist, und anderer vordefinierter Kategorien, die während der Systeminstallation ausgewählt wurden.

Um die vom aktuellen Profil vorgenommenen Einstellungsänderungen rückgängig zu machen, wechseln Sie entweder zu einem anderen Profil oder deaktivieren den tuned-Dämon. Deaktivieren Sie die **tuned**-Tuning-Aktivität mit **tuned-adm off**.

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

Verwalten von Profilen mit Web Console

Melden Sie sich mit privilegiertem Zugriff an, um Systemleistungsprofile mit Web Console zu verwalten. Aktivieren Sie die Option **Reuse my password for privileged tasks**. Dies ermöglicht dem Benutzer Befehle mit sudo-Berechtigungen auszuführen, die Systemleistungsprofile ändern.

Kapitel 3 | Tuning der Systemleistung

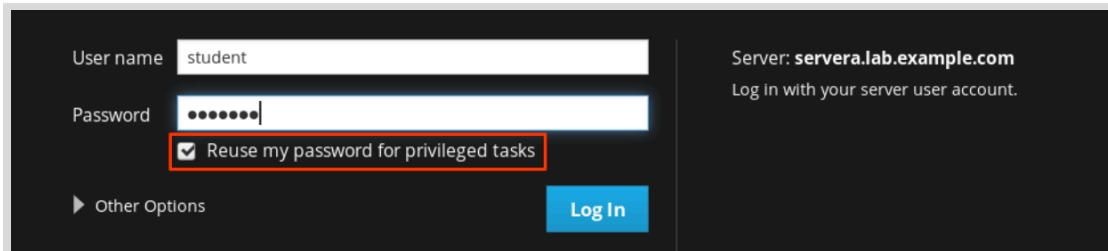


Abbildung 3.1: Privilegierte Anmeldung bei Web Console

Klicken Sie als privilegierter Benutzer in der linken Navigationsleiste auf die Menüoption **Systems**. Das aktuell aktive Profil wird im Feld **Performance Profile** angezeigt. Um ein anderes Profil auszuwählen, klicken Sie auf den Link für das aktive Profil.

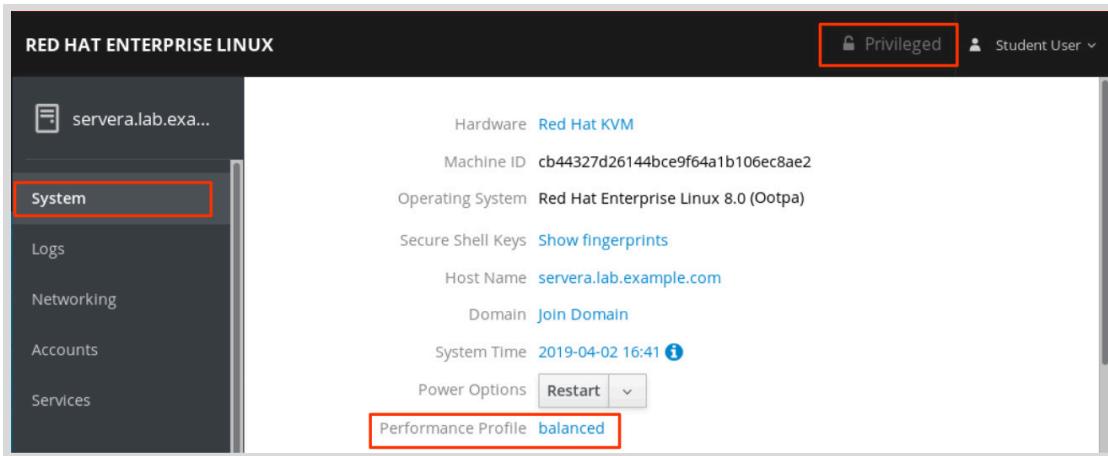


Abbildung 3.2: Aktives Leistungsprofil

Scrollen Sie in der Benutzeroberfläche **Change Performance Profile** durch die Profilliste, um ein für den Systemzweck am besten geeignetes Profil auszuwählen.

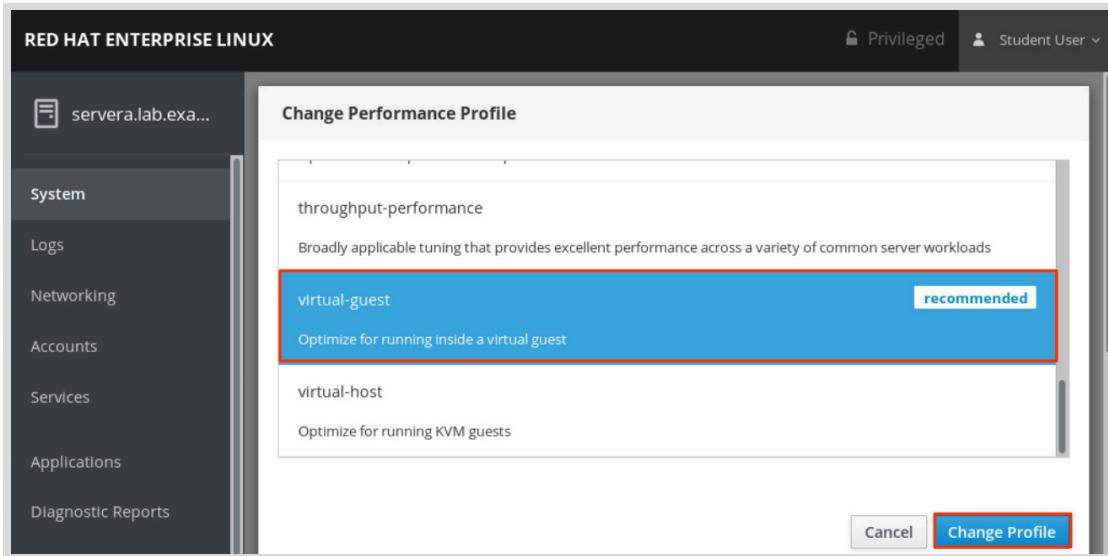


Abbildung 3.3: Ein bevorzugtes Leistungsprofil auswählen

Um die Änderungen zu überprüfen, kehren Sie zur **System**-Hauptseite zurück und vergewissern Sie sich, ob das gewünschte Profil im Feld **Performance Profile** als aktives Profil angezeigt wird.

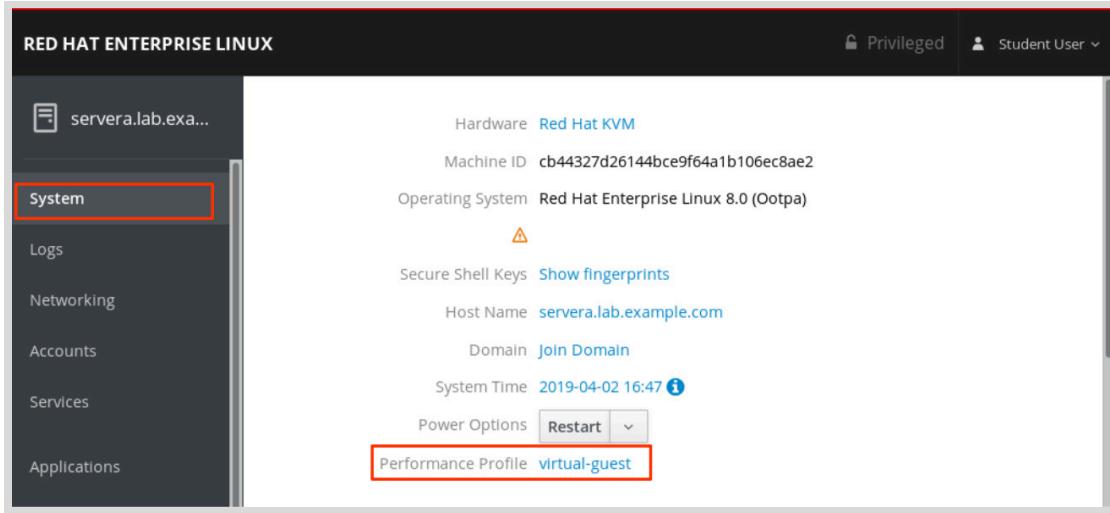


Abbildung 3.4: Aktives Profil überprüfen



Literaturhinweise

Manpages **tuned(8)**, **tuned.conf(5)**, **tuned-main.conf(5)** und **tuned-adm(1)**

► Angeleitete Übung

Anpassen von Tuning-Profilen

In dieser Übung optimieren Sie die Leistung eines Servers durch Aktivieren des Service **tuned** und Anwendung eines Tuning-Profilis.

Ergebnisse

Sie sollten in der Lage sein, ein System für die Verwendung eines Tuning-Profilis zu konfigurieren.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab tuning-profiles start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **servera** im Netzwerk erreichbar ist.

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

- 1. Verwenden Sie auf **workstation** SSH, um sich bei **servera** als Benutzer **student** anzumelden. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Überprüfen Sie, ob das Paket **tuned** installiert, aktiviert und gestartet ist.

- 2.1. Überprüfen Sie mit **yum**, ob das Paket **tuned** installiert ist.

```
[student@servera ~]$ yum list tuned  
...output omitted...  
Installed Packages  
tuned.noarch 2.10.0-15.el8 @anaconda
```

- 2.2. Der Befehl **systemctl is-enabled tuned** zeigt an, ob der Service aktiviert ist.

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

- 2.3. Der Befehl **systemctl is-active tuned** zeigt an, ob der Service derzeit ausgeführt wird.

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

- 3. Listen Sie die verfügbaren Tuning-Profile auf und ermitteln Sie das aktive Profil. Geben Sie **student** ein, wenn **sudo** Sie zur Eingabe eines Passworts auffordert.

```
[student@servera ~]$ sudo tuned-adm list  
[sudo] password for student: student  
Available profiles:  
- balanced - General non-specialized tuned profile  
- desktop - Optimize for the desktop use-case  
- 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  
- 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. Ändern Sie das aktuell aktive Tuning-Profil in **powersave** und überprüfen Sie dann die Ergebnisse. Geben Sie **student** ein, wenn **sudo** Sie zur Eingabe eines Passworts auffordert.

- 4.1. Ändern Sie das aktuell aktive Tuning-Profil.

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

- 4.2. Überprüfen Sie, ob **powersave** das aktive Tuning-Profil ist.

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

- 5. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab tuning-profiles finish** aus, um diese Übung abzuschließen.

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

Hiermit ist die angeleitete Übung beendet.

Beeinflussen der Prozessplanung

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, mit den Befehlen **nice** und **renice** für bestimmte Prozesse Prioritäten zu setzen oder aufzuheben.

Planen von Linux-Prozessen und Multitasking

Moderne Computersysteme reichen von Low-End-Systemen mit einer einzigen CPU, die gleichzeitig nur eine Anweisung ausführen kann, bis hin zu leistungsstarken Supercomputern mit Hunderten von **CPUs** und Dutzenden oder sogar Hunderten von Prozessorkernen in jeder **CPU**, die eine riesige Anzahl von Anweisungen parallel ausführen können. All diese Systeme haben jedoch eins gemeinsam: Sie müssen mehr Prozess-Threads ausführen, als sie CPUs haben.

Linux und andere Betriebssysteme führen mithilfe einer Technik, die *Zeitscheiben-Verfahren* oder *Multitasking* genannt wird, mehr Prozesse aus, als es Verarbeitungseinheiten gibt. Der *Prozess-Scheduler* des Betriebssystems wechselt schnell zwischen Prozessen auf einem einzelnen Kern, was den Eindruck hervorruft, dass mehrere Prozesse gleichzeitig ausgeführt werden.

Relative Prioritäten

Unterschiedliche Prozesse haben unterschiedliche Wichtigkeitsstufen. Der Prozess-Scheduler kann so konfiguriert werden, dass unterschiedliche Planungsrichtlinien für verschiedene Prozesse verwendet werden. Die Planungsrichtlinie für die meisten Prozesse, die auf einem regulären System ausgeführt werden, lautet **SCHED_OTHER** (oder **SCHED_NORMAL**). Für unterschiedliche Workload-Anforderungen gibt es jedoch auch unterschiedliche Richtlinien.

Da nicht alle Prozesse gleich wichtig sind, kann Prozessen, die mit der Richtlinie **SCHED_NORMAL** ausgeführt werden, eine relative Priorität zugewiesen werden. Diese Priorität wird der *Nice-Wert* eines Prozesses genannt. Jeder Prozess umfasst **40** verschiedene Nice-Level.

Die Nice-Level reichen von -20 (höchste Priorität) bis 19 (niedrigste Priorität). Standardmäßig erben Prozesse den Nice-Level von ihrem übergeordneten Prozess (normalerweise 0). Höhere Nice-Level geben eine niedrigere Priorität an (der Prozess gibt seine CPU-Verwendung eher auf), während niedrigere Nice-Level eine höhere Priorität angeben (der Prozess gibt die CPU eher nicht frei). Wenn kein Konflikt um Ressourcen besteht, beispielsweise wenn weniger aktive Prozesse als verfügbare CPU-Kerne vorhanden sind, werden auch Prozesse mit einem hohen Nice-Level weiterhin alle verfügbaren CPU-Ressourcen nutzen. Wenn allerdings mehr Prozesse CPU-Zeit erfordern als Kerne verfügbar sind, erhalten die Prozesse mit einem höheren Nice-Level weniger CPU-Zeit als die mit einem niedrigeren Nice-Level.

Festlegen von Nice-Leveln und Berechtigungen

Da die Festlegung eines niedrigen Nice-Levels für einen CPU-intensiven Prozess die Leistung anderer im selben System ausgeführten Prozesse negativ beeinflussen könnte, kann nur der **root**-Benutzer den Nice-Level eines Prozesses *verringern*.

Nicht privilegierte Benutzer dürfen Nice-Level für eigene Prozesse nur *erhöhen*. Sie können weder die Nice-Level ihrer Prozesse verringern, noch die Nice-Level der Prozesse anderer Benutzer ändern.

Anzeigen von Nice-Leveln

Verschiedene Tools zeigen die Nice-Level der laufenden Prozesse an. Prozessmanagementtools, wie z. B. top, zeigen standardmäßig den Nice-Level an. Andere Tools, wie der Befehl **ps**, zeigen Nice-Level an, wenn die richtigen Optionen verwendet werden.

Anzeigen von Nice-Leveln mit top

Mit dem Befehl **top** zeigen Sie Prozesse an und verwalten sie interaktiv. In der Standardkonfiguration werden zwei relevante Spalten für Nice-Level und Prioritäten angezeigt. In der Spalte **NI** wird der Nice-Wert des Prozesses und in der Spalte **PR** die geplante Priorität angezeigt. In der **top**-Oberfläche ist der Nice-Level einer internen Prioritätswarteschlange des Systems zugeordnet, wie in der folgenden Grafik dargestellt. Zum Beispiel ist in der Spalte **PR** der Nice-Level -20 dem Wert 0 zugeordnet. In der Spalte **PR** ist der Nice-Level 19 der Priorität 39 zugeordnet.

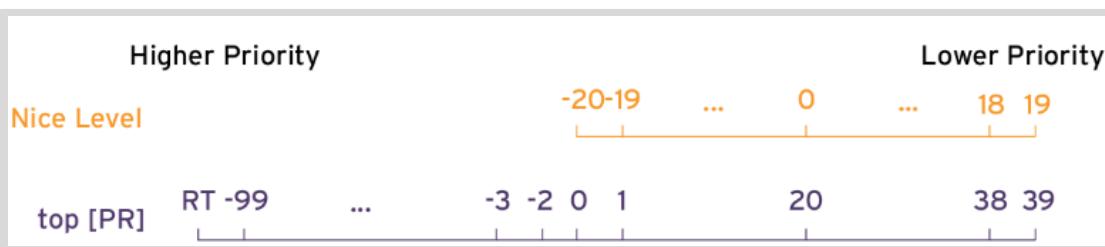


Abbildung 3.5: Von top angezeigte Nice-Level

Anzeigen von Nice-Leveln über die Befehlszeile

Der Befehl **ps** zeigt die Nice-Level von Prozessen an, jedoch nur, wenn die korrekten Formatierungsoptionen angegeben sind.

Der folgende **ps**-Befehl führt alle Prozesse mit PID, Prozessname, Nice-Level und Planungsklasse nach Nice-Level in absteigender Reihenfolge auf. Prozesse mit dem Eintrag **TS** in der Planungsklassenspalte **CLS** werden mit der Planungsrichtlinie **SCHED_NORMAL** ausgeführt. Prozesse mit einem Bindestrich (-) als Nice-Level werden mit anderen Planungsrichtlinien ausgeführt und vom Scheduler als höhere Priorität interpretiert. Details zu den zusätzlichen Planungsrichtlinien überschreiten den Rahmen dieses Kurses.

```
[user@host ~]$ ps axo pid,comm,nice,cls --sort=-nice
  PID COMMAND      NI  CLS
  30 khugepaged    19  TS
  29 ksmd         5   TS
  1 systemd        0   TS
  2 kthreadd       0   TS
  9 ksoftirqd/0    0   TS
 10 rcu_sched      0   TS
 11 migration/0    -  FF
 12 watchdog/0     -  FF
...output omitted...
```

Starten von Prozessen mit unterschiedlichen Nice-Leveln

Während der Prozesserstellung erbt ein Prozess den Nice-Level des übergeordneten Prozesses. Wenn ein Prozess in der Befehlszeile gestartet wird, erbt er den Nice-Level des Shell-Prozesses, in

Kapitel 3 | Tuning der Systemleistung

dem er gestartet wurde. Normalerweise führt dies zu neuen Prozessen, die mit einem Nice-Level von 0 ausgeführt werden.

Das folgende Beispiel startet einen Prozess von der Shell aus und zeigt den Nice-Wert des Prozesses an. Beachten Sie die Verwendung der PID-Option im Befehl **ps**, um die gewünschten Ausgabe anzugeben.

```
[user@host ~]$ sha1sum /dev/zero &
[1] 3480
[user@host ~]$ ps -o pid,comm,nice 3480
  PID COMMAND      NI
 3480 sha1sum      0
```

Der Befehl **nice** kann von allen Benutzern zum Starten von Befehlen mit einem Standard- oder einem höheren Nice-Level verwendet werden. Ohne Optionen startet der Befehl **nice** einen Prozess mit dem Standard-Nice-Wert 10.

Das folgende Beispiel startet den Befehl **sha1sum** als Hintergrundjob mit dem Standard-Nice-Level und zeigt den Nice-Level des Prozesses an:

```
[user@host ~]$ nice sha1sum /dev/zero &
[1] 3517
[user@host ~]$ ps -o pid,comm,nice 3517
  PID COMMAND      NI
 3517 sha1sum      10
```

Verwenden Sie die Option **-n**, um einen benutzerdefinierten Nice-Level für den zu startenden Prozess anzugeben. Standardmäßig wird 10 zum aktuellen Nice-Level des Prozesses hinzugefügt. Das folgende Beispiel startet einen Befehl als Hintergrundjob mit einem benutzerdefinierten Nice-Wert und zeigt den Nice-Level des Prozesses an:

```
[user@host ~]$ nice -n 15 sha1sum &
[1] 3521
[user@host ~]$ ps -o pid,comm,nice 3521
  PID COMMAND      NI
 3521 sha1sum      15
```



Wichtig

Unprivilegierte Benutzer können den Nice-Level nur vom aktuellen Wert auf maximal 19 erhöhen. Nach der Erhöhung können unprivilegierte Benutzer den Wert nicht wieder verringern, um zum vorherigen Level zurückzukehren. Der **root**-Benutzer kann den Nice-Level von jedem aktuellen Level auf ein Minimum von -20 verringern.

Ändern des Nice-Levels eines bestehenden Prozesses

Der Nice-Level eines bestehenden Prozesses kann mit dem Befehl **renice** geändert werden. In diesem Beispiel wird der PID-Bezeichner aus dem vorherigen Beispiel verwendet, um vom aktuellen Nice-Level 15 auf den gewünschten Nice-Level 19 zu wechseln.

```
[user@host ~]$ renice -n 19 3521
3521 (process ID) old priority 15, new priority 19
```

Der Nice-Level eines Prozesses kann auch mit dem Befehl **top** geändert werden. Wählen Sie in der interaktiven **top**-Oberfläche die Option **r**, um auf den Befehl **renice** zuzugreifen, und geben Sie die zu ändernde PID und den neuen Nice-Level ein.



Literaturhinweise

Manpages **nice(1)**, **renice(1)**, **top(1)** und **sched_setscheduler(2)**.

► Angeleitete Übung

Beeinflussen der Prozessplanung

In dieser Übung passen Sie die Planungspriorität von Prozessen mit den Befehlen **nice** und **renice** an und beobachten die Auswirkungen auf die Prozessausführung.

Ergebnisse

Sie sollten in der Lage sein, Planungsprioritäten für Prozesse anzupassen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab tuning-procscheduling start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **servera** im Netzwerk erreichbar ist.

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

- 1. Verwenden Sie auf **workstation** SSH, um sich bei **servera** als Benutzer **student** anzumelden. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Ermitteln Sie die Anzahl der CPU-Kerne auf **servera** und starten Sie dann zwei Instanzen des Befehls **sha1sum /dev/zero** & für jeden Kern.
- 2.1. Analysieren Sie mit **grep** die Anzahl der vorhandenen virtuellen Prozessoren (CPU-Kerne) aus der Datei **/proc/cpuinfo**.

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

- 2.2. Verwenden Sie einen Schleifenbefehl, um mehrere Instanzen des Befehls **sha1sum /dev/zero** & zu starten. Starten Sie pro im vorherigen Schritt gefundenen virtuellen Prozessor zwei Instanzen. In diesem Beispiel wären das vier Instanzen. Die PID-Werte in Ihrer Ausgabe können vom Beispiel abweichen.

```
[student@servera ~]$ for i in $(seq 1 4); do sha1sum /dev/zero & done  
[1] 2643  
[2] 2644  
[3] 2645  
[4] 2646
```

Kapitel 3 | Tuning der Systemleistung

- 3. Vergewissern Sie sich, dass die Hintergrundjobs für jeden **sha1sum**-Prozess ausgeführt werden.

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

- 4. Zeigen Sie mit den Befehlen **ps** und **pgrep** die prozentuale CPU-Auslastung für jeden **sha1sum**-Prozess an.

```
[student@servera ~]$ ps u $(pgrep sha1sum)
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
student   2643 49.8  0.0 228360  1744 pts/0      R   11:15  6:09 sha1sum /dev/zero
student   2644 49.8  0.0 228360  1780 pts/0      R   11:15  6:09 sha1sum /dev/zero
student   2645 49.8  0.0 228360  1748 pts/0      R   11:15  6:09 sha1sum /dev/zero
student   2646 49.8  0.0 228360  1780 pts/0      R   11:15  6:09 sha1sum /dev/zero
```

- 5. Beenden Sie alle **sha1sum**-Prozesse und vergewissern Sie sich dann, dass keine Jobs ausgeführt werden.

- 5.1. Beenden Sie mit dem Befehl **pkill** alle ausgeführten Prozesse mit dem Namensmuster **sha1sum**.

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

- 5.2. Stellen Sie sicher, dass keine Jobs ausgeführt werden.

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

- 6. Starten Sie mehrere Instanzen von **sha1sum /dev/zero &** und starten Sie dann eine weitere Instanz von **sha1sum /dev/zero &** mit einem Nice-Level von 10. Starten Sie mindestens so viele Instanzen wie das System über virtuelle Prozessoren verfügt. In diesem Beispiel werden 3 reguläre Instanzen und eine weitere mit dem höheren Nice-Level gestartet.

- 6.1. Verwenden Sie eine Schleife, um drei Instanzen von **sha1sum /dev/zero &** zu starten.

```
[student@servera ~]$ for i in $(seq 1 3); do sha1sum /dev/zero & done
[1] 1947
[2] 1948
[3] 1949
```

- 6.2. Starten Sie mit dem Befehl **nice** die vierte Instanz mit einem Nice-Level von 10.

Kapitel 3 | Tuning der Systemleistung

```
[student@servera ~]$ nice -n 10 sha1sum /dev/zero &
[4] 1953
```

- 7. Zeigen Sie mit den Befehlen **ps** und **pgrep** folgende Werte für jeden Prozess an: PID, prozentuale CPU-Auslastung, Nice-Wert und Name des ausgeführten Prozesses. Für die Instanz mit dem Nice-Wert 10 sollte eine niedrigere prozentuale CPU-Auslastung als für die anderen Instanzen angezeigt werden.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
  PID %CPU  NI COMMAND
1947 66.0   0 sha1sum
1948 65.7   0 sha1sum
1949 66.1   0 sha1sum
1953 6.7 10 sha1sum
```

- 8. Setzen Sie mit dem Befehl **sudo renice** den Nice-Level eines Prozesses aus dem vorherigen Schritt herab. Merken Sie sich den PID-Wert der Prozessinstanz mit dem Nice-Level 10. Verwenden Sie diese Prozess-PID, um ihren Nice-Level auf 5 zu verringern.

```
[student@servera ~]$ sudo renice -n 5 1953
[sudo] password for student:
1953 (process ID) old priority 10, new priority 5
```

- 9. Wiederholen Sie die Befehle **ps** und **pgrep**, um den CPU-Prozentsatz und den Nice-Level erneut anzuzeigen.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
  PID %CPU  NI COMMAND
1947 63.8   0 sha1sum
1948 62.8   0 sha1sum
1949 65.3   0 sha1sum
1953 9.1 5 sha1sum
```

- 10. Beenden Sie mit dem Befehl **pkill** alle ausgeführten Prozesse mit dem Namensmuster **sha1sum**.

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

- 11. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab tuning-procscheduling finish** aus, um diese Übung abzuschließen.

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

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Tuning der Systemleistung

Leistungscheckliste

In dieser praktischen Übung wenden Sie ein bestimmtes Tuning-Profil an und passen die Planungspriorität eines vorhandenen Prozesses mit hoher CPU-Auslastung an.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ein bestimmtes Tuning-Profil für ein Computersystem aktivieren
- CPU-Planungspriorität eines Prozesses anpassen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab tuning-review start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **serverb** im Netzwerk erreichbar ist.

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

1. Ändern Sie das aktuelle Tuning-Profil für **serverb** in **balanced**, ein allgemeines nicht spezialisiertes tuned-Profil.
2. Zwei Prozesse auf **serverb** lasten die CPU zu einem hohen Prozentsatz aus. Ändern Sie den **Nice**-Level dieser Prozesse in 10, um für andere Prozesse mehr CPU-Zeit zu ermöglichen.

Bewertung

Führen Sie auf **workstation** den Befehl **lab tuning-review grade** aus, um den Erfolg dieser praktischen Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab tuning-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Tuning der Systemleistung

Leistungscheckliste

In dieser praktischen Übung wenden Sie ein bestimmtes Tuning-Profil an und passen die Planungspriorität eines vorhandenen Prozesses mit hoher CPU-Auslastung an.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ein bestimmtes Tuning-Profil für ein Computersystem aktivieren
- CPU-Planungspriorität eines Prozesses anpassen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab tuning-review start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **serverb** im Netzwerk erreichbar ist.

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

1. Ändern Sie das aktuelle Tuning-Profil für **serverb** in **balanced**, ein allgemeines nicht spezialisiertes tuned-Profil.
 - 1.1. Öffnen Sie auf **workstation** als Benutzer **student** eine SSH-Sitzung zu **serverb**. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 1.2. Überprüfen Sie mit **yum**, ob das Paket **tuned** installiert ist.

```
[student@serverb ~]$ yum list tuned  
...output omitted...  
Installed Packages  
tuned.noarch 2.10.0-15.el8 @anaconda
```

- 1.3. Zeigen Sie mit dem Befehl **systemctl is-active tuned** den Status des Service **tuned** an.

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

Kapitel 3 | Tuning der Systemleistung

- 1.4. Listen Sie alle verfügbaren Tuning-Profile und ihre Beschreibungen auf. Beachten Sie, dass das aktuell aktive Profil **virtual-guest** ist.

```
[student@serverb ~]$ sudo tuned-adm list
[sudo] password for student: student
Available profiles:
- balanced           - General non-specialized tuned profile
- desktop            - Optimize for the desktop use-case
- 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
- 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. Ändern Sie das aktuell aktive Tuning-Profil in das Profil **balanced**.

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

- 1.6. Listen Sie zusammenfassende Informationen des aktuell aktiven tuned-Profils auf. Überprüfen Sie mit dem Befehl **tuned-adm profile_info**, ob das aktive Profil das Profil **balanced** ist.

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

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

2. Zwei Prozesse auf **serverb** lasten die CPU zu einem hohen Prozentsatz aus. Ändern Sie den **Nice**-Level dieser Prozesse in 10, um für andere Prozesse mehr CPU-Zeit zu ermöglichen.

- 2.1. Ermitteln Sie die Prozesse mit der höchsten CPU-Auslastung auf **serverb**. Die Prozesse mit der höchsten CPU-Auslastung werden in der Befehlsausgabe zuletzt aufgeführt. Die CPU-Prozentwerte können variieren.

```
[student@serverb ~]$ ps aux --sort=pcpu
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
...output omitted...
root      2983  100  0.0 228360  1744 ?          R<   21:08   0:23 md5sum /dev/zero
root      2967  101  0.0 228360  1732 ?          RN   21:08   0:23 sha1sum /dev/zero
[student@serverb ~]$
```

Kapitel 3 | Tuning der Systemleistung

- 2.2. Ermitteln Sie den aktuellen **Nice**-Level für jeden der beiden Prozesse mit der höchsten CPU-Auslastung.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm \
$(pgrep sha1sum;pgrep md5sum)
 PID %CPU NI COMMAND
2967 99.6 2 sha1sum
2983 99.7 -2 md5sum
```

- 2.3. Ändern Sie mit dem Befehl **sudo renice -n 10 2967 2983** den **Nice**-Level für jeden der beiden Prozesse in **10**. Verwenden Sie die PID-Werte, die in der vorherigen Befehlsausgabe ausgewiesen wurden.

```
[student@serverb ~]$ sudo renice -n 10 2967 2983
[sudo] password for student:
2967 (process ID) old priority 2, new priority 10
2983 (process ID) old priority -2, new priority 10
```

- 2.4. Vergewissern Sie sich, dass der aktuelle **Nice**-Level für jeden Prozess 10 lautet.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm \
$(pgrep sha1sum;pgrep md5sum)
 PID %CPU NI COMMAND
2967 99.6 10 sha1sum
2983 99.7 10 md5sum
```

- 2.5. Beenden Sie **serverb**.

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

Bewertung

Führen Sie auf **workstation** den Befehl **lab tuning-review grade** aus, um den Erfolg dieser praktischen Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab tuning-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Der Service **tuned** ändert automatisch Geräteeinstellungen, um bestimmte Systemanforderungen zu erfüllen, die einem vordefinierten, ausgewählten Tuning-Profil entsprechen.
- Um alle Änderungen rückgängig zu machen, die von einem ausgewählten Profil an den Systemeinstellungen vorgenommenen wurden, wechseln Sie entweder zu einem anderen Profil oder deaktivieren den Service **tuned**.
- Das System weist einem Prozess eine relative Priorität zu, um seinen CPU-Zugriff zu bestimmen. Diese Priorität wird als **nice**-Wert eines Prozesses bezeichnet.
- Der Befehl **nice** weist einem Prozess beim Start eine Priorität zu. Der Befehl **renice** ändert die Priorität eines ausgeführten Prozesses.

Kapitel 4

Steuern des Dateizugriffs mit ACLs

Ziel

Access Control Lists (ACLs, Zugriffssteuerungslisten) für Dateien interpretieren und festlegen, um komplexe Benutzer- und Gruppenzugriffsberechtigungen zu steuern

Ziele

- Anwendungsfälle für ACLs beschreiben, Dateien identifizieren, für die ACLs festgelegt sind, und die Auswirkungen dieser ACLs interpretieren
- ACLs für Dateien festlegen und entfernen sowie Default-ACLs definieren, die für neu erstellte Dateien automatisch durch ein Verzeichnis festgelegt werden

Abschnitte

- Interpretieren von Datei-ACLs (und angeleitete Übung)
- Sichern von Dateien mit ACLs (und angeleitete Übung)

Praktische Übung

Steuern des Dateizugriffs mit ACLs

Interpretieren von Datei-ACLs

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- ACLs und Optionen zum Mounten von Dateisystemen beschreiben
- ACLs mit **ls** und **getfacl** anzeigen und interpretieren
- Die ACL-Maske und die ACL-Berechtigungsrangfolge beschreiben
- Ermitteln, wo Red Hat Enterprise Linux standardmäßig ACLs verwendet

Konzepte für Access Control Lists

Linux-Standarddateiberechtigungen sind ausreichend, wenn Dateien nur von einem einzigen Eigentümer und einer einzigen, festgelegten Personengruppe verwendet werden. In bestimmten Anwendungsfällen ist es jedoch erforderlich, dass mehrere benannte Benutzer und Gruppen mit unterschiedlichen Dateiberechtigungssätzen auf Dateien zugreifen. *Access Control Lists (ACLs)* stellen diese Funktion bereit.

Mit ACLs können Sie mehreren Benutzern und Gruppen, die durch Benutzernamen, Gruppennamen, UID oder GID identifiziert werden, Berechtigungen erteilen, indem Sie dieselben Berechtigungsflags wie für reguläre Dateiberechtigungen verwenden: Lesen, Schreiben und Ausführen. Diese zusätzlichen Benutzer und Gruppen über den Dateieigentümer und die Gruppenzugehörigkeit der Datei hinaus werden als *benannte Benutzer* und *benannte Gruppen* bezeichnet, weil sie nicht in einer langen Liste, sondern innerhalb einer ACL namentlich aufgeführt sind.

Benutzer können ACLs für Dateien und Verzeichnisse festlegen, deren Eigentümer sie sind. Privilegierte Benutzer, denen die Linux-Fähigkeit **CAP_FOWNER** zugewiesen ist, können ACLs für jede Datei oder jedes Verzeichnis festlegen. Neue Dateien oder Unterverzeichnissen erben automatisch die ACL-Einstellungen der Default-ACL des übergeordneten Verzeichnisses, falls diese festgelegt sind. Ähnlich wie bei normalen Dateizugriffsregeln muss für die Hierarchie des übergeordneten Verzeichnisses mindestens die Berechtigung „Suchen“ (Ausführen) des Typs *sonstige* festgelegt sein, damit benannte Benutzer und Gruppen Zugriff erhalten können.

ACL-Unterstützung für das Dateisystem

Dateisysteme müssen mit der aktivierte ACL-Unterstützung gemountet sein. XFS-Dateisysteme verfügen über integrierte ACL-Unterstützung. Für andere Dateisysteme wie ext3 oder ext4, die auf Red Hat Enterprise Linux 8 erstellt wurden, ist die Option **acl** standardmäßig aktiviert. In früheren Versionen sollten Sie jedoch überprüfen, ob die ACL-Unterstützung aktiviert ist. Um die ACL-Unterstützung für das Dateisystem zu aktivieren, verwenden Sie die ACL-Option mit dem Befehl **mount** oder im Eintrag für das Dateisystem in der Konfigurationsdatei **/etc/fstab**.

Anzeigen und Interpretieren von ACL-Berechtigungen

Mit dem Befehl **ls -l** werden nur minimale Informationen zu ACL-Einstellungen ausgegeben:

```
[user@host content]$ ls -l reports.txt
-rwxrw----+ 1 user operators 130 Mar 19 23:56 reports.txt
```

Das Pluszeichen (+) am Ende der 10-stelligen Berechtigungszeichenfolge gibt an, dass eine erweiterte ACL-Struktur mit Einträgen in dieser Datei vorhanden ist.

user:

Zeigt die *Benutzer*-ACL-Einstellungen an, die den standardmäßigen *Benutzer*-Dateieinstellungen entsprechen: **rwx**.

group:

Zeigt die aktuellen Einstellungen der ACL-Maske an, nicht die Einstellungen des *Gruppeneigentümers*: **rw**.

other:

Zeigt die *sonstigen* ACL-Einstellungen an, die den standardmäßigen *sonstigen* Dateieinstellungen entsprechen: kein Zugriff.

**Wichtig**

Durch die Änderung der Gruppenberechtigungen für eine Datei mit einer ACL mithilfe von **chmod** werden nicht die Gruppeneigentümerberechtigungen, sondern die ACL-Maske verändert. Verwenden Sie den Befehl **setfacl -m g::perms file**, falls Sie die Gruppeneigentümerberechtigungen der Datei aktualisieren möchten.

Anzeigen von Datei-ACLS

Zum Anzeigen der ACL-Einstellungen für eine Datei verwenden Sie **getfacl file**:

```
[user@host content]$ getfacl reports.txt
# file: reports.txt
# owner: user
# group: operators
user::rwx
user:consultant3:---
user:1005:rwx      #effective:rwx-
group::rwx        #effective:rwx-
group:consultant1:r--
group:2210:rwx    #effective:rwx-
mask::rwx-
other::---
```

Sehen Sie sich jeden Abschnitt des vorherigen Beispiels an:

Kommentierte Einträge:

```
# file: reports.txt
# owner: user
# group: operators
```

Bei den ersten drei Zeilen handelt es sich um Kommentare, in denen der Dateiname, der Eigentümer (**user**) und der Gruppeneigentümer (**operators**) angegeben werden. Falls

Kapitel 4 | Steuern des Dateizugriffs mit ACLs

zusätzliche Dateiflags verwendet werden, wie zum Beispiel **setuid** oder **setgid**, wird eine vierte Kommentarzeile angezeigt, aus der ersichtlich ist, welche Flags festgelegt sind.

Einträge für „User“:

```
user::rwx          ①
user:consultant3:--- ②
user:1005:rwx      #effective:rw- ③
```

- ① Berechtigungen des Dateieigentümers. **user** verfügt über **rwx**.
- ② Berechtigungen von benannten Benutzern. Ein Eintrag für jeden mit dieser Datei verknüpften benannten Benutzer. **consultant3** hat *keine* Berechtigungen.
- ③ Berechtigungen von benannten Benutzern. UID **1005** hat die Berechtigungen **rwx**. Durch die Maske werden diese jedoch effektiv auf **rw** beschränkt.

Einträge für „Group“:

```
group::rwx          #effective:rw- ①
group:consultant1:r-- ②
group:2210:rwx      #effective:rw- ③
```

- ① Berechtigungen von Gruppeneigentümern. **operators** hat die Berechtigungen **rwx**. Durch die Maske werden diese jedoch effektiv auf **rw** beschränkt.
- ② Berechtigungen von benannten Gruppen. Ein Eintrag für jede mit dieser Datei verknüpfte benannte Gruppe. **consultant1** hat nur die Berechtigung **r**.
- ③ Berechtigungen von benannten Gruppen. GID **2210** hat die Berechtigungen **rwx**. Durch die Maske werden diese jedoch effektiv auf **rw** beschränkt.

Eintrag für „Mask“:

```
mask::rw-
```

Die Maskeneinstellungen zeigen die maximal möglichen Berechtigungen für alle benannten Benutzer, den Gruppeneigentümer und benannte Gruppen. Obwohl für jeden Eintrag die Ausführungsberechtigung festgelegt ist, können UID **1005**, **operators** und GID **2210** diese Datei nicht ausführen.

Einträge für „Other“:

```
other::---
```

Sonstige oder „world“-Berechtigungen. Alle anderen UIDs und GIDs haben KEINE Berechtigungen.

Anzeigen von Verzeichnis-ACLs

Zum Anzeigen der ACL-Einstellungen für ein Verzeichnis verwenden Sie den Befehl **getfac1 directory**:

```
[user@host content]$ getfac1 .
# file: .
# owner: user
# group: operators
```

```
# flags: -s-
user::rwx
user:consultant3:---
user:1005:rwx
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant3:---
default:group::rwx
default:group:consultant1:r-x
default:mask::rwx
default:other::---
```

Sehen Sie sich jeden Abschnitt des vorherigen Beispiels an:

Öffnen von Kommentareinträgen:

```
# file: .
# owner: user
# group: operators
# flags: -s-
```

Bei den ersten drei Zeilen handelt es sich um Kommentare, in denen der Verzeichnisname, der Eigentümer (**user**) und der Gruppeneigentümer (**operators**) angegeben werden. Falls zusätzliche Verzeichnisflags verwendet werden (**setuid**, **setgid**, **sticky**), wird eine vierte Kommentarzeile angezeigt, aus der ersichtlich ist, welche Flags festgelegt sind. In diesem Fall **setgid**.

Standard-ACL-Einträge:

```
user::rwx
user:consultant3:---
user:1005:rwx
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
```

Die ACL-Berechtigungen für dieses Verzeichnis sind dieselben wie im vorherigen Dateibeispiel. Sie gelten jedoch für das Verzeichnis. Der wichtigste Unterschied ist der Einbezug der Ausführungsberechtigung für diese Einträge (falls angemessen). Auf diese Weise kann die Berechtigung zum Durchsuchen des Verzeichnisses erteilt werden.

Standardeinträge für „User“:

```
default:user::rwx ①
default:user:consultant3:--- ②
```

Kapitel 4 | Steuern des Dateizugriffs mit ACLs

- ① Default-ACL-Berechtigung von Dateieigentümern. Der Dateieigentümer erhält die Berechtigungen **rwx**, „Lesen/Schreiben“ für neue Dateien und „Ausführen“ für neue Unterverzeichnisse.
- ② Default-ACL-Berechtigungen für benannte Benutzer. Ein Eintrag für jeden benannten Benutzer, bei dem automatisch die Default-ACL auf neue Dateien oder Unterverzeichnisse angewendet wird. **consultant3** hat standardmäßig *keine* Berechtigungen.

Default-Einträge für „Group“:

```
default:group::rwx          ①  
default:group:consultant1:r-x ②
```

- ① Default-ACL-Berechtigungen für Gruppeneigentümer. Der Gruppeneigentümer der Datei erhält die Berechtigungen **rwx**, „Lesen/Schreiben“ für neue Dateien und „Ausführen“ für neue Unterverzeichnisse.
- ② Default-ACL-Berechtigungen für benannte Gruppen. Ein Eintrag für jede benannte Gruppe, der automatisch die Default-ACL zugewiesen wird. **consultant1** erhält die Berechtigungen **rx**, „Nur Lesen“ für neue Dateien und „Ausführen“ für neue Unterverzeichnisse.

Default-Eintrag für „ACL-Maske“:

```
default:mask::rwx
```

Die Default-Maskeneinstellungen zeigen die anfangs maximal möglichen Berechtigungen für alle neu erstellten Dateien und Verzeichnisse, die über ACLs für benannte Benutzer, Gruppeneigentümer oder benannte Gruppen verfügen: Lese- und Schreibberechtigung für neue Dateien sowie Ausführungsberrechtigung für neue Unterverzeichnisse. Ausführungsberrechtigungen werden niemals für neue Dateien erteilt.

Default-Eintrag für „Other“:

```
default:other::---
```

Default-Berechtigungen für *other* oder „world“. Alle anderen UIDs und GIDs haben *keine* Berechtigungen für neue Dateien oder neue Unterverzeichnisse.

Die **default**-Einträge im vorherigen Beispiel umfassen nicht den benannten Benutzer (UID **1005**) und die benannte Gruppe (GID **2210**); somit werden bei diesen für neue Dateien und neue Unterverzeichnisse nicht automatisch anfängliche ACL-Einträge hinzugefügt. Dadurch sind sie auf Dateien und Unterverzeichnisse beschränkt, für die sie bereits über ACLs verfügen. Außerdem kann der Dateieigentümer die ACL später mithilfe von **setfac1** hinzufügen. Sie können jedoch weiterhin eigene Dateien und Unterverzeichnisse erstellen.



Anmerkung

Die Ausgabe des Befehls **getfac1** kann als Eingabe für **setfac1** dienen, um ACLs wiederherzustellen oder aus einer Quelldatei oder einem Quellverzeichnis zu kopieren und in einer neuen Datei zu speichern. Zum Beispiel: Verwenden Sie zum Wiederherstellen von ACLs aus einem Backup **getfac1 -R /dir1 > file1**, um eine rekursive ACL-Ausgabe-Dumpdatei für das Verzeichnis und die zugehörigen Inhalte zu erstellen. Die Ausgabe kann anschließend für die Wiederherstellung der ursprünglichen ACLs durch Übergabe der gespeicherten Ausgabe als Eingabe an den Befehl **setfac1** verwendet werden. Um beispielsweise eine Massenaktualisierung desselben Verzeichnisses im aktuellen Pfad durchzuführen, verwenden Sie den folgenden Befehl: **setfac1 --set-file=file1**

Die ACL-Maske

Die ACL-Maske legt die maximale Anzahl an Berechtigungen fest, die Sie benannten Benutzern, dem Gruppeneigentümer und benannten Gruppen erteilen können. Die Berechtigungen des Dateieigentümers oder **sonstiger** Benutzer werden durch die ACL-Maske nicht eingeschränkt. Alle Dateien und Verzeichnisse, die ACLs integrieren, verfügen über eine ACL-Maske.

Die Maske kann mit **getfac1** angezeigt und mit **setfac1** genau festgelegt werden. Falls sie nicht genau festgelegt ist, wird sie automatisch berechnet und hinzugefügt. Es können jedoch auch die Default-Maskeneinstellungen eines übergeordneten Verzeichnisses vererbt werden. Sobald eine der betroffenen ACLs hinzugefügt, verändert oder gelöscht wird, wird die Maske standardmäßig neu berechnet.

ACL-Berechtigungsrangfolge

Bei der Entscheidung, ob ein Prozess (d. h. ein laufendes Programm) auf eine Datei zugreifen darf, werden die Dateiberechtigungen und ACLs wie folgt angewendet:

- Wenn der Prozess vom Dateieigentümer ausgeführt wird, greifen die Benutzer-ACL-Berechtigungen der Datei.
- Wenn der Prozess von einem Benutzer ausgeführt wird, der in einem ACL-Eintrag für benannte Benutzer aufgeführt ist, greifen die ACL-Berechtigungen für benannte Benutzer (solange von der Maske zugelassen).
- Wenn der Prozess von einer Gruppe, die dem Gruppeneigentümer der Datei entspricht, oder einer Gruppe mit einem ausdrücklichen ACL-Eintrag für benannte Gruppen ausgeführt wird, gelten die entsprechenden ACL-Berechtigungen (solange von der Maske zugelassen).
- Andernfalls greifen die *sonstigen* ACL-Berechtigungen der Datei.

Beispiele für die Verwendung der ACL durch das Betriebssystem

Red Hat Enterprise Linux enthält Beispiele, die die typische ACL-Verwendung für erweiterte Berechtigungsanforderungen demonstrieren.

ACLs für Systemd-Journaldateien

systemd-journald verwendet ACL-Einträge, um den Lesezugriff auf die Datei **/run/log/journal/cb44...8ae2/system.journal** für die Gruppen **adm** und **wheel** zuzulassen. Diese ACL ermöglicht den Mitgliedern der Gruppen **adm** und **wheel** den Lesezugriff auf die von

journald verwalteten Protokolle, ohne dass den privilegierten Inhalten in **/var/log/**, wie **messages**, **secure** oder **audit**, besondere Berechtigungen erteilt werden müssen.

Aufgrund der **systemd-journald**-Konfiguration kann sich der übergeordnete Ordner der Datei **system.journal** ändern, aber **systemd-journald** wendet ACLs automatisch auf den neuen Ordner und die neue Datei an.



Anmerkung

Wenn **systemd-journald** für die Verwendung von persistentem Speicher konfiguriert ist, sollten Systemadministratoren eine ACL für den Ordner **/var/log/journal/** festlegen.

```
[user@host ]$ getfacl /run/log/journal/cb44...8ae2/system.journal
getfacl: Removing leading '/' from absolute path names
# file: run/log/journal/cb44...8ae2/system.journal
# owner: root
# group: systemd-journal
user::rw-
group::r--
group:adm:r--
group:wheel:r--
mask::r--
other::---
```

ACL auf von Systemd verwalteten Geräten

systemd-udev verwendet eine Reihe von **udev**-Regeln, die das Tag **uaccess** für bestimmte Geräte ermöglichen, z. B. CD/DVD-Player oder -Brenner, USB-Speichergeräte, Soundkarten und viele andere. Die zuvor erwähnten **udev**-Regeln legen ACLs für diese Geräte fest, um bei einer grafischen Benutzeroberfläche angemeldeten Benutzern (zum Beispiel **gdm**) die volle Kontrolle über diese Geräte zu ermöglichen.

Die ACLs bleiben aktiv, bis sich der Benutzer von der GUI abmeldet. Der nächste Benutzer, der sich bei der GUI anmeldet, erhält eine neue ACL für die erforderlichen Geräte.

Im folgenden Beispiel hat **user** einen ACL-Eintrag mit den Berechtigungen **rw** für das Gerät **/dev/sr0** (CD/DVD-Laufwerk).

```
[user@host ]$ getfacl /dev/sr0
getfacl: Removing leading '/' from absolute path names
# file: dev/sr0
# owner: root
# group: cdrom
user::rw-
user:group:rw-
group::rw-
mask::rw-
other::---
```



Literaturhinweise

Manpages **acl(5)**, **getfacl(1)**, **journald.conf(5)**, **ls(1)**, **systemd-journald(8)** und **systemd-udevd(8)**

► Quiz

Interpretieren von Datei-ACLs

Ordnen Sie die nachstehenden Elemente ihren Entsprechungen in der Tabelle zu.

default:m::rx /directory

default:user:mary:rx /directory

g::rw /directory

g::rw file

getfacl /directory

group:hug:rwx /directory

user::rx file

user:mary:rx file

Beschreibung	ACL-Vorgang
Die ACL für ein Verzeichnis anzeigen	
Benannter Benutzer mit Lese- und Ausführungsberechtigungen für eine Datei	
Dateieigentümer mit Lese- und Ausführungsberechtigungen für eine Datei	
Lese- und Schreibberechtigungen für ein Verzeichnis, die dem Gruppeneigentümer des Verzeichnisses erteilt wurden	
Lese- und Schreibberechtigungen für eine Datei, die dem Gruppeneigentümer der Datei erteilt wurden	
Lese-, Schreib- und Ausführungsberechtigungen für ein Verzeichnis, die einer benannten Gruppe erteilt wurden	
Lese- und Ausführungsberechtigungen, die als Default-Maske festgelegt sind	
Benannter Benutzer, dem eine anfängliche Leseberechtigung für neue Dateien und eine Lese- und Ausführungsberechtigung für neue Unterverzeichnisse erteilt wurde	

► Lösung

Interpretieren von Datei-ACLs

Ordnen Sie die nachstehenden Elemente ihren Entsprechungen in der Tabelle zu.

Beschreibung	ACL-Vorgang
Die ACL für ein Verzeichnis anzeigen	getfacl /directory
Benannter Benutzer mit Lese- und Ausführungsberechtigungen für eine Datei	user:mary:rx file
Dateieigentümer mit Lese- und Ausführungsberechtigungen für eine Datei	user::rx file
Lese- und Schreibberechtigungen für ein Verzeichnis, die dem Gruppeneigentümer des Verzeichnisses erteilt wurden	g::rw /directory
Lese- und Schreibberechtigungen für eine Datei, die dem Gruppeneigentümer der Datei erteilt wurden	g::rw file
Lese-, Schreib- und Ausführungsberechtigungen für ein Verzeichnis, die einer benannten Gruppe erteilt wurden	group:hug:rwx /directory
Lese- und Ausführungsberechtigungen, die als Default-Maske festgelegt sind	default:m::rx /directory
Benannter Benutzer, dem eine anfängliche Leseberechtigung für neue Dateien und eine Lese- und Ausführungsberechtigung für neue Unterverzeichnisse erteilt wurde	default:user:mary:rx /directory

Sichern von Dateien mit ACLs

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Reguläre ACL-Dateiberechtigungen mit **setfac1** ändern
- Default-ACL-Dateiberechtigungen für neue Dateien und Verzeichnisse steuern

Ändern von ACL-Dateiberechtigungen

Verwenden Sie **setfac1**, um Standard-ACLs für Dateien und Verzeichnisse hinzuzufügen, zu löschen und zu ändern.

ACLs nutzen die normale Dateisystemdarstellung von Berechtigungen: „**r**“ für Leseberechtigung (read), „**w**“ für Schreibberechtigung (write) und „**x**“ für Ausführungsberechtigung (execute). Ein „-“ (Bindestrich) zeigt an, dass die entsprechende Berechtigung nicht vorhanden ist. Wenn ACLs (rekursiv) gesetzt werden, kann ein Großbuchstabe „**X**“ verwendet werden, um anzugeben, dass die Ausführungsberechtigung nur für Verzeichnisse und nicht für normale Dateien gesetzt werden sollte, es sei denn, die Datei hat bereits die entsprechende Ausführungsberechtigung. Das Verhalten ist das gleiche wie bei **chmod**.

Hinzufügen und Ändern von ACLs

ACLs können über die Befehlszeile mit der Option **-m** festgelegt oder über eine Datei mit der Option **-M** übergeben werden (verwenden Sie einen „-“ (Bindestrich) anstelle eines Dateinamens für **stdin**). Diese zwei Optionen dienen zum Ändern von ACLs. Damit können Sie neue ACL-Einträge hinzufügen oder bestimmte vorhandene ACL-Einträge für ein Verzeichnis oder eine Datei ersetzen. Alle weiteren vorhandenen ACL-Einträge für die Datei oder das Verzeichnis bleiben unverändert bestehen.



Anmerkung

Verwenden Sie die Optionen **--set** oder **--set-file**, um die ACL-Einstellungen für eine Datei vollständig zu ersetzen.

Wenn bei der erstmaligen Definition einer ACL für eine Datei im Hinzufügenvorgang keine Einstellungen für *Dateieigentümer*, *Gruppeneigentümer* oder *sonstige* Berechtigungen angegeben sind, werden diese anhand der aktuellen Standarddateiberechtigungen (auch als *Basis-ACL*-Einträge bezeichnet, die nicht geändert werden können) festgelegt und ein neuer *Masken*-Wert wird berechnet und hinzugefügt.

Hinzufügen oder Ändern einer ACL für einen *Benutzer* oder *benannten Benutzer*:

```
[user@host ~]$ setfac1 -m u:name:rX file
```

Wenn *name* leer gelassen wird, bezieht er sich auf den *Dateieigentümer*. Andernfalls kann *name* ein Benutzername oder ein UID-Wert sein. In diesem Beispiel wird ausschließlich eine Leseberechtigung und, falls schon festgelegt, eine Ausführungsberechtigung gewährt. (Sofern es

Kapitel 4 | Steuern des Dateizugriffs mit ACLs

sich bei *file* nicht um ein Verzeichnis handelt. In diesem Fall würde die Ausführungsberechtigung für das Verzeichnis gesetzt werden, um Verzeichnissuchen zu ermöglichen).

Dateieigentümer- und Standard-*Dateieigentümer-ACL*-Berechtigungen sind absolut gleichwertig. Wird **chmod** auf die *Dateieigentümer*-Berechtigungen angewendet, hat dies die gleichen Auswirkungen wie die Anwendung von **setfac1** auf die *Dateieigentümer*-Berechtigungen. **chmod** hat keine Auswirkungen auf benannte Benutzer.

Hinzufügen oder Ändern einer ACL für eine Gruppe oder benannte Gruppe:

```
[user@host ~]$ setfac1 -m g:name:rw file
```

Bei diesem Vorgang können Sie dem gleichen Muster wie beim Hinzufügen oder Ändern eines Benutzer-ACL-Eintrags folgen. Wenn *name* leer gelassen wird, bezieht er sich auf den *Gruppeneigentümer*. Andernfalls sollten Sie einen Gruppennamen oder GID-Wert für eine *benannte Gruppe* angeben. Die Berechtigungen in diesem Beispiel würden Lese- und Schreibzugriff gewähren.

chmod hat für Gruppenberechtigungen mit ACL-Einstellungen keine Auswirkungen. Allerdings wird durch den Befehl die ACL-Maske aktualisiert.

Hinzufügen oder Ändern einer ACL für sonstige Benutzer:

```
[user@host ~]$ setfac1 -m o:::- file
```

Für *sonstige* können ausschließlich Berechtigungseinstellungen angegeben werden. Typische Berechtigungseinstellungen für sonstige Benutzer sind: keine Berechtigungen, die mit einem Bindestrich (-) festgelegt werden, und Leseberechtigungen, die wie üblich mit **r** angegeben werden. Natürlich können Sie beliebige Standardberechtigungen festlegen.

Die ACL-Berechtigungen *sonstige* und die Standardberechtigungen *sonstige* sind absolut gleichwertig. Die Anwendung von **chmod** auf die Berechtigung *sonstiger* Benutzer hat somit die gleiche Auswirkung wie die Anwendung von **setfac1** auf die Berechtigung der *sonstigen* Benutzer.

Sie können mehrere Einträge mit demselben Befehl hinzufügen. Verwenden Sie dazu eine durch Komma getrennte Eintragsliste:

```
[user@host ~]$ setfac1 -m u::rwx,g:consultants:rX,o:::- file
```

Der *Dateieigentümer* erhält dadurch Lese-, Schreib- und Ausführungsberechtigungen, die benannte Gruppe **consultants** Lese- und bedingte Ausführungsberechtigungen und alle **sonstigen** Benutzer erhalten *keine* Berechtigungen. Der *Gruppeneigentümer* behält seine bestehenden Datei- oder ACL-Berechtigungen und die anderen „benannten“ Einträge bleiben unverändert.

Verwenden von **getfac1** als Eingabe

Sie können die Ausgabe von **getfac1** als Eingabe für **setfac1** verwenden:

```
[user@host ~]$ getfac1 file-A | setfac1 --set-file=- file-B
```

Die Option **--set-file** akzeptiert Eingaben aus einer Datei oder von *stdin*. Der Bindestrich (-) legt die Verwendung von *stdin* fest. In diesem Fall hat *file-B* die gleichen ACL-Einstellungen wie *file-A*.

Festlegen einer expliziten ACL-Maske

Sie können eine ACL-Maske explizit für eine Datei oder ein Verzeichnis festlegen, um die effektive Maximalanzahl an Berechtigungen für benannte Benutzer, den Gruppeneigentümer und benannte Gruppen zu beschränken. Dadurch werden sämtliche Berechtigungen, die über die Maske hinausgehen, beschränkt, während Berechtigung, die restiktiver sind als die Maske, unverändert bleiben.

```
[user@host ~]$ setfacl -m m::r file
```

Durch diesen Befehl wird ein Maskenwert hinzugefügt, der die Berechtigungen von allen benannten Benutzern, dem Gruppeneigentümer und sämtlichen benannten Gruppen unabhängig von den vorhandenen Einstellungen auf Lesezugriff beschränkt. Der Dateieigentümer und die **sonstigen** Benutzer sind von dieser Maskeneinstellung nicht betroffen.

Mit **getfacl** wird ein **effektiver** Kommentar neben Einträgen angezeigt, die durch eine Maskeneinstellung beschränkt sind.



Wichtig

Sobald eine der betroffenen ACL-Einstellungen (benannte Benutzer, Gruppeneigentümer oder benannte Gruppen) verändert oder gelöscht wird, wird die ACL-Maske standardmäßig neu berechnet. Dadurch wird unter Umständen eine vorherige explizite Maskeneinstellung zurückgesetzt.

Um zu verhindern, dass die Maske neu berechnet wird, verwenden Sie die Option **-n** oder fügen Sie eine Maskeneinstellung (**-m m::perms**) bei jedem **setfacl**-Vorgang hinzu, bei dem ACL-Maskeneinstellungen verändert werden.

Rekursive ACL-Änderungen

Wenn Sie eine ACL für ein Verzeichnis festlegen, verwenden Sie die Option **-R** zum rekursiven Anwenden der ACL. Denken Sie daran, bei der Rekursion möglichst die „X“-Berechtigung (großes X) zu verwenden. Dateien mit Ausführungsberechtigung können auf diese Weise ihre Einstellung beibehalten, während Verzeichnisse Ausführungsberechtigungen erhalten, sodass Verzeichnisse durchsucht werden können. Die Verwendung des Großbuchstabens „X“ ist auch für das nicht rekursive Festlegen von ACLs bewährte Praxis, da Administratoren auf diese Weise Ausführungsberechtigungen für eine reguläre Datei nicht versehentlich hinzufügen können.

```
[user@host ~]$ setfacl -R -m u:name:rX directory
```

Mit diesem Befehl wird der Name des Benutzers dem Verzeichnis und allen vorhandenen Dateien und Unterverzeichnissen hinzugefügt und Leseberechtigungen sowie bedingte Ausführungsberechtigungen erteilt.

Löschen von ACLs

Das Löschen bestimmter ACL-Einträge erfolgt nach dem gleichen Muster wie der Änderungsvorgang. Allerdings sollten dabei für „:perms“ keine Angaben gemacht werden.

```
[user@host ~]$ setfacl -x u:name,g:name file
```

Durch diesen Befehl werden nur der benannte Benutzer und die benannte Gruppe aus der Datei- oder Verzeichnis-ACL gelöscht. Alle anderen vorhandenen ACL-Einträge blieben aktiv.

Sie können den Lösch- (**-x**) und den Änderungsvorgang (**-m**) im gleichen **setfacl**-Vorgang verwenden.

Die Maske kann nur gelöscht werden, wenn keine anderen ACLs festgelegt sind (exklusive der *Basis-ACL*, die nicht gelöscht werden kann), und muss somit zuletzt gelöscht werden. Die Datei weist nach der Löschung keine ACLs mehr auf und **ls -l** zeigt kein Pluszeichen (+) neben der Berechtigungszeichenfolge an. Alternativ können Sie mit dem folgenden Befehl alle ACL-Einträge für eine Datei oder ein Verzeichnis (einschließlich der *Default-ACL* für Verzeichnisse) löschen:

```
[user@host ~]$ setfacl -b file
```

Steuern von Default-ACL-Dateiberechtigungen

Um sicherzustellen, dass Dateien und Verzeichnisse, die in einem Verzeichnis erstellt werden, bestimmte ACLs erben, verwenden Sie die *Default-ACL* für ein Verzeichnis. Sie können eine *Default-ACL* und beliebige Standard-ACL-Einstellungen, einschließlich einer Default-Maske, festlegen.

Für die Zugriffssteuerung erfordert das Verzeichnis selbst aber in jedem Fall noch Standard-ACLs, da *Default-ACLs* keine Zugriffsteuerung für das Verzeichnis implementieren, sondern ausschließlich Unterstützung für die Vererbung von ACL-Berechtigungen bieten. Zum Beispiel:

```
[user@host ~]$ setfacl -m d:u:username:rx directory
```

Durch diesen Befehl wird ein benannter Default-Benutzer (**d:u:*name***) mit Leseberechtigung und Ausführungsberechtigung für Unterverzeichnisse hinzugefügt.

Der Befehl **setfacl** zum Hinzufügen einer *Default-ACL* für jeden der ACL-Typen entspricht genau dem für Standard-ACLs, allerdings mit einem zusätzlich vorangestellten **d:**. Alternativ können Sie die Option **-d** in der Befehlszeile verwenden.



Wichtig

Stellen Sie beim Setzen von *Default-ACLs* für ein Verzeichnis sicher, dass Benutzer auf Inhalte in neu erstellten Unterverzeichnissen zugreifen können, indem Sie die Ausführungsberechtigung mit in die *Default-ACL* aufnehmen.

Benutzer erhalten nicht automatisch Ausführungsberechtigungen für neu erstellte, reguläre Dateien, da im Gegensatz zu neuen Verzeichnissen die ACL-Maske einer neuen regulären Datei **rw-** lautet.



Anmerkung

Neue Dateien und Unterverzeichnisse erhalten ihre Eigentümer-UID und die GID-Werte der primären Gruppe weiterhin von dem Benutzer, der diese erstellt. Ist jedoch das **setgid**-Flag des übergeordneten Verzeichnisses aktiviert, entspricht der GID-Wert der primären Gruppe dem GID-Wert des übergeordneten Verzeichnisses.

Löschen von Default-ACL-Einträgen

Löschen Sie eine *Default*-ACL genauso, wie Sie eine Standard-ACL löschen und stellen Sie ein **d**: voran oder verwenden Sie die Option **-d**.

```
[user@host ~]$ setfacl -x d:u:name directory
```

Mit diesem Befehl wird der **Default**-ACL-Eintrag gelöscht, der im vorherigen Beispiel hinzugefügt wurde.

Zum Löschen aller *Default*-ACL-Einträge eines Verzeichnisses verwenden Sie **setfacl -k directory**.



Literaturhinweise

Manpages **acl(5)**, **setfacl(1)** und **getfacl(1)**

► Angeleitete Übung

Sichern von Dateien mit ACLs

In dieser Übung erteilen Sie mit ACL-Einträgen einer Gruppe Zugriff auf ein Verzeichnis und verweigern einem Benutzer den Zugriff, legen die Default-ACL für ein Verzeichnis fest und überprüfen, ob die in diesem Verzeichnis erstellten neuen Dateien die Default-ACL erben.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Mit ACL-Einträgen einer Gruppe Zugriff erteilen und einem ihrer Mitglieder den Zugriff verweigern
- Überprüfen, ob die vorhandenen Dateien und Verzeichnisse die neuen ACL-Berechtigungen berücksichtigen
- Die Default-ACL für ein Verzeichnis festlegen und sicherstellen, dass neue Dateien und Verzeichnisse ihre Konfiguration erben

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab acl-secure start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem werden die in dieser Übung verwendeten Benutzer, Gruppen, Verzeichnisse und Dateien erstellt.

```
[student@workstation ~]$ lab acl-secure start
```

„Operators“ und „Consultants“ sind Mitglieder eines IT-Support-Unternehmens. Sie müssen mit dem Informationsaustausch beginnen. **servera** enthält ein ordnungsgemäß konfiguriertes Freigabeverzeichnis unter **/shares/content**, das Dateien hostet.

Derzeit haben nur Mitglieder der Gruppe **operators** Zugriff auf dieses Verzeichnis, aber Mitglieder der Gruppe **consultants** benötigen ebenfalls vollen Zugriff auf dieses Verzeichnis.

Der Benutzer **consultant1** ist Mitglied der Gruppe **consultants**, hat jedoch in vielen Fällen Probleme verursacht, sodass dieser Benutzer keinen Zugriff auf das Verzeichnis haben sollte.

Ihre Aufgabe besteht darin, dem Verzeichnis und seinen Inhalten die entsprechenden ACL-Einträge hinzuzufügen, sodass Mitglieder der Gruppe **consultants** vollständigen Zugriff haben. Dem Benutzer **consultant1** soll der Zugriff jedoch verwehrt werden. Sorgen Sie dafür, dass auf zukünftige Dateien und Verzeichnisse, die in **/shares/content** gespeichert werden, die richtigen ACL-Einträge angewendet werden.

Wichtige Informationen:

- Die Benutzer **sysadmin1** und **operator1** sind Mitglieder der Gruppe **operators**.
- Die Benutzer **consultant1** und **consultant2** sind Mitglieder der Gruppe **consultants**.

- Das Verzeichnis **/shares/content** enthält das Unterverzeichnis **server-info** und zahlreiche Dateien zum Testen der ACL. Das Verzeichnis **/shares/content** enthält auch ein ausführbares Skript, **loadvg.sh**, das Sie zum Testen verwenden können.
- Die Passwörter der Benutzer **sysadmin1**, **operator1**, **consultant1** und **consultant2** lauten **redhat**.
- Alle Änderungen sollten am Verzeichnis **/shares/content** und an den enthaltenen Dateien vorgenommen werden. Nehmen Sie keine Änderungen am Verzeichnis **/shares** vor.

► 1. Melden Sie sich bei **servera** an und wechseln Sie zum Benutzer **root**.

- Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

► 2. Fügen Sie dem Verzeichnis **/shares/content** und all seinen Inhalten die benannte ACL hinzu.

- Aktualisieren Sie mit **setfacl** das Verzeichnis **/shares/content** rekursiv und erteilen Sie der Gruppe **consultants** Lese-, Schreib- und bedingte Ausführungsberechtigungen.

```
[root@servera ~]# setfacl -Rm g:consultants:rwx /shares/content
```

Die Option **-R** bedeutet rekursiv, die Option **-m** bedeutet Ändern/Hinzufügen, **rwx** bedeutet, Lese-, Schreib- und bedingte Ausführungsberechtigungen anwenden.

- Aktualisieren Sie mit **setfacl** das Verzeichnis **/shares/content** rekursiv und verweigern Sie dem Benutzer **consultant1** aus der Gruppe **consultants** jeden Zugriff.

```
[root@servera ~]# setfacl -Rm u:consultant1:- /shares/content
```

Die Option **-R** bedeutet rekursiv, die Option **-m** bedeutet Ändern/Hinzufügen, **-** bedeutet keinen Zugriff gewähren.

► 3. Fügen Sie die benannte ACL als *Default*-ACL hinzu, um das Hinzufügen zukünftiger Dateien und Verzeichnisse zu unterstützen.

- Fügen Sie mit **setfacl** eine Default-Zugriffsregel für die Gruppe **consultants** hinzu. Erteilen Sie dem Verzeichnis **content** Lese-, Schreib- und Ausführungsberechtigungen.

```
[root@servera ~]# setfacl -m d:g:consultants:rwx /shares/content
```

Die Option **-m** bedeutet Ändern/Hinzufügen, **d:g** bedeutet Standardgruppe, **rwx** bedeutet Lese-/Schreib-/Ausführungsberechtigungen anwenden (erforderlich für die ordnungsgemäße Erstellung von Unterverzeichnissen und den Zugriff auf diese).

- 3.2. Fügen Sie mit **setfacl** eine Default-Zugriffsregel für den Benutzer **consultant1** hinzu. Verwehren Sie sämtlichen Zugriff auf das Verzeichnis **content**.

```
[root@servera ~]# setfacl -m d:u:consultant1:- /shares/content
```

Die Option **-m** bedeutet Ändern/Hinzufügen, **d:u** bedeutet Default-Benutzer, **-** bedeutet keine Berechtigungen.

- 4. Überprüfen Sie Ihre ACL-Änderungen.

consultant2 sollte sämtliche Dateien lesen und neue Verzeichnisse mit einer neuen Datei darin erstellen können.

consultant1 sollte keine Dateien lesen, schreiben oder ausführen können. Dies schließt auch die Auflistung der Verzeichnisinhalte ein.

Verwenden Sie **su - user**, um zu Ihren Testbenutzern zu wechseln. Verlassen Sie mit **exit** oder **Strg+D** die Testbenutzer-Shell.

```
[root@servera ~]# exit  
[student@servera ~]$ su - consultant2  
Password: redhat  
[consultant2@servera ~]$ cd /shares/content/
```

- 4.1. Überprüfen Sie mit **cat**, ob **consultant2** eine Datei lesen kann.

```
[consultant2@servera content]$ cat serverb-loadavg.txt  
#####  
serverb.lab.example.com  
#####  
Wed Mar 25 15:25:19 EDT 2019  
#####  
ldavg 0.18, 0.06, 0.05  
#####
```

- 4.2. Überprüfen Sie mit dem Skript **loadavg.sh**, ob **consultant2** eine Datei ausführen kann.

```
[consultant2@servera content]$ ./loadavg.sh  
ldavg 0.00, 0.00, 0.04
```

- 4.3. Erstellen Sie das Verzeichnis **reports**.

Erstellen Sie mit **echo** eine Datei mit etwas Inhalt, nennen Sie die Datei **test.txt** und legen Sie sie in dem neuen Verzeichnis ab.

Wechseln Sie wieder zurück zu **student**, wenn Sie fertig sind.

```
[consultant2@servera content]$ mkdir reports
[consultant2@servera content]$ echo "TEST REPORT" > reports/test.txt
[consultant2@servera content]$ exit
logout
[student@servera ~]$
```

- 4.4. Melden Sie sich als Benutzer **consultant1** an. Verwenden Sie **cd**, um als Benutzer **consultant1** in das Verzeichnis zu wechseln, und **ls**, um das Verzeichnis aufzulisten. Beide Befehle sollten mit der Meldung **Permission denied** fehlschlagen.

Versuchen Sie es mit einem oder mehreren der Befehle von **consultant2**, aber diesmal als **consultant1**, um den fehlenden Zugriff weiter zu überprüfen. Verwenden Sie den vollständigen Pfad **/shares/content**, weil Sie mit **cd** nicht in das Verzeichnis wechseln können.

Wechseln Sie zurück zu **student**, wenn Sie mit dem Testen von **consultant1** fertig sind.

```
[student@servera ~]$ su - consultant1
Password: redhat
[consultant1@servera ~]$ cd /shares/content/
-bash: cd: /shares/content/: Permission denied
[consultant1@servera ~]$ ls /shares/content/
ls: cannot open directory '/shares/content/': Permission denied
[consultant1@servera ~]$ cat /shares/content/serverb-loadavg.txt
cat: /shares/content/serverb-loadavg.txt: Permission denied
[consultant1@servera ~]$ exit
logout
[student@servera ~]$
```

- 4.5. Melden Sie sich als Benutzer **sysadmin1** an. Zeigen Sie mit **getfacl** alle ACL-Einträge in **/shares/content** und die ACL-Einträge in **/shares/content/reports** an.

Wechseln Sie zurück zu **student**, wenn Sie mit dem Testen von **consultant1** fertig sind.

```
[student@servera ~]$ su - sysadmin1
Password: redhat
[sysadmin1@servera ~]$ getfacl /shares/content
getfacl: Removing leading '/' from absolute path names
# file: shares/content/
# owner: root
# group: operators
# flags: -s-
user::rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
```

Kapitel 4 | Steuern des Dateizugriffs mit ACLs

```
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other::---

[sysadmin1@servera ~]$ getfacl /shares/content/reports
getfacl: Removing leading '/' from absolute path names
# file: shares/content/reports
# owner: consultant2
# group: operators
# flags: -s-
user::rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other::---

[sysadmin1@servera ~]$ exit
logout
[student@servera ~]$
```

4.6. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab acl-secure finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab acl-secure finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Steuern des Dateizugriffs mit ACLs

Leistungscheckliste

In dieser Übung richten Sie ein gemeinschaftliches Verzeichnis für Benutzer in zwei Gruppen ein und kombinieren die Berechtigung „set-GID“ und Default-ACL-Einträge, um korrekte Zugriffsberechtigungen bereitzustellen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- set-GID-Berechtigung für einen Ordner konfigurieren, um die Gruppeneigentümerschaft für darin befindliche Dateien und Ordner zu vererben
- ACL-Einträge so konfigurieren, dass sie Lese-/Schreib-/Ausführungs berechtigungen für Benutzer und Gruppen für Dateien und Verzeichnisse zulassen oder verweigern
- Die Default-ACL so konfigurieren, dass die richtigen ACL- und Dateiberechtigungen für neue Dateien und Verzeichnisse automatisch abgerufen werden

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab acl-review start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem werden die in dieser Übung verwendeten Benutzer, Gruppen, Verzeichnisse und Dateien erstellt.

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

Eine Börsenfinanzgesellschaft richtet ein gemeinsames Freigabeverzeichnis für Falldateien ein, für das Mitglieder der Gruppe **managers** Lese- und Schreibberechtigungen haben sollen.

Der Mitbegründer der Gesellschaft, **manager1**, möchte außerdem, dass auch Mitgliedern der Gruppe **contractors** Lese- und Schreibzugriff auf das Freigabeverzeichnis erteilt wird. **manager1** traut jedoch dem Benutzer **contractor3** (ein Mitglied der Gruppe **contractors**) nicht und daher sollte **contractor3** nur Lesezugriff auf das Verzeichnis erhalten.

manager1 hat die Benutzer und Gruppen erstellt und mit dem Einrichten des Freigabeverzeichnisses begonnen, indem er einige Vorlagendateien in das Verzeichnis kopiert hat. Weil **manager1** zu beschäftigt war, müssen Sie den Job beenden.

Ihre Aufgabe ist somit der Abschluss der Einrichtung des Freigabeverzeichnisses. Der Eigentümer des Verzeichnisses und all seiner Inhalte soll die Gruppe **managers** sein. Für alle Dateien sollen außerdem der Eigentümer und die Gruppe (**managers**) Schreib- und Leseberechtigungen erhalten. Anderen Benutzern sollen keine Berechtigungen erteilt werden. Mit Ausnahme von **contractor3**, der ausschließlich Leseberechtigungen erhält, müssen Sie der Gruppe **contractors** noch Lese- und Schreibberechtigungen erteilen. Stellen Sie sicher, dass Sie die Einrichtung so vornehmen, dass sie sowohl für vorhandene als auch für zukünftige Dateien gilt.

Wichtige Informationen:

- Freigabeverzeichnis: **/shares/cases** auf **serverb**
 - Die Benutzer **manager1** und **manager2** sind Mitglieder der Gruppe **managers**.
 - Die Benutzer **contractor1**, **contractor2** und **contractor3** sind Mitglieder der Gruppe **contractors**.
 - Zwei Dateien sind bereits im Verzeichnis vorhanden: **shortlist.txt** und **backlog.txt**.
 - Alle fünf Benutzerpasswörter lauten **redhat**.
 - Alle Änderungen sollten am Verzeichnis **/shares/cases** und an den enthaltenen Dateien vorgenommen werden. Nehmen Sie keine Änderungen am Verzeichnis **/shares** vor.
1. Der Eigentümer des Verzeichnisses **cases** und dessen Inhalt sollte die Gruppe **managers** sein. Eigentümer von neuen Dateien, die dem Verzeichnis **cases** hinzugefügt werden, sollte automatisch die Gruppe **managers** sein. Die Benutzer- und Gruppeneigentümer für die vorhandenen Dateien sollten über Lese- und Schreibberechtigungen verfügen und andere Benutzer sollten keine Berechtigungen haben.



Anmerkung

Hinweis: Verwenden Sie **setfac1** nicht.

2. Fügen Sie dem Verzeichnis **cases** (und seinen Inhalten) ACL-Einträge hinzu, mit denen alle Mitglieder der Gruppe **contractors** Lese-/Schreibzugriff auf die Dateien und Ausführungsberechtigungen für das Verzeichnis erhalten. Beschränken Sie die Berechtigungen des Benutzers **contractor3** auf Lesezugriff auf Dateien und Ausführungsberechtigungen für das Verzeichnis.
3. Fügen Sie ACL-Einträge hinzu, um sicherzustellen, dass auf alle neuen Dateien und Verzeichnisse im Verzeichnis **cases** für *alle* berechtigten Benutzer und Gruppen die richtigen Berechtigungen angewendet werden.
4. Überprüfen Sie, ob die ACL- und Dateisystemänderungen korrekt vorgenommen wurden. Überprüfen Sie mit **ls** und **getfac1** Ihre Einstellungen für **/shares/cases**. Wechseln Sie als Benutzer **student** mit **su - user** zu **manager1** und dann zu **contractor1**. Überprüfen Sie, ob Sie eine Datei lesen und in diese schreiben, ein Verzeichnis erstellen und in Dateien in dem neuen Verzeichnis schreiben können. Überprüfen Sie mit **ls** die neuen Verzeichnisberechtigungen und mit **getfac1** die neue Verzeichnis-ACL. Wechseln Sie als Benutzer **student** mit **su - contractor3** den Benutzer. Versuchen Sie, in eine Datei zu schreiben und ein neues Verzeichnis zu erstellen (beides sollte fehlschlagen). Als Benutzer **contractor3** sollten Sie die Datei **shortlist.txt** im Verzeichnis **cases** lesen und die „Testdateien“ in einem der neuen von den Benutzern **manager1** und **contractor1** erstellten Verzeichnissen lesen können.



Anmerkung

Die obigen Tests sind eine Möglichkeit zur Überprüfung der korrekten Zuweisung der Zugriffsberechtigungen. Sie sollten entsprechende Tests für die Zugriffsüberprüfung für Ihre Umgebung entwickeln.

Bewertung

Führen Sie auf **workstation** den Befehl **lab acl-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** den Befehl **lab acl-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Steuern des Dateizugriffs mit ACLs

Leistungscheckliste

In dieser Übung richten Sie ein gemeinschaftliches Verzeichnis für Benutzer in zwei Gruppen ein und kombinieren die Berechtigung „set-GID“ und Default-ACL-Einträge, um korrekte Zugriffsberechtigungen bereitzustellen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- set-GID-Berechtigung für einen Ordner konfigurieren, um die Gruppeneigentümerschaft für darin befindliche Dateien und Ordner zu vererben
- ACL-Einträge so konfigurieren, dass sie Lese-/Schreib-/Ausführungsberechtigungen für Benutzer und Gruppen für Dateien und Verzeichnisse zulassen oder verweigern
- Die Default-ACL so konfigurieren, dass die richtigen ACL- und Dateiberechtigungen für neue Dateien und Verzeichnisse automatisch abgerufen werden

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student**-Benutzer mit dem Passwort **student** an.

Führen Sie den Befehl **lab acl-review start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem werden die in dieser Übung verwendeten Benutzer, Gruppen, Verzeichnisse und Dateien erstellt.

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

Eine Börsenfinanzgesellschaft richtet ein gemeinsames Freigabeverzeichnis für Falldateien ein, für das Mitglieder der Gruppe **managers** Lese- und Schreibberechtigungen haben sollen.

Der Mitbegründer der Gesellschaft, **manager1**, möchte außerdem, dass auch Mitgliedern der Gruppe **contractors** Lese- und Schreibzugriff auf das Freigabeverzeichnis erteilt wird. **manager1** traut jedoch dem Benutzer **contractor3** (ein Mitglied der Gruppe **contractors**) nicht und daher sollte **contractor3** nur Lesezugriff auf das Verzeichnis erhalten.

manager1 hat die Benutzer und Gruppen erstellt und mit dem Einrichten des Freigabeverzeichnisses begonnen, indem er einige Vorlagendateien in das Verzeichnis kopiert hat. Weil **manager1** zu beschäftigt war, müssen Sie den Job beenden.

Ihre Aufgabe ist somit der Abschluss der Einrichtung des Freigabeverzeichnisses. Der Eigentümer des Verzeichnisses und all seiner Inhalte soll die Gruppe **managers** sein. Für alle Dateien sollen außerdem der Eigentümer und die Gruppe (**managers**) Schreib- und Leseberechtigungen erhalten. Anderen Benutzern sollen keine Berechtigungen erteilt werden. Mit Ausnahme von **contractor3**, der ausschließlich Leseberechtigungen erhält, müssen Sie der Gruppe **contractors** noch Lese- und Schreibberechtigungen erteilen. Stellen Sie sicher, dass Sie die Einrichtung so vornehmen, dass sie sowohl für vorhandene als auch für zukünftige Dateien gilt.

Wichtige Informationen:

- Freigabeverzeichnis: **/shares/cases** auf **serverb**
 - Die Benutzer **manager1** und **manager2** sind Mitglieder der Gruppe **managers**.
 - Die Benutzer **contractor1**, **contractor2** und **contractor3** sind Mitglieder der Gruppe **contractors**.
 - Zwei Dateien sind bereits im Verzeichnis vorhanden: **shortlist.txt** und **backlog.txt**.
 - Alle fünf Benutzerpasswörter lauten **redhat**.
 - Alle Änderungen sollten am Verzeichnis **/shares/cases** und an den enthaltenen Dateien vorgenommen werden. Nehmen Sie keine Änderungen am Verzeichnis **/shares** vor.
1. Der Eigentümer des Verzeichnisses **cases** und dessen Inhalt sollte die Gruppe **managers** sein. Eigentümer von neuen Dateien, die dem Verzeichnis **cases** hinzugefügt werden, sollte automatisch die Gruppe **managers** sein. Die Benutzer- und Gruppeneigentümer für die vorhandenen Dateien sollten über Lese- und Schreibberechtigungen verfügen und andere Benutzer sollten keine Berechtigungen haben.



Anmerkung

Hinweis: Verwenden Sie **setfac1** nicht.

1. Melden Sie sich bei **serverb** als Benutzer **student** an.

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

2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. Aktualisieren Sie mit dem Befehl **chgrp** den Gruppeneigentümer des Verzeichnisses und seiner Inhalte rekursiv.

```
[root@serverb ~]# chgrp -R managers /shares/cases
```

4. Aktualisieren Sie mit dem Befehl **chmod** das Flag **set-GID** für das Verzeichnis.

```
[root@serverb ~]# chmod g+s /shares/cases
```

5. Aktualisieren Sie mit **chmod** alle vorhandenen Dateiberechtigungen für Benutzer und Gruppe zu **rW**.

```
[root@serverb ~]# chmod 660 /shares/cases/*
```

2. Fügen Sie dem Verzeichnis **cases** (und seinen Inhalten) ACL-Einträge hinzu, mit denen alle Mitglieder der Gruppe **contractors** Lese-/Schreibzugriff auf die Dateien und Ausführungsberechtigungen für das Verzeichnis erhalten. Beschränken Sie die Berechtigungen des Benutzers **contractor3** auf Lesezugriff auf Dateien und Ausführungsberechtigungen für das Verzeichnis.
 - 2.1. Aktualisieren Sie mit **setfac1** das vorhandene Verzeichnis **cases** und seine Inhalte rekursiv. Erteilen Sie der Gruppe **contractors** Lese-, Schreib- und bedingte Ausführungsberechtigungen.

```
[root@serverb ~]# setfac1 -Rm g:contractors:rwx /shares/cases
```

- 2.2. Aktualisieren Sie mit **setfac1** das vorhandene Verzeichnis **cases** und seine Inhalte rekursiv. Erteilen Sie dem Benutzer **contractor3** Lese- und bedingte Ausführungsberechtigungen.

```
[root@serverb ~]# setfac1 -Rm u:contractor3:rx /shares/cases
```

3. Fügen Sie ACL-Einträge hinzu, um sicherzustellen, dass auf alle neuen Dateien und Verzeichnisse im Verzeichnis **cases** für *alle* berechtigten Benutzer und Gruppen die richtigen Berechtigungen angewendet werden.
 - 3.1. Aktualisieren Sie mit **setfac1** die *Default*-Berechtigungen für die Mitglieder der Gruppe **contractors**. Default-Berechtigungen sind Lese-, Schreib- und Ausführungsberechtigungen, die für die ordnungsgemäße Erstellung von Unterverzeichnissen und den Zugriff auf diese benötigt werden.

```
[root@serverb ~]# setfac1 -m d:g:contractors:rwx /shares/cases
```

- 3.2. Aktualisieren Sie mit **setfac1** die *Default*-Berechtigungen für den Benutzer **contractor3**. Default-Berechtigungen sind Lese- und Ausführungsberechtigungen, die für den ordnungsgemäßen Zugriff auf Unterverzeichnisse benötigt werden.

```
[root@serverb ~]# setfac1 -m d:u:contractor3:rx /shares/cases
```

4. Überprüfen Sie, ob die ACL- und Dateisystemänderungen korrekt vorgenommen wurden. Überprüfen Sie mit **ls** und **getfac1** Ihre Einstellungen für **/shares/cases**. Wechseln Sie als Benutzer **student** mit **su - user** zu **manager1** und dann zu **contractor1**. Überprüfen Sie, ob Sie eine Datei lesen und in diese schreiben, ein Verzeichnis erstellen und in Dateien in dem neuen Verzeichnis schreiben können. Überprüfen Sie mit **ls** die neuen Verzeichnisberechtigungen und mit **getfac1** die neue Verzeichnis-ACL.

Wechseln Sie als Benutzer **student** mit **su - contractor3** den Benutzer. Versuchen Sie, in eine Datei zu schreiben und ein neues Verzeichnis zu erstellen (beides sollte fehlschlagen). Als Benutzer **contractor3** sollten Sie die Datei **shortlist.txt** im Verzeichnis **cases** lesen und die „Testdateien“ in einem der neuen von den Benutzern **manager1** und **contractor1** erstellten Verzeichnissen lesen können.

- 4.1. Überprüfen Sie als Benutzer **root** mit **ls** das Verzeichnis **cases** und dessen Inhalt. Suchen Sie nach den Berechtigungen für Gruppeneigentümer, Verzeichnis- und Dateiberechtigungen. Das „**s**“ in den Gruppendateiberechtigungen gibt an, dass

Kapitel 4 | Steuern des Dateizugriffs mit ACLs

das **set-GID**-Flag gesetzt und das „+“ gibt an, dass ACL-Einträge vorhanden sind. Beenden Sie dann die Benutzersitzung **root**.

```
[root@serverb ~]# ls -ld /shares/cases
drwxrws---+ 2 root managers 46 Mar 29 00:40 /shares/cases
[root@serverb ~]# ls -l /shares/cases
total 8
-rw-rw----+ 1 root managers 44 Mar 29 00:33 backlog.txt
-rw-rw----+ 1 root managers 46 Mar 29 00:33 shortlist.txt
```

- 4.2. Verwenden Sie **getfacl** und überprüfen Sie die Ausgabe. Suchen Sie in der Standard- und der Default-ACL nach Einträgen für benannte Gruppen und Benutzer.

```
[root@serverb ~]# getfacl /shares/cases
# file: shares/cases
# owner: root
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---
```



```
[root@serverb ~]# exit
logout
```

- 4.3. Wechseln Sie zum Benutzer **manager1** und führen Sie Folgendes aus. Überprüfen Sie, ob der Zugriff wie erwartet gewährt wird.

```
[student@serverb ~]$ su - manager1
Password: redhat
[manager1@serverb ~]$ cd /shares/cases
[manager1@serverb cases]$ echo hello > manager1.txt
[manager1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[manager1@serverb cases]$ mkdir manager1.dir
[manager1@serverb cases]$ echo hello > manager1.dir/test.txt
[manager1@serverb cases]$ ls -ld manager1.dir
drwxrws---+ 2 manager1 managers 22 Mar 29 00:59 manager1.dir
[manager1@serverb cases]$ ls -l manager1.dir
total 4
-rw-rw----+ 1 manager1 managers 6 Mar 29 00:59 test.txt
[manager1@serverb cases]$ getfacl manager1.dir
# file: manager1.dir/
# owner: manager1
```

```
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[manager1@serverb cases]$ exit
logout
```

- 4.4. Wechseln Sie zum Benutzer **contractor1** und führen Sie Folgendes aus. Überprüfen Sie, ob der Zugriff wie erwartet gewährt wird.

```
[student@serverb ~]$ su - contractor1
Password: redhat
[contractor1@serverb ~]$ cd /shares/cases
[contractor1@serverb cases]$ echo hello > manager1.txt
[contractor1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor1@serverb cases]$ mkdir contractor1.dir
[contractor1@serverb cases]$ echo hello > contractor1.dir/test.txt
[contractor1@serverb cases]$ ls -ld contractor1.dir
drwxrws---+ 2 contractor1 managers 22 Mar 29 01:05 contractor1.dir
[contractor1@serverb cases]$ ls -l contractor1.dir
total 4
-rw-rw----+ 1 contractor1 managers 6 Mar 29 01:07 test.txt
[manager1@serverb cases]$ getfacl contractor1.dir
# file: contractor1.dir/
# owner: contractor1
# group: managers
# flags: -s-
user::rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[contractor1@serverb cases]$ exit
logout
```

- 4.5. Wechseln Sie zum Benutzer **contractor3** und führen Sie Folgendes aus. Überprüfen Sie, ob der Zugriff wie erwartet gewährt wird.

```
[student@serverb ~]# su - contractor3
Password: redhat
[contractor3@serverb ~]# cd /shares/cases
[contractor3@serverb cases]# echo hello > contractor3.txt
-bash: contractor3.txt: Permission denied
[contractor3@serverb cases]# cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor3@serverb cases]# mkdir contractor3.dir
mkdir: cannot create directory 'contractor3.dir': Permission denied
[contractor3@serverb cases]# cat manager1.dir/test.txt
hello
[contractor3@serverb cases]# cat contractor1.dir/test.txt
hello
[contractor3@serverb cases]# exit
logout
[student@serverb ~]#
```

- 4.6. Melden Sie sich von **serverb** ab.

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



Anmerkung

Die obigen Tests sind eine Möglichkeit zur Überprüfung der korrekten Zuweisung der Zugriffsberechtigungen. Sie sollten entsprechende Tests für die Zugriffsüberprüfung für Ihre Umgebung entwickeln.

Bewertung

Führen Sie auf **workstation** den Befehl **lab acl-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** den Befehl **lab acl-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- ACLs bieten granulare Zugriffskontrolle für Dateien und Verzeichnisse.
- Der Befehl **getfac1** zeigt die ACLs für eine Datei oder ein Verzeichnis an.
- Mit dem Befehl **setfac1** werden Default- und Standard-ACLs für Dateien und Verzeichnisse festgelegt, geändert und entfernt.
- Default-ACLs zur Steuerung neuer Datei- und Verzeichnisberechtigungen verwenden.
- Red Hat Enterprise Linux verwendet **systemd** und **udev**, um vordefinierte ACLs auf Geräte, Ordner und Dateien anzuwenden.

Kapitel 5

Verwalten der SELinux-Sicherheit

Ziel

Die Sicherheit eines Servers mit SELinux schützen und verwalten

Ziele

- Beschreiben, wie SELinux Ressourcen schützt und wie der Enforcement-Modus ausgewählt wird
- Den SELinux-Kontext einer Datei konfigurieren, um zu steuern, wie Prozesse mit dieser Datei interagieren
- Boolesche SELinux-Werte konfigurieren, um Änderungen der Laufzeitrichtlinie für unterschiedliche Zugriffsanforderungen zuzulassen
- SELinux-Protokollmeldungen untersuchen und SELinux-AVC-Verweigerungen beheben

Abschnitte

- Ändern des SELinux-Enforcement-Modus (und angeleitete Übung)
- Steuern von SELinux-Dateikontexten (und angeleitete Übung)
- Anpassen der SELinux-Richtlinie mit booleschen Werten (und angeleitete Übung)
- Untersuchen und Beheben von SELinux-Problemen (und angeleitete Übung)

Praktische Übung

Verwalten der SELinux-Sicherheit

Ändern des SELinux-Enforcement-Modus

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Erklären, wie SELinux Ressourcen schützt
- Den aktuellen SELinux-Modus eines Systems ändern
- Den SELinux-Standardmodus eines Systems festlegen

So schützt SELinux Ressourcen

SELinux ist eine wichtige Sicherheitsfunktion von Linux. Der Zugriff auf Dateien und andere Ressourcen wird auf granularster Ebene gesteuert. Prozesse dürfen nur auf die Ressourcen zugreifen, die durch ihre Richtlinie oder die booleschen SELinux-Einstellungen festgelegt sind.

Dateiberechtigungen steuern, welche Benutzer oder Benutzergruppen auf welche Dateien zugreifen können. Ein Benutzer, dem Lese- oder Schreibzugriff auf eine bestimmte Datei erteilt wurde, kann diese Datei jedoch auf jede vom Benutzer gewählte Art und Weise verwenden, auch wenn diese Verwendung nicht dem entspricht, wie die Datei verwendet werden sollte.

Sollte beispielsweise eine strukturierte Datendatei mit Schreibzugriff, die nur mit einem bestimmten Programm geschrieben werden kann, von anderen Editoren geöffnet und geändert werden dürfen, was zu Beschädigungen führen könnte?

Dateiberechtigungen können einen solchen unerwünschten Zugriff nicht verhindern. Dateiberechtigungen wurden nicht zur Steuerung entwickelt, wie eine Datei verwendet wird, sondern nur wer eine Datei lesen, schreiben oder ausführen darf.

SELinux besteht aus einer Reihe von Anwendungsentwicklern definierten Richtlinien, die genau angeben, welche Aktionen und Zugriffe für jede von einer Anwendung verwendete ausführbare Binärdatei, Konfigurationsdatei und Datendatei korrekt und zulässig sind. Dies wird *Zielrichtlinie* genannt, weil eine Richtlinie geschrieben wurde, die die Aktivitäten einer einzelnen Anwendung abdeckt. Richtlinien deklarieren vordefinierte Labels für einzelne Programme, Dateien und Netzwerkports.

Warum sollten Sie Security Enhanced Linux verwenden?

Nicht alle Sicherheitsfragen können im Voraus erkannt werden. SELinux erzwingt eine Reihe von Zugriffsregeln, die verhindern, dass eine Schwachstelle in einer Anwendung andere Anwendungen oder das zugrunde liegende System beeinträchtigt. SELinux bietet eine zusätzliche Sicherheitsschicht. Es fügt zudem eine Komplexitätsschicht hinzu, die für nicht mit diesem Subsystem vertraute Personen abschreckend sein kann. Das Erlernen der Arbeit mit SELinux kann einige Zeit in Anspruch nehmen, aber die Enforcement-Richtlinie bedeutet, dass eine Schwäche in einem Teil des Systems nicht auf andere Teile übergreift. Wenn SELinux mit einem bestimmten Subsystem schlecht funktioniert, können Sie Enforcement für diesen bestimmten Service deaktivieren, bis Sie eine Lösung für das zugrunde liegende Problem gefunden haben.

SELinux hat drei Modi:

- Enforcing: SELinux erzwingt Zugriffssteuerungsregeln. Computer werden normalerweise in diesem Modus ausgeführt.
- Permissive: SELinux ist aktiv, aber anstatt die Zugriffssteuerungsregeln durchzusetzen, werden Warnungen aufgrund verletzter Regeln aufgezeichnet. Dieser Modus wird hauptsächlich zum Testen und zur Fehlerbehebung verwendet.
- Disabled: SELinux ist vollständig deaktiviert: Es werden weder SELinux-Verstöße verweigert, noch aufgezeichnet. Nicht empfohlen!

Grundlegende Sicherheitskonzepte von SELinux

Security Enhanced Linux (SELinux) ist eine zusätzliche Systemsicherheitsschicht. Das Hauptziel von SELinux ist der Schutz von Benutzerdaten vor kompromittierten Systemservices. Die meisten Linux-Administratoren sind mit dem standardmäßigen Sicherheitsmodell der Berechtigungen für Benutzer/Gruppe/sonstige vertraut. Hierbei handelt es sich um ein gruppenbasiertes Modell, auch bekannt als uneingeschränkte Zugangskontrolle. SELinux bietet eine zusätzliche Sicherheitsschicht, die objektbasiert ist und von strenger Regeln gesteuert wird, bekannt als obligatorische Zugangskontrolle.

Damit der anonyme Remote-Zugriff auf einen Webserver möglich ist, müssen Firewall-Ports offen sein. Dies gibt böswilligen Personen jedoch die Möglichkeit, das System durch eine Sicherheitslücke zu kompromittieren. Wenn es ihnen gelingt, den Webserverprozess zu kompromittieren, erhalten sie seine Berechtigungen. Insbesondere die Berechtigungen des **Apache**-Benutzers und der **Apache**-Gruppe. Dieser Benutzer und diese Gruppe haben Lesezugriff auf das Document Root **/var/www/html**. Sie haben auch Zugriff auf **/tmp** und **/var/tmp** sowie auf alle anderen Dateien und Verzeichnisse, die von allen beschreibbar sind.

SELinux umfasst eine Reihe von Sicherheitsregeln, die festlegen, welcher Prozess auf welche Dateien, Verzeichnisse und Ports zugreifen darf. Alle Dateien, Prozesse, Verzeichnisse und Ports sind mit einem als SELinux-Kontext bezeichneten Sicherheits-Label versehen. Ein Kontext ist ein von der SELinux-Richtlinie verwendeter Name, um festzustellen, ob ein Prozess auf eine Datei, ein Verzeichnis oder einen Port zugreifen darf. Standardmäßig erlaubt die Richtlinie keine Interaktion, sofern nicht eine explizite Regel den Zugriff gestattet. Ohne entsprechende Regel ist kein Zugriff erlaubt.

SELinux-Label haben mehrere Kontexte: **Benutzer**, **Rolle**, **Typ** und **Sensitivität**. Die Regeln der Zielrichtlinie, die in Red Hat Enterprise Linux standardmäßig aktivierte Richtlinie ist, basieren auf der dritten Kontextvariante: dem Typkontext. Typkontextnamen enden gewöhnlich auf **_t**.

unconfined_u:object_r:httpd_sys_content_t:s0	/var/www/html/file2			
SELinux User	Role	Type	Level	File

Abbildung 5.1: SELinux-Dateikontext

Der Typkontext für einen Webserver ist **httpd_t**. Der Typkontext für Dateien und Verzeichnisse, die sich normalerweise in **/var/www/html** befinden, lautet **httpd_sys_content_t**. Die Kontexte für Dateien und Verzeichnisse in **/tmp** und **/var/tmp** lauten **tmp_t**. Der Typkontext für Webserverports ist **httpd_port_t**.

Apache hat den Typkontext **httpd_t**. Es gibt eine Richtlinienregel, die Apache den Zugriff auf Dateien und Verzeichnisse mit dem Typkontext **httpd_sys_content_t** erlaubt. Standardmäßig haben Dateien in **/var/www/html** und anderen Webserver-Verzeichnissen den Typkontext

httpd_sys_content_t. Es gibt keine **Zulassungsregel** in der Richtlinie für Dateien, die normalerweise in **/tmp** und **/var/tmp** vorhanden sind, daher wird der Zugriff nicht erlaubt. Wenn SELinux aktiviert ist, könnte ein Benutzer mit böswilligen Absichten, der den Webserverprozess kompromittieren wollte, nicht auf das Verzeichnis **/tmp** zugreifen.

Der Server **MariaDB** hat den Typkontext **mysqld_t**. Dateien in **/data/mysql** haben standardmäßig den Typkontext **mysqld_db_t**. Dieser Typkontext ermöglicht **MariaDB** den Zugriff auf diese Dateien, deaktiviert jedoch den Zugriff anderer Services, z. B. den Apache-Webservice.

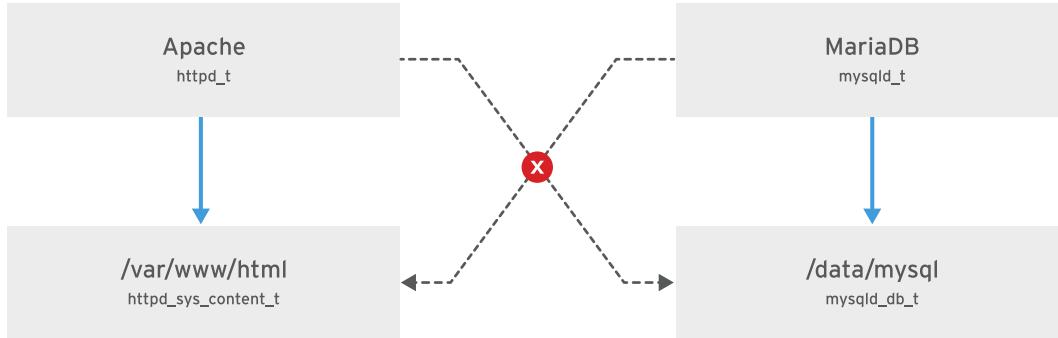


Abbildung 5.2: SELinux-Zugriff

Viele Befehle für Dateien verwenden die Option **-Z**, um SELinux-Kontexte anzuzeigen oder festzulegen. Bei **ps**, **ls**, **cp** und **mkdir** können SELinux-Kontexte beispielsweise mit der Option **-Z** angezeigt oder festgelegt werden.

```
[root@host ~]# ps axZ
LABEL PID TTY STAT TIME COMMAND
system_u:system_r:init_t:s0 1 ? Ss 0:09 /usr/lib/systemd/...
system_u:system_r:kernel_t:s0 2 ? S 0:00 [kthreadd]
system_u:system_r:kernel_t:s0 3 ? S 0:00 [ksoftirqd/0]
...output omitted...
[root@host ~]# systemctl start httpd
[root@host ~]# ps -ZC httpd
LABEL PID TTY TIME CMD
system_u:system_r:httpd_t:s0 1608 ? 00:00:05 httpd
system_u:system_r:httpd_t:s0 1609 ? 00:00:00 httpd
...output omitted...
[root@host ~]# ls -Z /home
drwx-----. root root system_u:object_r:lost_found_t:s0 lost+found
drwx-----. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@host ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

Ändern des aktuellen SELinux-Modus

Das SELinux-Subsystem bietet Tools zum Anzeigen und Ändern von Modi. Um den aktuellen SELinux-Modus zu ermitteln, führen Sie den Befehl **getenforce** aus. Mit dem Befehl **setenforce** legen Sie SELinux auf einen anderen Modus fest:

```
[user@host ~]# getenforce
Enforcing
[user@host ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[user@host ~]# setenforce 0
[user@host ~]# getenforce
Permissive
[user@host ~]# setenforce Enforcing
[user@host ~]# getenforce
Enforcing
```

Alternativ können Sie den SELinux-Modus beim Booten festlegen, indem Sie einen Parameter an den Kernel übergeben: Das Kernel-Argument **enforcing=0** bootet das System im permissiven Modus, der Wert **enforcing=1** legt den Enforcing-Modus fest. Sie können SELinux auch vollständig deaktivieren, indem Sie den Kernel-Parameter **selinux=0** übergeben. Mit dem Wert **selinux=1** wird SELinux aktiviert.

Festlegen des SELinux-Standardmodus

Sie können SELinux auch dauerhaft mithilfe der Datei **/etc/selinux/config** konfigurieren. Im folgenden Beispiel (die Standardkonfiguration) legt die Konfigurationsdatei SELinux auf **enforcing** fest. Die Kommentare zeigen auch die anderen gültigen Werte: **permissive** und **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.
SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes
#                  are protected.
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Das System liest diese Datei beim Booten und konfiguriert SELinux wie gezeigt. Kernel-Argumente (**selinux=0|1** und **enforcing=0|1**) überschreiben diese Konfiguration.



Literaturhinweise

Manpages **getenforce(8)**, **setenforce(8)** und **selinux_config(5)**

► Angeleitete Übung

Ändern des SELinux-Enforcement-Modus

In dieser praktischen Übung verwalten Sie SELinux-Modi sowohl temporär als auch dauerhaft.

Ergebnisse

Sie sollten in der Lage sein, den aktuellen SELinux-Modus anzuzeigen und festzulegen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-opsmode start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist.

```
[student@workstation ~]$ lab selinux-opsmode start
```

- ▶ 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- ▶ 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Ändern Sie den SELinux-Standardmodus in „Permissive“ und booten Sie das System neu.

- 3.1. Überprüfen Sie mit dem Befehl **getenforce**, ob **servera** im Enforcing-Modus ausgeführt wird.

```
[root@servera ~]# getenforce
Enforcing
```

- 3.2. Öffnen Sie mit dem Befehl **vim** die Konfigurationsdatei **/etc/selinux/config**. Ändern Sie den Parameter **SELINUX** von **enforcing** in **permissive**.

```
[root@servera ~]# vim /etc/selinux/config
```

- 3.3. Überprüfen Sie mit dem Befehl **grep**, ob der Parameter **SELINUX** auf **permissive** festgelegt ist.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=permissive
SELINUXTYPE=targeted
```

- 3.4. Booten Sie mit dem Befehl **systemctl reboot servera** neu.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- 4. Das Booten von **servera** dauert einige Minuten. Melden Sie sich nach einigen Minuten bei **servera** als Benutzer **student** an. Führen Sie den Befehl **sudo -i** aus, um **root**-Berechtigungen zu erhalten. Zeigen Sie mit dem Befehl **getenforce** den aktuellen SELinux-Modus an.

- 4.1. Melden Sie sich auf **workstation** mit dem Befehl **ssh** bei **servera** als Benutzer **student** an.

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

- 4.2. Führen Sie den Befehl **sudo -i** aus, um **root**-Berechtigungen zu erhalten.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 4.3. Zeigen Sie mit dem Befehl **getenforce** den aktuellen SELinux-Modus an.

```
[root@servera ~]# getenforce
Permissive
```

- 5. Ändern Sie in der Datei **/etc/selinux/config** den SELinux-Standardmodus in „Enforcing“. Diese Änderung wird erst beim nächsten Rebooten wirksam.

- 5.1. Öffnen Sie mit dem Befehl **vim** die Konfigurationsdatei **/etc/selinux/config**. Ändern Sie **SELINUX** wieder in **enforcing**.

```
[root@servera ~]# vim /etc/selinux/config
```

- 5.2. Überprüfen Sie mit dem Befehl **grep**, ob der Parameter **SELINUX** auf **enforcing** festgelegt ist.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config  
SELINUX=enforcing  
SELINUXTYPE=targeted
```

- 6. Legen Sie mit dem Befehl **setenforce** den aktuellen SELinux-Modus ohne Rebooten auf **enforcing** fest. Überprüfen Sie mit dem Befehl **getenforce**, ob der Modus auf **enforcing** festgelegt wurde.

```
[root@servera ~]# setenforce 1  
[root@servera ~]# getenforce  
Enforcing
```

- 7. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab selinux-opsmode finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab selinux-opsmode finish
```

Hiermit ist die angeleitete Übung beendet.

Steuern von SELinux-Dateikontexten

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- SELinux-Richtlinienregeln mit dem Befehl **semanage fcontext** verwalten, die den Standardkontext für Dateien und Verzeichnisse festlegen
- Den von der SELinux-Richtlinie festgelegten Kontext auf Dateien und Verzeichnisse mit dem Befehl **restorecon** anwenden

Anfänglicher SELinux-Kontext

Auf Systemen mit SELinux werden alle Prozesse und Dateien mit einem Label gekennzeichnet. Das Label repräsentiert die sicherheitsrelevanten Informationen, die als SELinux-Kontext bezeichnet werden.

Neue Dateien erben in der Regel ihren SELinux-Kontext vom übergeordneten Verzeichnis. Auf diese Weise ist sichergestellt, dass sie den richtigen Kontext haben.

Dieses Vererbungsverfahren kann jedoch auf zwei verschiedene Arten unterlaufen werden. Erstens: Wenn Sie eine Datei an einem anderen Ort als dem endgültig vorgesehenen Ort erstellen und dann die Datei verschieben, hat die Datei immer noch den SELinux-Kontext des Verzeichnisses, in dem sie erstellt wurde, nicht den des Zielverzeichnisses. Zweitens: Wenn Sie eine Datei mit einem Befehl wie **cp -a** kopieren, der den SELinux-Kontext beibehält, gibt der SELinux-Kontext den Speicherort der Originaldatei wieder.

Das folgende Beispiel demonstriert die Vererbung und ihre Tücken. Es sind zwei Dateien vorhanden, die in **/tmp** erstellt wurden, eine wurde nach **/var/www/html** verschoben und die zweite in dasselbe Verzeichnis kopiert. Beachten Sie die SELinux-Kontexte der Dateien. Die Datei, die in das Verzeichnis **/var/www/html** verschoben wurde, behält den Dateikontext für das Verzeichnis **/tmp** bei. Die Datei, die in das Verzeichnis **/var/www/html** kopiert wurde, hat den SELinux-Kontext des Verzeichnisses **/var/www/html** geerbt.

Der Befehl **ls -Z** zeigt den SELinux-Kontext einer Datei an. Beachten Sie das Label der Datei.

```
[root@host ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

Und der Befehl **ls -Zd** zeigt den SELinux-Kontext eines Verzeichnisses an:

```
[root@host ~]# ls -Zd /var/www/html/
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

Beachten Sie, dass **/var/www/html/index.html** das gleiche Label wie das übergeordnete Verzeichnis **/var/www/html/** hat. Erstellen Sie jetzt Dateien außerhalb des Verzeichnisses **/var/www/html** und beachten Sie deren Dateikontext:

```
[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
```

Verschieben Sie eine dieser Dateien in das Verzeichnis **/var/www/html**, kopieren Sie eine andere und beachten Sie das Label dieser Dateien:

```
[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
```

Die verschobene Datei behält ihr ursprüngliches Label bei, während die kopierte Datei das Label des Verzeichnisses **/var/www/html** erbt. **unconfined_u**: ist der Benutzer, **object_r**: bezeichnet die Rolle und **s0** ist die Stufe. Die Sensitivitätsstufe 0 ist die niedrigste mögliche Sensitivitätsstufe.

Ändern des SELinux-Kontextes einer Datei

Mit den Befehlen **semanage fcontext**, **restorecon** und **chcon** kann der SELinux-Kontext für Dateien geändert werden.

Die bevorzugte Methode zum Festlegen des SELinux-Kontextes für eine Datei besteht darin, das Standard-Label für eine Datei mit dem Befehl **semanage fcontext** zu definieren und dann diesen Kontext auf die Datei mit dem Befehl **restorecon** anzuwenden. Dadurch wird sichergestellt, dass das Label auch nach einer vollständigen Umbenennung des Dateisystems wie gewünscht erhalten bleibt.

Der Befehl **chcon** ändert die SELinux-Kontexte. **chcon** legt den Sicherheitskontext für die im Dateisystem gespeicherte Datei fest. Dieser Befehl ist zum Testen und Experimentieren sehr hilfreich. Er speichert jedoch keine Kontextänderungen in der SELinux-Kontextdatenbank. Wenn ein **restorecon**-Befehl ausgeführt wird, bleiben von dem Befehl **chcon** vorgenommene Änderungen ebenfalls nicht erhalten. Wenn das gesamte Dateisystem umbenannt wird, wird auch der SELinux-Kontext für Dateien zurückgesetzt, die mit **chcon** geändert wurden.

Das folgende Beispiel zeigt ein Verzeichnis, das gerade erstellt wird. Das Verzeichnis hat den Typwert **default_t**.

```
[root@host ~]# mkdir /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

Der Befehl **chcon** ändert den Dateikontext des Verzeichnisses **/virtual**: Der Typwert ändert sich in **httpd_sys_content_t**.

```
[root@host ~]# chcon -t httpd_sys_content_t /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
```

Der Befehl **restorecon** wird ausgeführt und der Typwert wird auf den Wert **default_t** zurückgesetzt. Beachten Sie die Meldung **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
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

Definieren von Regeln für den SELinux-Standarddateikontext

Mit dem Befehl **semanage fcontext** können die Regeln, mit denen **restorecon** den Standarddateikontext festlegt, angezeigt und geändert werden. Die Pfad- und Dateinamen werden dabei mithilfe erweiterter regulärer Ausdrücke angegeben. Der häufigste erweiterte reguläre Ausdruck in **fcontext-Regeln** ist **(/.*)?**, d. h. „optional, suche einen / gefolgt von einer beliebigen Anzahl von Zeichen“. Damit werden das vor diesem Ausdruck stehende Verzeichnis und der gesamte Verzeichnisinhalt rekursiv abgeglichen.

Einfache Dateikontextvorgänge

In der folgenden Tabelle sind die Optionen von **semanage fcontext** zum Hinzufügen, Entfernen oder Auflisten von SELinux-Dateikontexten aufgeführt.

semanage fcontext-Befehle

Option	Beschreibung
-a, --add	Einen Datensatz des festgelegten Objekttyps hinzufügen
-d, --delete	Einen Datensatz des festgelegten Objekttyps löschen
-l, --list	Einen Datensatz des festgelegten Objekttyps aufführen

Um über die Tools zur Verwaltung von SELinux-Kontexten zu verfügen, installieren Sie bei Bedarf das Paket **policycoreutils** sowie das Paket **policycoreutils-python**. Diese enthalten den Befehl **restorecon** bzw. den Befehl **semanage**.

Um sicherzustellen, dass alle Dateien in einem Verzeichnis den richtigen Dateikontext haben, führen Sie **semanage fcontext -l** gefolgt von dem Befehl **restorecon** aus. Beachten Sie im folgenden Beispiel den Dateikontext jeder Datei vor und nach der Ausführung der Befehle **semanage** und **restorecon**.

```
[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
```

```
[root@host ~]# semanage fcontext -l
...output omitted...
/var/www(/.*)?    all files    system_u:object_r:httpd_sys_content_t:s0
...output omitted...
```

```
[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
```

Das folgende Beispiel zeigt, wie **semanage** verwendet wird, um einen Kontext für ein neues Verzeichnis hinzuzufügen.

```
[root@host ~]# mkdir /virtual  
[root@host ~]# touch /virtual/index.html  
[root@host ~]# ls -Zd /virtual/  
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
```

```
[root@host ~]# ls -Z /virtual/  
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html  
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(/.*)?'  
[root@host ~]# restorecon -RFvv /virtual  
[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
```



Literaturhinweise

Manpages **chcon(1)**, **restorecon(8)**, **semanage(8)** und **semanage-fcontext(8)**

► Angeleitete Übung

Steuern von SELinux-Dateikontexten

In dieser praktischen Übung ändern Sie den SELinux-Kontext eines Verzeichnisses und seiner Inhalte dauerhaft.

Ergebnisse

Sie sollten in der Lage sein, den Apache-HTTP-Server zum Veröffentlichen von Webinhalten aus einem vom Standard abweichenden Document Root zu konfigurieren.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-filecontexts start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Er installiert zudem den **httpd**-Service und konfiguriert die Firewall auf **servera** so, dass HTTP-Verbindungen zugelassen werden.

```
[student@workstation ~]$ lab selinux-filecontexts start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Konfigurieren Sie Apache so, dass ein vom Standard abweichender Document Root verwendet werden kann.

- 3.1. Erstellen Sie mit dem Befehl **mkdir** den neuen Document Root **/custom**.

```
[root@servera ~]# mkdir /custom
```

- 3.2. Erstellen Sie im Document Root **/custom** mit dem Befehl **echo** die Datei **index.html**.

```
[root@servera ~]# echo 'This is SERVERA.' > /custom/index.html
```

- 3.3. Konfigurieren Sie Apache so, dass der neue Document Root-Speicherort verwendet wird. Bearbeiten Sie dazu die Apache-Konfigurationsdatei **/etc/httpd/conf/httpd.conf** und ersetzen Sie die beiden Vorkommen von **/var/www/html** durch **/custom**.

```
...output omitted...
DocumentRoot "/custom"
...output omitted...
<Directory "/custom">
...output omitted...
```

- 4. Starten und aktivieren Sie den Apache-Webservice und überprüfen Sie, ob der Service ausgeführt wird.

- 4.1. Starten Sie und aktivieren Sie den Apache-Webservice mit dem Befehl **systemctl**.

```
[root@servera ~]# systemctl enable --now httpd
```

- 4.2. Überprüfen Sie mit dem Befehl **systemctl**, ob der Service ausgeführt wird.

```
[root@servera ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2019-03-25 19:16:48 CET; 15h ago
    Docs: man:httpd.service(8)
 Main PID: 6565 (httpd)
   Status: "Total requests: 16; Idle/Busy workers 100/0;Requests/sec: 0.000285;
 Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 11406)
   Memory: 37.3M
  CGroup: /system.slice/httpd.service
          ├─6565 /usr/sbin/httpd -DFOREGROUND
          ├─6566 /usr/sbin/httpd -DFOREGROUND
          ├─6567 /usr/sbin/httpd -DFOREGROUND
          ├─6568 /usr/sbin/httpd -DFOREGROUND
          └─6569 /usr/sbin/httpd -DFOREGROUND

Mar 25 19:16:48 servera.lab.example.com systemd[1]: Starting The Apache HTTP
Server...
Mar 25 19:16:48 servera.lab.example.com httpd[6565]: Server configured, listening
on: port 80
Mar 25 19:16:48 servera.lab.example.com systemd[1]: Started The Apache HTTP
Server.
```

- 5. Öffnen Sie auf **workstation** einen Webbrowser und versuchen Sie **http://servera/index.html** anzuzeigen. Sie erhalten eine Fehlermeldung, die Sie darauf hinweist, dass Sie keine Zugriffsberechtigung für diese Datei haben.

- 6. Für den Zugriff auf die Datei **index.html** auf **servera** muss SELinux konfiguriert sein. Definieren Sie eine SELinux-Dateikontextregel, die den Kontexttyp für das Verzeichnis **/custom** und alle darin befindlichen Dateien auf **httpd_sys_content_t** festlegt.

```
[root@servera ~]# semanage fcontext -a \
-t httpd_sys_content_t '/custom(/.*)?'
```

- 7. Ändern Sie mit dem Befehl **restorecon** die Dateikontexte.

```
[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
```

- 8. Versuchen Sie, <http://servera/index.html> erneut aufzurufen. Nun sollte die Meldung **This is SERVERA.** erscheinen.
- 9. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab_selinux-filecontexts_finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab_selinux-filecontexts finish
```

Hiermit ist die angeleitete Übung beendet.

Anpassen der SELinux-Richtlinie mit booleschen Werten

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- SELinux-Richtlinienregeln mit **setsebool** aktivieren und deaktivieren
- Den persistenten Wert boolescher SELinux-Werte mit dem Befehl **semanage boolean -l** verwalten
- Manpages, die mit **_selinux** enden zurate ziehen, um nützliche Informationen über boolesche SELinux-Werte zu erhalten

Boolesche SELinux-Werte

Mit booleschen SELinux-Werten können Sie das Verhalten der SELinux-Richtlinie ändern. Bei diesen Werten handelt es sich um aktivierbare und deaktivierbare Regeln. Sicherheitsadministratoren können mit ihrer Hilfe gezielt Feinabstimmungen der Richtlinie vornehmen.

In den SELinux-Manpages, die mit dem Paket *selinux-policy-doc* bereitgestellt werden, ist der Zweck der verfügbaren booleschen Werte beschrieben. Der Befehl **man -k '_selinux'** listet diese Manpages auf.

Befehle, die zur Verwaltung boolescher SELinux-Werte nützlich sind, umfassen den Befehl **getsebool**, der boolesche Werte und deren Status aufführt, und den Befehl **setsebool**, der boolesche Werte ändert. **setsebool -P** ändert die SELinux-Richtlinie, um die Änderung persistent zu machen. Und **semanage boolean -l** gibt zusammen mit einer kurzen Beschreibung des booleschen Werts an, ob ein boolescher Wert persistent ist oder nicht.

Unprivilegierte Benutzer können den Befehl **getsebool** ausführen, aber Sie müssen ein Superuser sein, um **semanage boolean -l** und **setsebool -P** auszuführen.

```
[user@host ~]$ getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
... output omitted...
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

```
[user@host ~]$ setsebool httpd_enable_homedirs on
Could not change active booleans. Please try as root: Permission denied
[user@host ~]$ sudo setsebool httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , off)  Allow httpd to enable homedirs
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> on
```

Mit der Option **-P** werden alle ausstehenden Werte in die Richtlinie geschrieben, sodass sie bei Reboots erhalten bleiben. Beachten Sie im folgenden Beispiel die Werte in Klammern: Beide sind jetzt auf **on** festgelegt.

```
[user@host ~]$ setsebool -P httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs          (on , on)  Allow httpd to enable homedirs
```

Um boolesche Werte aufzulisten, in denen der aktuelle Status vom Standardstatus abweicht, führen Sie **semanage boolean -l -C** aus.

```
[user@host ~]$ sudo semanage boolean -l -C
SELinux boolean           State  Default Description
cron_can_relabel          (off , on)  Allow cron to can relabel
```



Literaturhinweise

Manpages **booleans(8)**, **getsebool(8)**, **setsebool(8)**, **semanage(8)**,
semanage-boolean(8)

► Angeleitete Übung

Anpassen der SELinux-Richtlinie mit booleschen Werten

Apache kann Webinhalte veröffentlichen, die in Benutzerverzeichnissen gehostet werden. Bei SELinux hingegen ist diese Option standardmäßig unterdrückt. In dieser Übung identifizieren und ändern Sie den booleschen SELinux-Wert, mit dem Apache auf Benutzerverzeichnisse zugreifen kann.

Ergebnisse

Sie sollten in der Lage sein, Apache so zu konfigurieren, dass Webinhalte aus Benutzerverzeichnissen veröffentlicht werden.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-booleans start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Er installiert zudem den **httpd**-Service und konfiguriert die Firewall auf **servera** so, dass HTTP-Verbindungen zugelassen werden.

```
[student@workstation ~]$ lab selinux-booleans start
```

- ▶ 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- ▶ 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- ▶ 3. Um das Apache-Feature zu aktivieren, das Benutzern die Veröffentlichung von Webinhalten aus ihren Benutzerverzeichnissen ermöglicht, müssen Sie die Konfigurationsdatei **/etc/httpd/conf.d/userdir.conf** bearbeiten. Kommentieren Sie die Zeile aus, die **UserDir** auf **disabled** festlegt, und heben Sie die Auskommentierung der Zeile auf, die **UserDir** auf **public_html** festlegt.

```
[root@servera ~]# vim /etc/httpd/conf.d/userdir.conf
#UserDir disabled
UserDir public_html
```

- 4. Überprüfen Sie mit dem Befehl **grep** die Änderungen.

```
[root@servera ~]# grep '#UserDir' /etc/httpd/conf.d/userdir.conf
#UserDir disabled
[root@servera ~]# grep '^ *UserDir' /etc/httpd/conf.d/userdir.conf
UserDir public_html
```

- 5. Starten und aktivieren Sie den Apache-Webservice, damit die Änderungen wirksam werden.

```
[root@servera ~]# systemctl enable --now httpd
```

- 6. Melden Sie sich in einem anderen Terminalfenster als **student** an. Melden Sie sich über SSH bei **servera** an. Erstellen Sie Webinhalt, der aus einem Verzeichnis des Benutzers veröffentlicht werden soll.

- 6.1. Melden Sie sich in einem anderen Terminalfenster als **student** an. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an.

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

- 6.2. Erstellen Sie mit dem Befehl **mkdir** ein Verzeichnis mit dem Namen **~/public_html**.

```
[student@servera ~]$ mkdir ~/public_html
```

- 6.3. Erstellen Sie die Datei **index.html** mit dem folgenden Inhalt:

```
[student@servera ~]$ echo 'This is student content on SERVERA.' > \
~/public_html/index.html
```

- 6.4. Ändern Sie mit dem Befehl **chmod** die Berechtigungen des Benutzerverzeichnisses von **student** so, dass Apache auf das Unterverzeichnis **public_html** zugreifen kann.

```
[student@servera ~]$ chmod 711 ~
```

- 7. Öffnen Sie auf **workstation** einen Webbrowser und versuchen Sie, folgende URL aufzurufen: <http://servera/~student/index.html>. Sie erhalten eine Fehlermeldung, die Sie darauf hinweist, dass Sie keine Zugriffsberechtigung für diese Datei haben.

- 8. Stellen Sie in einem Terminalfenster mit **root**-Zugriff mit dem Befehl **getsebool** fest, ob es boolesche Werte gibt, die den Zugriff auf Benutzerverzeichnisse beschränken.

```
[root@servera ~]# getsebool -a | grep home  
...output omitted...  
httpd_enable_homedirs --> off  
...output omitted...
```

- 9. Aktivieren Sie mit dem Befehl **setsebool** in dem Terminalfenster mit **root**-Zugriff den Zugriff auf Benutzerverzeichnisse dauerhaft.

```
[root@servera ~]# setsebool -P httpd_enable_homedirs on
```

- 10. Versuchen Sie, `http://servera/~student/index.html` erneut aufzurufen. Es erscheint folgende Meldung: **This is student content on SERVERA.**

- 11. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab selinux-booleans finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab selinux-booleans finish
```

Hiermit ist die angeleitete Übung beendet.

Untersuchen und Beheben von SELinux-Problemen

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Tools zur Analyse von SELinux-Protokollen verwenden
- Mit dem Befehl **sealert** nützliche Informationen bei der SELinux-Fehlerbehebung anzeigen

Beheben von SELinux-Problemen

Es ist wichtig, dass Sie wissen, welche Maßnahmen Sie ergreifen müssen, wenn SELinux den Zugriff auf Dateien auf einem Server verhindert, von denen Sie wissen, dass sie zugänglich sein sollten. Führen Sie die folgenden Schritte aus, um diese Probleme zu beheben:

1. Bevor Sie darüber nachdenken, Einstellungen zu ändern, sollten Sie erwägen, dass es korrekt sein könnte, dass SELinux den versuchten Zugriff verweigert. Wenn ein Webserver versucht, auf Dateien in **/home** zuzugreifen, könnte dies ein Hinweis auf eine Kompromittierung des Service sein, sofern Webinhalte nicht von Benutzern veröffentlicht werden. Wenn der Zugriff aber hätte gewährt werden sollen, dann sind weitere Schritte erforderlich, um das Problem zu lösen.
2. Das häufigste SELinux-Problem ist ein falscher Dateikontext. Das kann passieren, wenn eine Datei an einem Ort mit einem bestimmten Dateikontext erstellt wird und dann an einen Ort verschoben wird, an dem ein anderer Kontext erwartet wird. In den meisten Fällen wird dieses Problem durch den Befehl **restorecon** behoben. Eine Problembehebung dieser Art hat sehr geringe Auswirkungen auf die Sicherheit des restlichen Systems.
3. Ein anderes Mittel gegen übermäßig restriktive Zugriffsberechtigungen kann die Anpassung eines booleschen Werts sein. So regelt beispielsweise der boolesche Wert **ftpd_anon_write**, ob anonyme FTP-Benutzer Dateien hochladen können. Sie müssen diesen boolesche Wert aktivieren, damit anonyme FTP-Benutzer Dateien auf einen Server hochladen können. Die Anpassung boolescher Werte erfordert eine größere Sorgfalt, weil sie beträchtliche Auswirkungen auf die Systemsicherheit haben können.
4. Es kann sein, dass ein Programmfehler in der SELinux-Richtlinie den rechtmäßigen Zugriff verhindert. Da SELinux inzwischen ausgereifter ist, kommt dies selten vor. Wenn definitiv ein Programmfehler in der Richtlinie gefunden wurde, teilen Sie diesen Fehler dem Red Hat Support mit, damit er behoben werden kann.

Überwachen von SELinux-Verstößen

Installieren Sie das Paket **setroubleshoot-server**, um SELinux-Meldungen an **/var/log/messages** zu senden. **setroubleshoot-server** überwacht **/var/log/audit/audit.log** auf Auditmeldungen und sendet eine kurze Zusammenfassung an **/var/log/messages**. Diese Zusammenfassung enthält eindeutige Bezeichner (**UUIDs**) für SELinux-Verstöße, mit deren Hilfe weitere Informationen erfasst werden können. Mit dem Befehl **sealert -l **UUID**** wird ein Bericht zu einem bestimmten Vorfall erstellt. Mit **sealert -a /var/log/audit/audit.log** werden Berichte für alle in dieser Datei enthaltenen Vorfälle erstellt.

Sehen Sie sich die folgenden Beispieldaten für Befehle auf einem standardmäßigen Apache-Webserver an:

```
[root@host ~]# touch /root/file3
[root@host ~]# mv /root/file3 /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/file3
<!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 /file3
on this server.</p>
</body></html>
```

Sie erwarten, dass der Webserver den Inhalt von **file3** bereitstellt, stattdessen gibt er einen **permission denied**-Fehler zurück. Bei der Untersuchung von **/var/log/audit/audit.log** und **/var/log/messages** erhalten Sie zusätzliche Informationen über diesen Fehler.

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1392944135.482:429): avc: denied { setattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...output omitted...
[root@host ~]# tail /var/log/messages
...output omitted...
Feb 20 19:55:42 host setroubleshoot: SELinux is preventing /usr/sbin/httpd
from setattr access on the file . For complete SELinux messages. run
sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

In beiden Protokolldateien ist angegeben, dass der Fehler auf eine SELinux-Zugriffsverweigerung zurückzuführen ist. Der Befehl **sealert**, der Bestandteil der Ausgabe in **/var/log/messages** ist, liefert weitere Informationen, einschließlich einer möglichen Behebung.

```
[root@host ~]# sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
SELinux is preventing /usr/sbin/httpd from setattr access on the file .

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed setattr access on the
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:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp

Additional Information:
Source Context          system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:admin_home_t:s0
Target Objects           [ file ]
```

```

Source httpd
Source Path /usr/sbin/httpd
Port <Unknown>
Host servera
Source RPM Packages httpd-2.4.6-14.el7.x86_64
Target RPM Packages
Policy RPM selinux-policy-3.12.1-124.el7.noarch
Selinux Enabled True
Policy Type targeted
Enforcing Mode Enforcing
Host Name servera
Platform Linux servera 3.10.0-84.el7.x86_64 #1
SMP Tue Feb 4 16:28:19 EST 2014 x86_64 x86_64
Alert Count 2
First Seen 2014-02-20 19:55:35 EST
Last Seen 2014-02-20 19:55:35 EST
Local ID 613ca624-248d-48a2-a7d9-d28f5bbe2763

Raw Audit Messages
type=AVC msg=audit(1392944135.482:429): avc: denied { getattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file

type=SYSCALL msg=audit(1392944135.482:429): arch=x86_64 syscall=lstat
success=no exit=EACCES a0=7f9fed0edeaa8 a1=7fff7bffc770 a2=7fff7bffc770
a3=0 items=0 ppid=1608 pid=1609 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

```



Anmerkung

Im Abschnitt **Raw Audit Messages** wird die problematische Zieldatei angegeben: **/var/www/html/file3**. Zudem sieht auch der Zielkontext **tcontext** nicht so aus, als ob er zu einem Webserver gehört. Korrigieren Sie mit dem Befehl **restorecon /var/www/html/file3** den Dateikontext. Wenn noch weitere Dateien angepasst werden müssen, kann der Kontext mit dem Befehl **restorecon** rekursiv zurückgesetzt werden: **restorecon -R /var/www/**.

Der Abschnitt **Raw Audit Messages** des Befehls **sealert** enthält Informationen aus **/var/log/audit.log**. Durchsuchen Sie mit dem Befehl **ausearch** die Datei **/var/log/audit.log**. Mit **-m** wird der Meldungstyp gesucht. Mit der Option **-ts** wird nach der Zeit gesucht.

```
[root@host ~]# ausearch -m AVC -ts recent
-----
time->Tue Apr  9 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
```

Webkonsole

Wenn *Web Console* installiert ist, können damit auch SELinux-Probleme behoben werden. Melden Sie sich bei Web Console an und wählen Sie aus dem Menü auf der linken Seite **SELinux** aus. Das Fenster „SELinux Policy“ enthält Informationen zur aktuellen Durchsetzungsrichtlinie. Eventuelle Probleme werden im Abschnitt **SELinux Access Control Errors** detailliert aufgeführt.

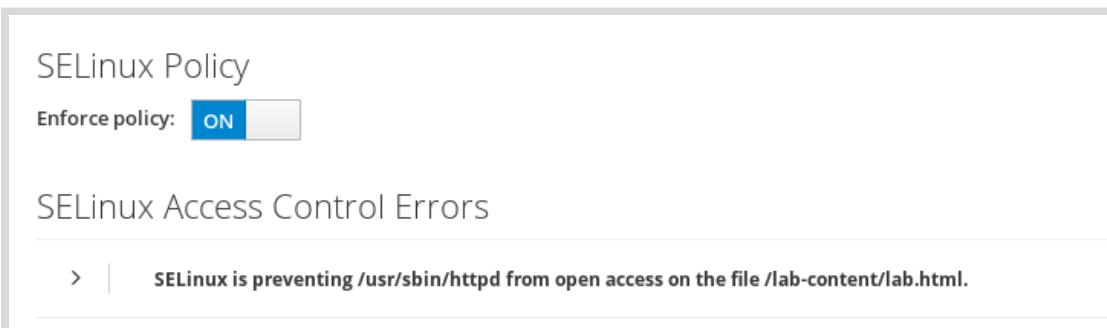


Abbildung 5.3: SELinux-Richtlinie in Web Console

Klicken Sie auf das Zeichen >, um Fehlerdetails anzuzeigen. Klicken Sie auf **solution details**, um alle Details und mögliche Lösungen anzuzeigen.

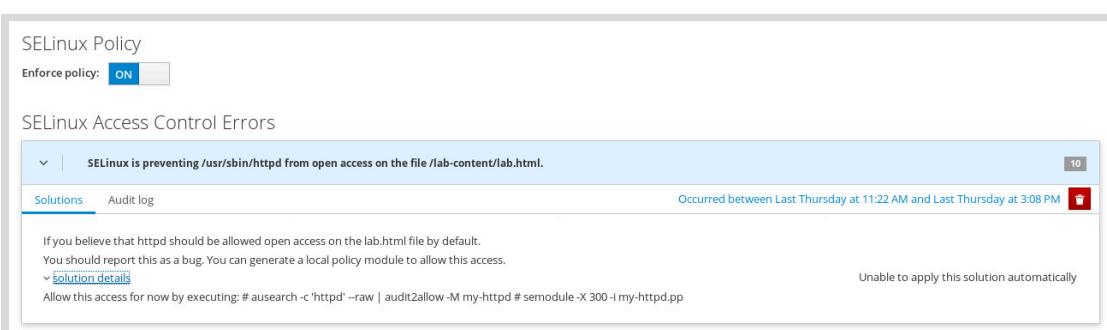


Abbildung 5.4: SELinux-Richtlinienlösung in Web Console

Sobald das Problem gelöst ist, sollte der Fehler im Abschnitt **SELinux Access Control Errors** nicht mehr erscheinen. Wenn die Meldung **No SELinux alerts** angezeigt wird, dann wurden alle Probleme behoben.

SELinux Policy

Enforce policy: **ON**

SELinux Access Control Errors

No SELinux alerts.

Abbildung 5.5: Keine SELinux-Warnmeldungen in Web Console

Literaturhinweise

Manpage **sealert(8)**

► Angeleitete Übung

Untersuchen und Beheben von SELinux-Problemen

In dieser praktischen Übung lernen Sie, Probleme im Zusammenhang mit SELinux-Sicherheitsverweigerungen zu beheben.

Ergebnisse

Sie gewinnen Erfahrung bei der Verwendung der SELinux-Tools zur Problembehebung.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-issues start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Es installiert den **httpd**-Service, konfiguriert die Firewall auf **servera**, damit HTTP-Verbindungen zugelassen werden, und entfernt den SELinux-Kontext für das Verzeichnis **/custom**.

```
[student@workstation ~]$ lab selinux-issues start
```

- ▶ 1. Öffnen Sie auf **workstation** einen Webbrowser und versuchen Sie `http://servera/index.html` anzuzeigen. Sie erhalten eine Fehlermeldung, die Sie darauf hinweist, dass Sie keine Zugriffsberechtigung für diese Datei haben.
- ▶ 2. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- ▶ 3. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- ▶ 4. Zeigen Sie mit dem Befehl **less** den Inhalt von **/var/log/messages** an. Verwenden Sie die Taste **/** und suchen Sie nach **sealert**. Kopieren Sie den vorgeschlagenen Befehl **sealert**, damit er im nächsten Schritt verwendet werden kann. Beenden Sie den Befehl **less** mit der Taste **q**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 28 06:07:03 servera setroubleshoot[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html. For complete SELinux
messages run: sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
Mar 28 06:07:03 servera platform-python[15326]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /custom/index.html.#012#012***** Plugin
catchall (100. confidence) suggests *****#012#012If
you believe that httpd should be allowed getattr access on the index.html file
by default.#012Then you should report this as a bug.#012You can generate a
local policy module to allow this access.#012Do#012allow this access for now by
executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
-X 300 -i my-httpd.pp#012
Mar 28 06:07:04 servera setroubleshoot[15326]: failed to retrieve rpm info for /
custom/index.html
...output omitted...
```

- 5. Führen Sie den vorgeschlagenen Befehl **sealert** aus. Beachten Sie den Quellkontext, die Zielobjekte, die Richtlinie und den Enforcing-Modus.

```
[root@servera ~]# sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
SELinux is preventing /usr/sbin/httpd from getattr access on the file /custom/
index.html.

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed getattr access on the index.html 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: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
Target RPM Packages
Policy RPM              selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled         True
Policy Type             targeted
Enforcing Mode          Enforcing
Host Name               servera.lab.example.com
Platform                Linux servera.lab.example.com
                           4.18.0-67.el8.x86_64 #1 SMP Sat Feb 9 12:44:00
```

```

UTC 2019 x86_64 x86_64
Alert Count 18
First Seen 2019-03-25 19:25:28 CET
Last Seen 2019-03-28 11:07:00 CET
Local ID b1c9cc8f-a953-4625-b79b-82c4f4f1fee3

Raw Audit Messages
type=AVC msg=audit(1553767620.970:16958): avc: denied { getattr } for
pid=15067 comm="httpd" path="/custom/index.html" dev="vda1" ino=4208311
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0

Hash: httpd,httpd_t,default_t,file,getattr

```

- 6. Der Abschnitt **Raw Audit Messages** des Befehls **sealert** enthält Informationen aus /var/log/audit/audit.log. Durchsuchen Sie mit dem Befehl **ausearch** die Datei /var/log/audit/audit.log. Mit der Option -m wird der Meldungstyp gesucht. Mit der Option -ts wird nach der Zeit gesucht. Dieser Eintrag identifiziert den relevanten Prozess und die Datei, die die Warnmeldung ausgelöst haben. Der Prozess ist der **httpd**-Apache-Webserver, die Datei ist **/custom/index.html** und der Kontext ist **system_r:httpd_t**.

```
[root@servera ~]# ausearch -m AVC -ts recent
-----
time->Thu Mar 28 13:39:30 2019
type=PROCTITLE msg=audit(1553776770.651:17000):
    proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553776770.651:17000): arch=c000003e syscall=257
    success=no exit=-13 a0=fffffff9c a1=7f8db803f598 a2=80000 a3=0 items=0 ppid=15063
    pid=15065 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(1553776770.651:17000): avc: denied
    { open } for pid=15065 comm="httpd" path="/custom/index.html"
    dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0
    tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

- 7. Beheben Sie mit den Befehlen **semanage** und **restorecon** das Problem. Der zu verwaltende Kontext ist **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
```

- 8. Versuchen Sie, <http://servera/index.html> erneut aufzurufen. Nun sollte die Meldung **This is SERVERA** erscheinen.

► 9. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab selinux-issues finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab selinux-issues finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Verwalten der SELinux-Sicherheit

Leistungscheckliste

In dieser praktischen Übung beheben Sie ein SELinux-Problem hinsichtlich einer Zugriffsverweigerung. Es gelingt den Systemadministratoren nicht, mit dem neuen Webserver Inhalte an Clients zu senden, wenn SELinux im Enforcing-Modus aktiv ist.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Probleme in Systemprotokolldateien identifizieren
- SELinux-Konfiguration anpassen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Es installiert zudem den httpd-Apache-Server, erstellt einen neuen Document Root für Apache und aktualisiert die Konfigurationsdatei.

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

1. Melden Sie sich als root-Benutzer bei **serverb** an.
2. Starten Sie einen Webbrower auf **workstation** und navigieren Sie zu `http://serverb/lab.html`. Es wird folgende Fehlermeldung angezeigt: **You do not have permission to access /lab.html on this server.**
3. Ermitteln Sie das SELinux-Problem, das Apache davon abhält, die Webinhalte auszugeben.
4. Zeigen Sie den SELinux-Kontext des neuen HTTP Document Root und des ursprünglichen HTTP Document Root an. Beheben Sie das SELinux-Problem, das Apache davon abhält, Webinhalte auszugeben.
5. Überprüfen Sie, ob das SELinux-Problem behoben ist und Apache Webinhalt anzeigen kann.
6. Beenden Sie **serverb**.

Bewertung

Führen Sie auf **workstation** das Skript **lab selinux-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab selinux-review finish** aus, um die praktische Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Verwalten der SELinux-Sicherheit

Leistungscheckliste

In dieser praktischen Übung beheben Sie ein SELinux-Problem hinsichtlich einer Zugriffsverweigerung. Es gelingt den Systemadministratoren nicht, mit dem neuen Webserver Inhalte an Clients zu senden, wenn SELinux im Enforcing-Modus aktiv ist.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Probleme in Systemprotokolldateien identifizieren
- SELinux-Konfiguration anpassen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab selinux-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Es installiert zudem den httpd-Apache-Server, erstellt einen neuen Document Root für Apache und aktualisiert die Konfigurationsdatei.

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

1. Melden Sie sich als root-Benutzer bei **serverb** an.

- 1.1. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 1.2. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

2. Starten Sie einen Webbroweser auf **workstation** und navigieren Sie zu **http://serverb/lab.html**. Es wird folgende Fehlermeldung angezeigt: **You do not have permission to access /lab.html on this server**.
3. Ermitteln Sie das SELinux-Problem, das Apache davon abhält, die Webinhalte auszugeben.

- 3.1. Zeigen Sie mit dem Befehl **less** den Inhalt von **/var/log/messages** an. Verwenden Sie die Taste **/** und suchen Sie nach **sealert**. Beenden Sie den Befehl **less** mit der Taste **q**.

```
[root@serverb ~]# less /var/log/messages
Mar 28 10:19:51 serverb setroubleshoot[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html. For complete SELinux
messages run: sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
Mar 28 10:19:51 serverb platform-python[27387]: SELinux is preventing /usr/sbin/
httpd from getattr access on the file /lab-content/lab.html.#012#012***** Plugin
catchall (100. confidence) suggests *****#012#012If
you believe that httpd should be allowed getattr access on the lab.html file
by default.#012Then you should report this as a bug.#012You can generate a
local policy module to allow this access.#012Do#012allow this access for now by
executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule
-X 300 -i my-httpd.pp#012
```

- 3.2. Führen Sie den vorgeschlagenen Befehl **sealert** aus. Beachten Sie den Quellkontext, die Zielobjekte, die Richtlinie und den Enforcing-Modus.

```
[root@serverb ~]# sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
SELinux is preventing /usr/sbin/httpd from getattr access on the file /lab-
content/lab.html.

***** Plugin catchall (100. confidence) suggests *****

If you believe that httpd should be allowed getattr access on the lab.html 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: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
Target RPM Packages
Policy RPM              selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled         True
Policy Type             targeted
Enforcing Mode          Enforcing
Host Name               serverb.lab.example.com
Platform                Linux serverb.lab.example.com
                           4.18.0-67.el8.x86_64 #1 SMP Sat Feb 9 12:44:00
                           UTC 2019 x86_64 x86_64
```

Kapitel 5 | Verwalten der SELinux-Sicherheit

```

Alert Count          2
First Seen         2019-03-28 15:19:46 CET
Last Seen          2019-03-28 15:19:46 CET
Local ID           8824e73d-3ab0-4caf-8258-86e8792fee2d

Raw Audit Messages
type=AVC msg=audit(1553782786.213:864): avc: denied { getattr } for
pid=15606 comm="httpd" path="/lab-content/lab.html" dev="vda1" ino=8763212
scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0

Hash: httpd,httpd_t,default_t,file,getattr

```

- 3.3. Der Abschnitt **Raw Audit Messages** des Befehls **sealert** enthält Informationen aus **/var/log/audit/audit.log**. Durchsuchen Sie mit dem Befehl **ausearch** die Datei **/var/log/audit/audit.log**. Mit der Option **-m** wird der Meldungstyp gesucht. Mit der Option **ts** wird nach der Zeit gesucht. Dieser Eintrag identifiziert den relevanten Prozess und die Datei, die die Warnmeldung ausgelöst haben. Der Prozess ist der **httpd**-Apache-Webserver, die Datei ist **/lab-content/lab.html** und der Kontext ist **system_r:httpd_t**.

```

[root@serverb ~]# ausearch -m AVC -ts recent
time->Thu Mar 28 15:19:46 2019
type=PROCTITLE msg=audit(1553782786.213:864):
    proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553782786.213:864): arch=c000003e syscall=6 success=no
    exit=-13 a0=7fb900004930 a1=7fb92dfca8e0 a2=7fb92dfca8e0 a3=1 items=0 ppid=15491
    pid=15606 auid=4294967295 uid=48 gid=48 euid=48 suid=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(1553782786.213:864): avc: denied { getattr } for
    pid=15606 comm="httpd" path="/lab-content/lab.html" dev="vda1" ino=8763212
    scontext=system_u:system_r:httpd_t:s0 tcontext=unconfined_u:object_r:default_t:s0
    tclass=file permissive=0

```

4. Zeigen Sie den SELinux-Kontext des neuen HTTP Document Root und des ursprünglichen HTTP Document Root an. Beheben Sie das SELinux-Problem, das Apache davon abhält, Webinhalte auszugeben.
- 4.1. Vergleichen Sie mit **ls -dZ** den Document Root von **/lab-content** und **/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. Erstellen Sie eine Dateikontextregel, durch die der Standardtyp für **/lab-content** und alle darin befindlichen Dateien auf **httpd_sys_content** gesetzt wird.

```

[root@serverb ~]# semanage fcontext -a \
-t httpd_sys_content_t '/lab-content(/.*)?'

```

- 4.3. Legen Sie mit dem Befehl **restorecon** den SELinux-Kontext für die Dateien in **/lab-content** fest.

```
[root@serverb ~]# restorecon -R /lab-content/
```

5. Überprüfen Sie, ob das SELinux-Problem behoben ist und Apache Webinhalt anzeigen kann.

Verwenden Sie Ihren Webbrowser, um den Link <http://serverb/lab.html> zu aktualisieren. Jetzt sollten Webinhalte angezeigt werden.

```
This is the html file for the SELinux final lab on SERVERB.
```

6. Beenden Sie **serverb**.

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

Bewertung

Führen Sie auf **workstation** das Skript **lab selinux-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab selinux-review finish** aus, um die praktische Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Mit den Befehlen **getenforce** und **setenforce** wird der SELinux-Modus eines Systems verwaltet.
- Mit dem Befehl **semanage** werden SELinux-Richtlinienregeln verwaltet. Mit dem Befehl **restorecon** wird der durch die Richtlinie definierte Kontext angewendet.
- Mit booleschen Werten können Sie das Verhalten der SELinux-Richtlinie ändern. Sie können aktiviert oder deaktiviert werden und werden zur Optimierung der Richtlinie verwendet.
- Der Befehl **sealert** zeigt nützliche Informationen zur Problembehebung in SELinux an.

Kapitel 6

Verwalten von Basisspeicher

Ziel

Speichergeräte, Partitionen, Dateisysteme und Swap-Speicher über die Befehlszeile erstellen und verwalten

Ziele

- Speicherpartitionen erstellen, mit Dateisystemen formatieren und mounten
- Swap-Speicher als Ergänzung zum physischen Arbeitsspeicher erstellen und verwalten

Abschnitte

- Hinzufügen von Partitionen, Dateisystemen und dauerhaften Mounts (und angeleitete Übung)
- Verwalten von Swap-Speicher (und angeleitete Übung)

Praktische Übung

Verwalten von Basisspeicher

Hinzufügen von Partitionen, Dateisystemen und dauerhaften Mounts

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, Speicherpartitionen zu erstellen, diese mit Dateisystemen zu formatieren und sie für die Verwendung zu mounten.

Partitionieren einer Festplatte

Durch Partitionieren können Systemadministratoren eine Festplatte in mehrere logische Speichereinheiten, die Partitionen, einteilen. Das Einteilen einer Festplatte in mehrere Partitionen ermöglicht Systemadministratoren, die einzelnen Partitionen für verschiedene Zwecke einzusetzen.

Die Partitionierung einer Festplatte ist beispielsweise in diesen Situationen erforderlich oder nützlich:

- Verfügbaren Speicherplatz für Anwendungen oder Benutzer begrenzen
- Betriebssystem und Programmdateien von den Benutzerdateien trennen
- Separaten Bereich für Speicherauslagerungen (Swap-Speicher) erstellen
- Verwendung des Speicherplatzes begrenzen, um die Leistung von Diagnosetools und Image-Backups zu steigern

MBR-Partitionierungsschema

Seit 1982 gibt das Partitionierungsschema *Master Boot Record (MBR)* vor, auf welche Weise Festplatten auf Systemen, die mit BIOS-Firmware ausgeführt werden, partitioniert werden sollen. Dieses Schema unterstützt maximal vier primäre Partitionen. Mit dem Einsatz von erweiterten und logischen Partitionen können Administratoren auf Linux-Systemen maximal 15 Partitionen erstellen. Da die Daten der Partitionsgröße als 32-Bit-Werte gespeichert werden, erreichen Festplatten, die mit dem MBR-Schema partitioniert sind, eine maximale Festplatten- und Partitionsgröße von 2 TiB.



Abbildung 6.1: MBR-Partitionierung des Speichergeräts /dev/vdb

Da physische Festplatten immer größer werden und SAN-basierte Volumes sogar noch größer sind, stellt die Größenbeschränkung von 2 TiB für Festplatten und Partitionen des MBR-Partitionierungsschemas keine theoretische Grenze mehr, sondern eher ein reales Problem dar, auf das Systemadministratoren in Produktionsumgebungen immer häufiger stoßen. Folglich wird das MBR-Schema nach und nach von dem neuen Schema *GUID Partition Table (GPT)* für die Festplattenpartitionierung abgelöst.

GPT-Partitionierungsschema

Bei Systemen mit *Unified Extensible Firmware Interface (UEFI)*-Firmware ist GPT der Standard für das Anordnen von Partitionstabellen auf physischen Festplatten. GPT ist Bestandteil des UEFI-Standards und löst viele der vom alten MBR-Schema auferlegten Begrenzungen.

Eine GPT bietet maximal 128 Partitionen. Im Gegensatz zu MBR, das 32 Bit zum Speichern der logischen Blockadressen und Größeninformationen verwendet, weist GPT 64 Bit für logische Blockadressen zu. Somit kann GPT Partitionen und Festplatten von bis zu acht Zebibyte (ZiB) oder acht Milliarden Tebibyte aufnehmen.

Neben dem Abbau der Begrenzungen des MBR-Partitionierungsschemas bietet eine GPT weitere Features und Vorteile. Eine GPT verwendet einen *Globally Unique Identifier (GUID)* (global eindeutiger Bezeichner), um jede Festplatte und Partition zu identifizieren. Im Gegensatz zu MBR, das einen Single Point of Failure hat, bietet eine GPT Redundanz der Daten in der Partitionstabelle. Die primäre GPT befindet sich am Anfang der Festplatte, während sich die sekundäre GPT mit der Sicherungskopie am Ende der Festplatte befindet. Eine GPT ermittelt anhand einer Prüfsumme Fehler und Beschädigungen im GPT-Header und der Partitionstabelle.



Abbildung 6.2: GPT-Partitionierung des Speichergeräts /dev/vdb

Verwalten von Partitionen mit Parted

Partitionseditoren sind Programme, mit denen Administratoren Änderungen an Festplattenpartitionen vornehmen können, wie etwa Partitionen erstellen und löschen oder den Partitionstyp ändern. Um diese Vorgänge durchzuführen, können Administratoren den Partitionseditor Parted sowohl für das MBR- als auch für das GPT-Partitionierungsschema verwenden.

Der Befehl **parted** übernimmt den Gerätenamen der gesamten Festplatte als erstes Argument und einen oder mehrere Sub-Befehle. Im folgenden Beispiel wird mit dem Sub-Befehl **print** die Partitionstabelle auf der Festplatte **/dev/vda** angezeigt.

```
[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
```

Wenn Sie keinen Sub-Befehl angeben, öffnet **parted** eine interaktive Sitzung zur Eingabe von Befehlen.

```
[root@host ~]# parted /dev/vda
GNU Parted 3.2
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 ~]#
```

Standardmäßig zeigt **parted** alle Größen in Zehnerpotenzen (KB, MB, GB) an. Sie können diesen Standard mit dem Sub-Befehl **unit**, der die folgenden Parameter akzeptiert, ändern:

- **s** für Sektor
- **B** für Byte
- **MiB, GiB oder TiB** (Zweierpotenzen)
- **MB, GB oder TB** (Zehnerpotenzen)

```
[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
```

Wie im obigen Beispiel gezeigt, können Sie auch mehrere Sub-Befehle (hier **unit** und **print**) in der gleichen Zeile angeben.

Schreiben der Partitionstabelle auf eine neue Festplatte

Um eine neue Festplatte zu partitionieren, müssen Sie zuerst eine Datenträgerbezeichnung schreiben. Die Datenträgerbezeichnung gibt an, welches Partitionsschema verwendet werden soll.



Warnung

Denken Sie daran, dass **parted** die Änderungen unverzüglich vornimmt. Ein Fehler bei **parted** könnte zu Datenverlust führen.

Kapitel 6 | Verwalten von Basisspeicher

Verwenden Sie als Benutzer **root** den folgenden Befehl, um eine MBR-Datenträgerbezeichnung auf eine Festplatte zu schreiben.

```
[root@host ~]# parted /dev/vdb mklabel msdos
```

Verwenden Sie den folgenden Befehl, um eine GPT-Datenträgerbezeichnung zu schreiben.

```
[root@host ~]# parted /dev/vdb mklabel gpt
```



Warnung

Der Sub-Befehl **mklabel** löscht die vorhandene Partitionstabelle. Verwenden Sie **mklabel** nur, wenn Sie beabsichtigen, die Festplatte ohne Rücksicht auf die vorhandenen Daten wiederzuverwenden. Wenn eine neue Bezeichnung die Partitionsgrenzen ändert, kann auf alle Daten in den vorhandenen Dateisystemen nicht mehr zugegriffen werden.

Erstellen von MBR-Partitionen

Das Erstellen einer MBR-Festplattenpartition umfasst mehrere Schritte:

1. Geben Sie das Festplattengerät an, auf dem die Partition erstellt wird.

Führen Sie als **root**-Benutzer den Befehl **parted** aus und geben Sie den Namen des Festplattengeräts als Argument an. Dadurch wird der Befehl **parted** im interaktiven Modus gestartet und eine Befehlseingabeaufforderung angezeigt.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. Erstellen Sie mit dem Sub-Befehl **mkpart** eine neue primäre oder erweiterte Partition.

```
(parted) mkpart
Partition type? primary/extended? primary
```



Anmerkung

Wenn Sie auf einer mit MBR partitionierten Festplatte mehr als vier Partitionen benötigen, erstellen Sie drei primäre Partitionen und eine erweiterte Partition. Diese erweiterte Partition dient als Container, in dem Sie mehrere logische Partitionen erstellen können.

3. Geben Sie den Typ des Dateisystems an, den Sie auf der Partition erstellen möchten, z. B. **xfs** oder **ext4**. Dadurch wird das Dateisystem nicht auf der Partition erstellt. Es handelt sich lediglich um die Angabe des Partitionstyps.

```
File system type? [ext2]? xfs
```

Kapitel 6 | Verwalten von Basisspeicher

Um die Liste der unterstützten Dateisystemtypen abzurufen, verwenden Sie den folgenden Befehl:

```
[root@host ~]# parted /dev/vdb help mkpart
mkpart PART-TYPE [FS-TYPE] START END      make a partition

PART-TYPE is one of: primary, logical, extended
FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
hp-ufs, sun-ufs, xfs, apfs2, apfs1, afs, amufs5, amufs4, amufs3,
amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
START and END are disk locations, such as 4GB or 10%. Negative values
count from the end of the disk. For example, -1s specifies exactly the
last sector.

'mkpart' makes a partition without creating a new file system on the
partition. FS-TYPE may be specified to set an appropriate partition
ID.
```

4. Geben Sie den Sektor auf der Festplatte an, der den Startpunkt der neuen Partition darstellt.

Start? **2048s**

Beachten Sie das Suffix **s** zur Angabe des Werts in Sektoren. Sie können auch die Suffixe **MiB**, **GiB**, **TiB**, **MB**, **GB** oder **TB** verwenden. Wenn Sie kein Suffix angeben, ist **MB** die Standardeinstellung. **parted** kann den angegebenen Wert runden, um Festplattenbeschränkungen zu entsprechen.

Wenn **parted** startet, ruft er die Festplattentopologie des Geräts ab. Beispielsweise ist die physische Blockgröße der Festplatte normalerweise ein Parameter, den **parted** erfasst. Mit dieser Information stellt **parted** sicher, dass die von Ihnen angegebene Startposition die Partition korrekt an der Festplattenstruktur ausrichtet. Die korrekte Ausrichtung der Partitionen ist wichtig, um eine optimale Leistung zu erzielen. Wenn die Startposition zu einer falsch ausgerichteten Partition führt, zeigt **parted** eine Warnung an. Bei den meisten Festplatten ist ein Startsektor, der ein Vielfaches von 2048 ist, eine sichere Annahme.

5. Geben Sie den Plattensektor an, an dem die neue Partition enden soll.

End? **1000MB**

Mit **parted** können Sie die Größe Ihrer Partition nicht direkt angeben, Sie können sie jedoch mit der folgenden Formel schnell berechnen:

Size = End - Start

Sobald Sie die Endposition angeben, aktualisiert **parted** die Partitionstabelle auf der Festplatte mit den neuen Partitionsdetails.

6. Beenden Sie **parted**.

```
(parted) quit  
Information: You may need to update /etc/fstab.  
[root@host ~]#
```

- Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet darauf, dass das System die neue Partition erkennt und die zugehörige Gerätedatei im Verzeichnis **/dev** erstellt. Der Befehl kehrt erst zurück, wenn der Vorgang abgeschlossen ist.

```
[root@host ~]# udevadm settle  
[root@host ~]#
```

Alternativ zum interaktiven Modus können Sie die Partition auch wie folgt erstellen:

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

Erstellen von GPT-Partitionen

Das GPT-Schema verwendet ebenfalls den Befehl **parted** zum Erstellen neuer Partitionen:

- Geben Sie das Festplattengerät an, auf dem die Partition erstellt wird.

Führen Sie als Benutzer **root** den Befehl **parted** mit der Festplatte als einzigem Argument aus, um **parted** im interaktiven Modus mit einer Befehlseingabeaufforderung zu starten.

```
[root@host ~]# parted /dev/vdb  
GNU Parted 3.2  
Using /dev/vdb  
Welcome to GNU Parted! Type 'help' to view a list of commands.  
(parted)
```

- Erstellen Sie mit dem Sub-Befehl **mkpart** die neue Partition.

Beim GPT-Schema erhält jede Partition einen Namen.

```
(parted) mkpart  
Partition name? []? usersdata
```

- Geben Sie den Typ des Dateisystems an, den Sie auf der Partition erstellen möchten, z. B. **xfs** oder **ext4**. Dadurch wird das Dateisystem nicht auf der Partition erstellt. Es handelt sich lediglich um die Angabe des Partitionstyps.

```
File system type? [ext2]? xfs
```

- Geben Sie den Sektor auf der Festplatte an, der den Startpunkt der neuen Partition darstellt.

```
Start? 2048s
```

- Geben Sie den Plattensektor an, an dem die neue Partition enden soll.

```
End? 1000MB
```

Sobald Sie die Endposition angeben, aktualisiert **parted** die Partitionstabelle auf der Festplatte mit den neuen Partitionsdetails.

6. Beenden Sie **parted**.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

7. Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet darauf, dass das System die neue Partition erkennt und die zugehörige Gerätedatei im Verzeichnis **/dev** erstellt. Der Befehl kehrt erst zurück, wenn der Vorgang abgeschlossen ist.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

Alternativ zum interaktiven Modus können Sie die Partition auch wie folgt erstellen:

```
[root@host ~]# parted /dev/vdb mkpart usersdata xfs 2048s 1000MB
```

Löschen von Partitionen

Die folgenden Schritte gelten sowohl für das MBR- als auch für das GPT-Partitionierungsschema.

1. Geben Sie die Festplatte an, die die zu entfernende Partition enthält.

Führen Sie als Benutzer **root** den Befehl **parted** mit der Festplatte als einzigem Argument aus, um **parted** im interaktiven Modus mit einer Befehlseingabeaufforderung zu starten.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. Ermitteln Sie die Partitionsnummer der zu löschenen Partition.

```
(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
```

3. Löschen Sie die Partition.

```
(parted) rm 1
```

Der Sub-Befehl **rm** löscht die Partition sofort aus der Partitionstabelle auf der Festplatte.

4. Beenden Sie **parted**.

```
(parted) quit  
Information: You may need to update /etc/fstab.  
[root@host ~]#
```

Erstellen von Dateisystemen

Nach der Erstellung eines Blockgeräts müssen Sie im nächsten Schritt ein Dateisystem hinzufügen. Red Hat Enterprise Linux unterstützt viele verschiedene Dateisystemtypen, die beiden verbreitetsten sind aber XFS und ext4. Anaconda, das Installationsprogramm für Red Hat Enterprise Linux, verwendet standardmäßig XFS.

Legen Sie als **root** mit dem Befehl **mkfs.xfs** ein XFS-Dateisystem auf ein Blockgerät an. Verwenden Sie für ext4 **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  
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
```

Mounten von Dateisystemen

Nachdem Sie das Dateisystem hinzugefügt haben, besteht der letzte Schritt darin, das Dateisystem in einem Verzeichnis in der Verzeichnisstruktur zu mounten. Nach dem Mounten eines Dateisystems in eine Verzeichnishierarchie verfügen Userspace-Dienstprogramme über Lese- und Schreibzugriff auf dem Gerät.

Manuelles Mounten von Dateisystemen

Administratoren ordnen mit dem Befehl **mount** das Gerät manuell einem Verzeichnisspeicherort oder Mount-Punkt zu. Der Befehl **mount** erwartet das Gerät, den Mount-Punkt und optional Dateisystemoptionen als Argumente. Die Dateisystemoptionen passen das Verhalten des Dateisystems an.

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

Sie können den Befehl **mount** auch zum Anzeigen der aktuell gemounteten Dateisysteme, der Mount-Punkte und der Optionen verwenden.

```
[root@host ~]# mount | grep vdb1
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Dauerhaftes Mounten von Dateisystemen

Durch manuelles Mounten eines Dateisystems lässt sich sehr gut überprüfen, ob auf ein formatiertes Gerät zugegriffen werden kann und ob es wie gewünscht arbeitet. Beim erneuten Booten des Servers wird das Dateisystem jedoch nicht automatisch erneut in die Verzeichnisstruktur gemountet. Die Daten sind im Dateisystem intakt, aber Benutzer können nicht darauf zugreifen.

Um sicherzustellen, dass das System das Dateisystem beim Booten automatisch mountet, fügen Sie der Datei **/etc/fstab** einen Eintrag hinzu. In dieser Konfigurationsdatei werden die Dateisysteme aufgelistet, die beim Booten des Systems gemountet werden sollen.

/etc/fstab ist eine durch Leerzeichen getrennte Datei mit sechs Feldern pro Zeile.

```
[root@host ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Feb 13 16:39:59 2019
#
# 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
```

Wenn Sie der Datei **/etc/fstab** einen Eintrag hinzufügen oder daraus entfernen, führen Sie den Befehl **systemctl daemon-reload** aus oder booten Sie den Server neu, damit **systemd** die neue Konfiguration registriert.

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

Das erste Feld gibt das Gerät an. In diesem Beispiel wird die UUID verwendet, um das Gerät anzugeben. Dateisysteme erstellen und speichern die UUID zum Zeitpunkt der Erstellung in ihrem Superblock. Alternativ können Sie die Gerätedatei, wie z. B. **/dev/vdb1**, verwenden.



Anmerkung

Die UUID wird bevorzugt verwendet, weil sich die Bezeichner von Blockgeräten in bestimmten Szenarien ändern können, so wie etwa ein Cloud-Provider, der die zugrunde liegende Speicherschicht eines virtuellen Rechners ändert, oder wenn die Festplatten bei jedem Booten des Systems in einer anderen Reihenfolge erkannt werden. Der Name der Blockgerätedatei kann sich ändern, die UUID bleibt jedoch im Superblock des Dateisystems konstant.

Verwenden Sie den Befehl **lsblk --fs**, um die an den Rechner angeschlossenen Blockgeräte abzufragen und die UUIDs des Dateisystems abzurufen.

```
[root@host ~]# lsblk --fs
  NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
  sr0
  vda
    └─vda1 xfs      a8063676-44dd-409a-b584-68be2c9f5570 /
  vdb
    └─vdb1 xfs      7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata
```

Das zweite Feld ist der Verzeichnis-Mount-Punkt, von dem aus auf das Blockgerät in der Verzeichnisstruktur zugegriffen werden kann. Der Mount-Punkt muss vorhanden sein. Wenn er nicht vorhanden ist, erstellen Sie ihn mit dem Befehl **mkdir**.

Das dritte Feld enthält den Typ des Dateisystems, wie z. B. **xfs** oder **ext4**.

Das vierte Feld ist die durch Komma getrennte Liste der Optionen für das Gerät. **defaults** ist eine Reihe von häufig verwendeten Optionen. Auf der **mount(8)** Manpage sind die anderen verfügbaren Optionen dokumentiert.

Das fünfte Feld wird vom Befehl **dump** für das Backup des Geräts verwendet. Andere Backup-Anwendungen verwenden dieses Feld normalerweise nicht.

Das letzte Feld, das **fsck**-Reihenfolgenfeld, legt fest, ob der Befehl **fsck** beim Booten des Systems ausgeführt werden soll, um zu überprüfen, ob die Dateisysteme bereinigt sind. Der Wert in diesem Feld gibt die Reihenfolge an, in der **fsck** ausgeführt werden soll. Legen Sie dieses Feld für XFS-Dateisysteme auf **0** fest, weil XFS den Dateisystemstatus nicht mit **fsck** überprüft. Legen Sie für ext4-Dateisysteme den Wert auf **1** für das Root-Dateisystem und auf **2** für die anderen ext4-Dateisysteme fest. Auf diese Weise verarbeitet **fsck** zuerst das Root-Dateisystem und prüft dann die Dateisysteme gleichzeitig auf separaten Datenträgern sowie die Dateisysteme auf demselben Datenträger nacheinander.



Anmerkung

Ein falscher Eintrag in der Datei **/etc/fstab** kann dazu führen, dass der Rechner nicht mehr bootfähig ist. Administratoren müssen überprüfen, ob der Eintrag gültig ist, indem sie das neue Dateisystem ummounten und den Befehl **mount /mountpoint** verwenden, der **/etc/fstab** liest, um das Dateisystem wieder zu mounten. Wenn der Befehl **mount** einen Fehler zurückgibt, muss dieser korrigiert werden, bevor das System erneut gebootet wird.

Alternativ können Sie mit dem Befehl **findmnt --verify** die Datei **/etc/fstab** steuern.



Literaturhinweise

info parted(*GNU Parted Benutzerhandbuch*)

Manpages **parted**(8), **mkfs**(8), **mount**(8), **lsblk**(8) und **fstab**(5)

Weitere Informationen finden Sie im Handbuch *Configuring and managing file systems* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_file_systems/

► Angeleitete Übung

Hinzufügen von Partitionen, Dateisystemen und dauerhaften Mounts

In dieser Übung erstellen Sie eine Partition auf einem neuen Speichergerät, formatieren die Partition mit einem XFS-Dateisystem, konfigurieren das Dateisystem so, dass es beim Booten gemountet wird, und mounten es für die Verwendung.

Ergebnisse

Sie sollten in der Lage sein, mit **parted**, **mkfs.xfs** und anderen Befehlen eine Partition auf einer neuen Festplatte zu erstellen, zu formatieren und dauerhaft zu mounten.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab storage-partitions start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Der Befehl bereitet auch die zweite Festplatte auf **servera** für die Übung vor.

```
[student@workstation ~]$ lab storage-partitions start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Verwenden Sie bei Aufforderung **student** als Passwort.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- 3. Erstellen Sie mit **parted** auf der Festplatte **/dev/vdb** eine neue Datenträgerbezeichnung mit dem Typ **msdos**, um diese neue Festplatte für das MBR-Partitionierungsschema vorzubereiten.

```
[root@servera ~]# parted /dev/vdb mklabel msdos  
Information: You may need to update /etc/fstab.
```

Kapitel 6 | Verwalten von Basisspeicher

- 4. Fügen Sie eine neue Partition mit einer Größe von 1 GB hinzu. Beginnen Sie zur korrekten Ausrichtung die Partition bei Sektor 2048. Legen Sie den Partitionsdateisystemtyp auf XFS fest.

4.1. Verwenden Sie den interaktiven Modus von **parted**, um die Partition zu erstellen.

```
[root@servera ~]# parted /dev/vdb
GNU Parted 3.2
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.
```

Da die Partition bei Sektor 2048 beginnt, legt der vorherige Befehl die Endposition auf 1001 MB fest, um eine Partitionsgröße von 1000 MB (1 GB) zu erhalten.

Alternativ können Sie denselben Vorgang mit dem folgenden nicht interaktiven Befehl ausführen: **parted /dev/vdb mkpart primary xfs 2048s 1001MB**

4.2. Überprüfen Sie Ihre Arbeit, indem Sie die Partitionen auf **/dev/vdb** auflisten.

```
[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
```

4.3. Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet darauf, dass das System die neue Partition registriert, und kehrt zurück, wenn dies abgeschlossen ist.

```
[root@servera ~]# udevadm settle
```

- 5. Formatieren Sie die neue Partition mit dem XFS-Dateisystem.

```
[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
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
```

- 6. Konfigurieren Sie das neue Dateisystem so, dass es unter **/archive** dauerhaft gemountet wird.
- 6.1. Verwenden Sie den Befehl **mkdir**, um das Verzeichnis **/archive** für den Mount-Punkt zu erstellen.

```
[root@servera ~]# mkdir /archive
```

- 6.2. Verwenden Sie den Befehl **lsblk** mit der Option **--fs**, um die UUID des Geräts **/dev/vdb1** zu ermitteln.

```
[root@servera ~]# lsblk --fs /dev/vdb
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
vdb
└─vdb1 xfs      e3db1abe-6d96-4faa-a213-b96a6f85dcc1
```

Die UUID in der vorherigen Ausgabe weicht möglicherweise auf Ihrem System ab.

- 6.3. Fügen Sie **/etc/fstab** einen Eintrag hinzu. Ersetzen Sie im folgenden Inhalt die UUID durch die, die Sie im vorherigen Schritt ermittelt haben.

```
...output omitted...
UUID=e3db1abe-6d96-4faa-a213-b96a6f85dcc1 /archive xfs defaults 0 0
```

- 6.4. Aktualisieren Sie **systemd**, damit das System die neue **/etc/fstab**-Konfiguration registriert.

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

- 6.5. Führen Sie den Befehl **mount /archive** aus, um das neue Dateisystem mithilfe des neuen Eintrags, den Sie der Datei **/etc/fstab** hinzugefügt haben, zu mounten.

```
[root@servera ~]# mount /archive
```

- 6.6. Prüfen Sie, ob das neue Dateisystem unter **/archive** gemountet ist.

```
[root@servera ~]# mount | grep /archive
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

- 7. Booten Sie **servera** neu. Melden Sie sich nach dem Booten des Servers an und prüfen Sie, ob **/dev/vdb1** unter **/archive** gemountet ist. Melden Sie sich von **servera** ab, wenn Sie fertig sind.

- 7.1. Booten Sie **servera** neu.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- 7.2. Warten Sie ein paar Minuten, bis **servera** neu gebootet ist, und melden Sie sich als Benutzer **student** an.

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

7.3. Prüfen Sie, ob **/dev/vdb1** unter **/archive** gemountet ist.

```
[student@servera ~]$ mount | grep /archive  
/dev/vdb1 on /archive type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

7.4. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab storage-partitions finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab storage-partitions finish
```

Hiermit ist die angeleitete Übung beendet.

Verwalten des Swap-Speichers

Ziele

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, Swap-Speicher zu erstellen und zu verwalten, um den physischen Arbeitsspeicher zu ergänzen.

Einführung in Konzepte für Swap-Speicher

Ein Swap-Speicher ist ein Bereich auf der Festplatte, der vom Verwaltungsteilsystem des Linux-Kernel-Speichers kontrolliert wird. Der Kernel verwendet Swap-Speicher, um das System-RAM zu ergänzen, indem inaktive Seiten des Arbeitsspeichers in den Swap-Speicher ausgelagert werden. Das System-RAM und der Swap-Speicher bilden gemeinsam den *virtuellen Speicher*.

Wenn die Speicherauslastung auf einem System das definierte Limit überschreitet, durchsucht der Kernel das RAM nach Speicherseiten, die Prozessen zugewiesen sind und sich im Leerlauf befinden. Der Kernel schreibt Seiten, die sich im Leerlauf befinden, in den Swap-Bereich und weist die RAM-Seiten anderen Prozessen neu zu. Wenn ein Programm Zugriff auf eine Seite auf der Festplatte anfordert, sucht der Kernel eine andere inaktive Speicherseite, schreibt sie auf die Festplatte und ruft die benötigte Seite wieder aus dem Swap-Bereich ab.

Da sich die Swap-Bereiche auf der Festplatte befinden, ist Swap im Vergleich zu RAM langsamer. Obwohl der Swap-Speicher zur Erweiterung des System-RAM verwendet wird, sollten Sie ihn nicht als nachhaltige Lösung für unzureichendes RAM für Ihr Workload betrachten.

Festlegen der Größe des Swap-Speichers

Administratoren sollten die Größe des Swap-Speichers basierend auf dem Arbeitsspeicher-Workload des Systems festlegen. Anwendungsanbieter geben manchmal Empfehlungen zu diesem Thema. Die folgende Tabelle enthält einige Hinweise, die sich auf den gesamten physischen Arbeitsspeicher beziehen.

Empfehlungen für RAM und Swap-Speicher

RAM	Swap-Speicher	Swap-Speicher, wenn Ruhezustand zugelassen
2 GiB oder weniger	Zweimal das RAM	Dreimal das RAM
Zwischen 2 GiB und 8 GiB	Gleich wie das RAM	Zweimal das RAM
Zwischen 8 GiB und 64 GiB	Mindestens 4 GiB	1,5-facher RAM
Mehr als 64 GiB	Mindestens 4 GiB	Ruhezustand wird nicht empfohlen

Die Ruhezustand-Funktion für Laptops und Desktops verwendet den Swap-Speicher, um den RAM-Inhalt zu speichern, bevor das System ausgeschaltet wird. Wenn Sie das System wieder einschalten, stellt der Kernel den RAM-Inhalt aus dem Swap-Speicher wieder her und benötigt keinen vollständigen Bootvorgang. Für diese Systeme muss der Swap-Speicher größer als das RAM sein.

Der Knowledgebase-Artikel im Abschnitt „Referenz“ am Ende dieses Abschnitts enthält weitere Hinweise zur Größenbestimmung des Swap-Speichers.

Erstellen eines Swap-Speichers

Um einen Swap-Speicher zu erstellen, müssen Sie Folgendes ausführen:

- Erstellen Sie eine Partition mit dem Dateisystemtyp **linux-swap**.
- Legen Sie eine Swap-Signatur auf dem Gerät fest.

Erstellen einer Swap-Partition

Erstellen Sie mit **parted** eine Partition in der gewünschten Größe und legen Sie den Dateisystemtyp auf **linux-swap** fest. In der Vergangenheit haben die Tools anhand des Dateisystemtyps der Partition festgelegt, ob das Gerät aktiviert werden soll; dies ist aber nicht mehr der Fall. Auch wenn der Dateisystemtyp der Partition nicht mehr von Dienstprogrammen verwendet wird, kann der Administrator anhand des festgelegten Typs rasch den Zweck der Partition bestimmen.

Im folgenden Beispiel wird eine 256 MB große Partition erstellt.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
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 ~]#
```

Führen Sie nach dem Erstellen der Partition den Befehl **udevadm settle** aus. Dieser Befehl wartet, bis das System die neue Partition erkannt und die zugehörige Gerätedatei **/dev** erstellt hat. Der Befehl kehrt erst zurück, wenn der Vorgang abgeschlossen ist.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

Formatieren des Geräts

Der Befehl **mkswap** wendet eine Swap-Signatur auf das Gerät an. Im Gegensatz zu anderen Formatierungsdienstprogramme schreibt **mkswap** einen einzelnen Datenblock an den Anfang des Geräts und beläßt den Rest des Geräts unformatiert, damit der Kernel diesen Bereich zum Speichern von Speicherseiten verwendet kann.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

Aktiveren eines Swap-Speichers

Mit dem Befehl **swapon** können Sie einen formatierten Swap-Speicher aktivieren.

Verwenden Sie **swapon** mit dem Gerät als Parameter oder verwenden Sie **swapon -a**, um alle in der Datei **/etc/fstab** aufgeführten Swap-Speicher zu aktivieren. Verwenden Sie die Befehle **swapon --show** und **free** zur Überprüfung der verfügbaren Swap-Speicher.

```
[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
```

Sie können Swap-Speicher mit dem Befehl **swapoff** deaktivieren. Wenn der Swap-Speicher Seiten enthält, versucht **swapoff**, diese Seiten in andere aktive Swap-Speicher oder wieder in den Arbeitsspeicher zu verschieben. Wenn Daten nicht an anderen Stellen geschrieben werden können, schlägt der Befehl **swapoff** mit einem Fehler fehl und die Swap-Speicher bleiben aktiv.

Dauerhaftes Aktivieren von Swap-Speicher

Um einen Swap-Speicher bei jedem Booten zu aktivieren, fügen Sie der Datei **/etc/fstab** einen Eintrag hinzu. Das folgende Beispiel zeigt eine typische Zeile in **/etc/fstab** für den oben erstellten Swap-Speicher.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881  swap  swap  defaults  0 0
```

Kapitel 6 | Verwalten von Basisspeicher

Das Beispiel verwendet die UUID als *erstes Feld*. Wenn Sie das Gerät formatieren, zeigt der Befehl **mkswap** diese UUID an. Wenn Sie die Ausgabe von **mkswap** nicht notiert haben, verwenden Sie den Befehl **lsblk - -fs**. Alternativ können Sie auch den Gerätenamen im ersten Feld verwenden.

Das *zweite Feld* ist in der Regel für den Mount-Punkt reserviert. Für Swap-Geräte, auf die nicht über die Verzeichnisstruktur zugegriffen werden kann, übernimmt dieses Feld den Platzhalterwert **swap**. Die **fstab(5)**-Manpage verwendet den Platzhalterwert **none**. Der Wert **swap** ermöglicht jedoch aussagekräftigere Fehlermeldungen für den Fall, dass Probleme auftreten.

Das *dritte Feld* ist der Dateisystemtyp. Der Dateisystemtyp für Swap-Speicher lautet **swap**.

Das *vierte Feld* steht für Optionen zur Verfügung. Das Beispiel verwendet die Option **defaults**. Die Option **defaults** enthält die Mount-Option **auto**, das bedeutet, den Swap-Speicher automatisch beim Booten des Systems zu aktivieren.

Bei den beiden letzten Feldern handelt es sich um das **dump**-Flag und die **fsck**-Reihenfolge. Swap-Speicher benötigen weder eine Sicherung noch eine Überprüfung des Dateisystems und daher sollten dieser Felder auf null gesetzt werden.

Wenn Sie der Datei **/etc/fstab** einen Eintrag hinzufügen oder daraus entfernen, führen Sie den Befehl **systemctl daemon-reload** aus oder booten Sie den Server neu, damit systemd die neue Konfiguration registriert.

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

Festlegen der Swap-Speicherpriorität

Standardmäßig verwendet das System Swap-Speicher der Reihe nach, d. h. der Kernel verwendet den ersten aktivierten Swap-Speicher, bis er voll ist, dann den zweiten usw. Sie können jedoch für jeden Swap-Speicher eine Priorität definieren, um eine bestimmte Reihenfolge zu erzwingen.

Mit der Option **pri** in **/etc/fstab** legen Sie die Priorität fest. Der Kernel verwendet zuerst den Swap-Speicher mit der höchsten Priorität. Die Standardpriorität ist -2.

Das folgende Beispiel zeigt drei Swap-Speicher, die in **/etc/fstab** definiert sind. Der Kernel verwendet den letzten Eintrag mit der Priorität **pri=10** zuerst. Wenn dieser Speicher voll ist, wird der zweite Eintrag mit der Priorität **pri=4** verwendet. Schließlich wird der erste Eintrag verwendet, der eine Standardpriorität von -2 hat.

```
UUID=af30cbb0-3866-466a-825a-58889a49ef33    swap    swap    defaults  0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881    swap    swap    pri=4      0 0
UUID=fb7fa60-b781-44a8-961b-37ac3ef572bf    swap    swap    pri=10     0 0
```

Mit **swapon --show** zeigen Sie die Swap-Speicherprioritäten an.

Wenn Swap-Speicher die gleiche Priorität haben, schreibt der Kernel im Round-Robin-Modus in die Speicher.



Literaturhinweise

Manpages **mkswap(8)**, **swapon(8)**, **swapoff(8)**, **mount(8)** und **parted(8)**

Knowledgebase: What is the recommended swap size for Red Hat platforms?

<https://access.redhat.com/solutions/15244>

► Angeleitete Übung

Verwalten des Swap-Speichers

In dieser Übung erstellen und formatieren Sie eine Partition, die als Swap-Speicher verwendet werden soll, formatieren sie als Swap und aktivieren sie dauerhaft.

Ergebnisse

Sie sollten in der Lage sein, eine Partition und einen Swap-Speicher auf einer Festplatte mit dem GPT-Partitionierungsschema zu erstellen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab storage-swap start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Der Befehl bereitet auch die zweite Festplatte auf **servera** für die Übung vor.

```
[student@workstation ~]$ lab storage-swap start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Verwenden Sie bei Aufforderung **student** als Passwort.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Untersuchen Sie mit dem Befehl **parted** die Festplatte **/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: gpt
Partition Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049KB 1001MB 1000MB          data
```

Kapitel 6 | Verwalten von Basisspeicher

Beachten Sie, dass die Festplatte bereits über eine Partitionstabelle verfügt und das GPT-Partitionierungsschema verwendet. Außerdem ist bereits eine 1 GB große Partition vorhanden.

- 4. Fügen Sie eine neue Partition mit einer Größe von 500 MB hinzu, die als Swap-Speicher verwendet werden soll. Legen Sie den Partitionstyp auf **linux-swap** fest.

- 4.1. Erstellen Sie die Partition mit **parted**. Da die Festplatte das GPT-Partitionierungsschema verwendet, müssen Sie der Partition einen Namen geben. Nennen Sie sie **myswap**.

```
[root@servera ~]# parted /dev/vdb mkpart myswap linux-swap \
1001MB 1501MB
Information: You may need to update /etc/fstab.
```

Beachten Sie im vorherigen Befehl, dass die Startposition, 1001 MB, das Ende der vorhandenen ersten Partition ist. Auf diese Weise stellt **parted** sicher, dass die neue Partition unmittelbar und ohne Lücke der vorherigen folgt.

Da die Partition bei der Position 1001 MB beginnt, legt der Befehl die Endposition auf 1501 MB fest, damit eine Partition in der Größe von 500 MB entsteht.

- 4.2. Überprüfen Sie Ihre Arbeit, indem Sie die Partitionen auf **/dev/vdb** auflisten.

```
[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  999MB   data
 2      1001MB  1501MB  499MB   myswap  swap
```

Die Größe der neuen Partition beträgt nicht genau 500 MB. Und zwar weil **parted** die Partition an das Festplattenlayout anpassen muss.

- 4.3. Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet darauf, dass das System die neue Partition registriert, und kehrt zurück, wenn dies abgeschlossen ist.

```
[root@servera ~]# udevadm settle
```

- 5. Initialisieren Sie die neu erstellte Partition als Swap-Speicher.

```
[root@servera ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 476 MiB (499118080 bytes)
no label, UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328
```

- 6. Aktivieren Sie den neu erstellten Swap-Speicher.

- 6.1. Zeigen Sie mit dem Befehl **swapon --show**, dass durch das Erstellen und Initialisieren von Swap-Speicher dieser noch nicht für die Verwendung aktiviert ist.

```
[root@servera ~]# swapon --show
```

- 6.2. Aktivieren Sie den neu erstellten Swap-Speicher.

```
[root@servera ~]# swapon /dev/vdb2
```

- 6.3. Überprüfen Sie, ob der neu erstellte Swap-Speicher jetzt verfügbar ist.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

- 6.4. Deaktivieren Sie den Swap-Speicher.

```
[root@servera ~]# swapoff /dev/vdb2
```

- 6.5. Vergewissern Sie sich, dass der Swap-Speicher deaktiviert ist.

```
[root@servera ~]# swapon --show
```

- 7. Konfigurieren Sie den neuen Swap-Speicher so, dass er beim Booten des Systems aktiviert wird.

- 7.1. Verwenden Sie den Befehl **lsblk** mit der Option **--fs**, um die UUID des Geräts **/dev/vdb2** zu ermitteln.

```
[root@servera ~]# lsblk --fs /dev/vdb2
NAME FSTYPE LABEL UUID                                     MOUNTPOINT
vdb2 swap          cb7f71ca-ee82-430e-ad4b-7dda12632328
```

Die UUID in der vorherigen Ausgabe weicht möglicherweise auf Ihrem System ab.

- 7.2. Fügen Sie **/etc/fstab** einen Eintrag hinzu. Ersetzen Sie im folgenden Befehl die UUID durch die, die Sie im vorherigen Schritt ermittelt haben.

```
...output omitted...
UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328  swap  swap  defaults  0 0
```

- 7.3. Aktualisieren Sie **systemd**, damit das System die neue **/etc/fstab**-Konfiguration registriert.

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

- 7.4. Aktivieren Sie den Swap-Speicher, indem Sie den gerade zu **/etc/fstab** hinzugefügten Eintrag verwenden.

```
[root@servera ~]# swapon -a
```

- 7.5. Überprüfen Sie, ob der neue Swap-Speicher aktiviert ist.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

- 8. Booten Sie **servera** neu. Melden Sie sich nach dem Booten des Servers an und überprüfen Sie, ob der Swap-Speicher aktiviert ist. Melden Sie sich von **servera** ab, wenn Sie fertig sind.

8.1. Booten Sie **servera** neu.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

8.2. Warten Sie ein paar Minuten, bis **servera** neu gebootet ist, und melden Sie sich als Benutzer **student** an.

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

8.3. Überprüfen Sie, ob der Swap-Speicher aktiviert ist.

```
[root@servera ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 476M   0B   -2
```

8.4. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab storage-swap finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab storage-swap finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Verwalten von Basisspeicher

Leistungscheckliste

In dieser Übung erstellen Sie mehrere Partitionen auf einer neuen Festplatte, formatieren einige davon mit Dateisystemen, mounten diese und aktivieren einige davon als Swap-Speicher.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Partitionen mit dem Befehl **parted** anzeigen und erstellen
- Neue Dateisysteme auf Partitionen erstellen und dauerhaft mounten
- Swap-Speicher erstellen und beim Booten aktivieren

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab storage-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Der Befehl bereitet auch die zweite Festplatte auf **serverb** für die Übung vor.

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

1. Auf **serverb** sind neue Festplatten verfügbar. Erstellen Sie auf der ersten neuen Festplatte eine GPT-Partition mit 2 GB und dem Namen **Backup**. Da es schwierig sein kann, die exakte Größe festzulegen, ist eine Größe zwischen 1,8 GB und 2,2 GB akzeptabel. Legen Sie auf dieser Partition das richtige Dateisystem zum Hosten eines XFS-Dateisystems fest.
Das Passwort für das Benutzerkonto **student** auf **serverb** lautet **student**. Dieser Benutzer hat über **sudo** vollen **root**-Zugriff.
2. Formatieren Sie die 2 GB große Partition mit einem XFS-Dateisystem und mounten Sie es dauerhaft unter **/backup**.
3. Erstellen Sie auf derselben neuen Festplatte zwei GPT-Partitionen mit jeweils 512 MB mit den Namen **swap1** und **swap2**. Eine Größe zwischen 460 MB und 564 MB ist akzeptabel. Legen Sie den richtigen Dateisystemtyp für diese Partitionen zum Hosten der Swap-Speicher fest.
4. Initialisieren Sie die beiden 512 MiB großen Partitionen als Swap-Speicher und konfigurieren Sie sie so, dass sie beim Booten aktiviert werden. Legen Sie den Swap-Speicher auf der Partition **swap2** als bevorzugten Swap-Speicher fest.
5. Booten Sie **serverb** neu, um Ihre Arbeit zu überprüfen. Überprüfen Sie, ob das System die erste Partition automatisch unter **/backup** mountet. Vergewissern Sie sich außerdem, dass das System die beiden Swap-Speicher aktiviert.
Melden Sie sich von **serverb** ab, wenn Sie fertig sind.

Bewertung

Führen Sie auf **workstation** das Skript **lab storage-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab storage-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

► Lösung

Verwalten von Basisspeicher

Leistungscheckliste

In dieser Übung erstellen Sie mehrere Partitionen auf einer neuen Festplatte, formatieren einige davon mit Dateisystemen, mounten diese und aktivieren einige davon als Swap-Speicher.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Partitionen mit dem Befehl **parted** anzeigen und erstellen
- Neue Dateisysteme auf Partitionen erstellen und dauerhaft mounten
- Swap-Speicher erstellen und beim Booten aktivieren

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab storage-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Der Befehl bereitet auch die zweite Festplatte auf **serverb** für die Übung vor.

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

1. Auf **serverb** sind neue Festplatten verfügbar. Erstellen Sie auf der ersten neuen Festplatte eine GPT-Partition mit 2 GB und dem Namen **Backup**. Da es schwierig sein kann, die exakte Größe festzulegen, ist eine Größe zwischen 1,8 GB und 2,2 GB akzeptabel. Legen Sie auf dieser Partition das richtige Dateisystem zum Hosten eines XFS-Dateisystems fest.
Das Passwort für das Benutzerkonto **student** auf **serverb** lautet **student**. Dieser Benutzer hat über **sudo** vollen **root**-Zugriff.
 - 1.1. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 1.2. Wechseln Sie mit dem Befehl **sudo -i** zum Benutzer **root**, weil für das Erstellen von Partitionen und Dateisystemen **root**-Zugriff erforderlich ist. Verwenden Sie bei Aufforderung **student** als Passwort.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- Ermitteln Sie mit dem Befehl **lsblk** die neuen Festplatten. Diese Festplatten sollten noch keine Partitionen haben.

```
[root@serverb ~]# lsblk  
NAME  MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sr0    11:0    1 1024M  0 rom  
vda   252:0    0   10G  0 disk  
└─vda1 252:1    0   10G  0 part /  
vdb  252:16   0    5G  0 disk  
vdc   252:32   0    5G  0 disk  
vdd   252:48   0    5G  0 disk
```

Wie Sie der Ausgabe entnehmen können, hat die erste neue Festplatte **vdb** keine Partitionen.

- Vergewissern Sie sich, dass die Festplatte keine Datenträgerbezeichnung hat.

```
[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:
```

- Definieren Sie mit **parted** und dem Sub-Befehl **mklabel** das GPT-Partitionierungsschema.

```
[root@serverb ~]# parted /dev/vdb mklabel gpt  
Information: You may need to update /etc/fstab.
```

- Erstellen Sie die 2 GB große Partition. Nennen Sie sie **backup** und legen Sie ihren Typ auf **xfs** fest. Beginnen Sie die Partition bei Sektor 2048.

```
[root@serverb ~]# parted /dev/vdb mkpart backup xfs 2048s 2GB  
Information: You may need to update /etc/fstab.
```

- Überprüfen Sie die korrekte Erstellung der neuen Partition.

```
[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

- 1.8. Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet, bis das System die neue Partition erkannt und die Gerätedatei **/dev/vdb1** erstellt hat. Der Befehl kehrt erst zurück, wenn der Vorgang abgeschlossen ist.

```
[root@serverb ~]# udevadm settle
[root@serverb ~]#
```

2. Formatieren Sie die 2 GB große Partition mit einem XFS-Dateisystem und mounten Sie es dauerhaft unter **/backup**.

- 2.1. Formatieren Sie mit dem Befehl **mkfs.xfs** die Partition **/dev/vbd1**.

```
[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
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. Erstellen Sie den Mount-Punkt **/backup**.

```
[root@serverb ~]# mkdir /backup
[root@serverb ~]#
```

- 2.3. Rufen Sie vor dem Hinzufügen des neuen Dateisystems zu **/etc/fstab** die UUID ab.

```
[root@serverb ~]# lsblk --fs /dev/vdb1
NAME FSTYPE LABEL UUID                                     MOUNTPOINT
vdb1 xfs     a3665c6b-4bfb-49b6-a528-74e268b058dd
```

Die UUID weicht möglicherweise auf Ihrem System ab.

- 2.4. Bearbeiten Sie **/etc/fstab** und definieren Sie das neue Dateisystem.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4bfb-49b6-a528-74e268b058dd  /backup  xfs  defaults  0 0
```

- 2.5. Erzwingen Sie, dass **systemd** die Datei **/etc/fstab** erneut liest.

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

Kapitel 6 | Verwalten von Basisspeicher

- 2.6. Mounten Sie **/backup** manuell, um Ihre Arbeit zu verifizieren. Überprüfen Sie, ob der Mount erfolgreich war.

```
[root@serverb ~]# mount /backup
[root@serverb ~]# mount | grep /backup
/dev/vdb1 on /backup type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

- 3.** Erstellen Sie auf derselben neuen Festplatte zwei GPT-Partitionen mit jeweils 512 MB mit den Namen **swap1** und **swap2**. Eine Größe zwischen 460 MB und 564 MB ist akzeptabel. Legen Sie den richtigen Dateisystemtyp für diese Partitionen zum Hosten der Swap-Speicher fest.

- 3.1. Rufen Sie die Endposition der ersten Partition ab, indem Sie die aktuelle Partitionstabelle auf **/dev/vdb** anzeigen. Im nächsten Schritt verwenden Sie diesen Wert als Startwert für die Partition **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. Erstellen Sie die erste Partition in der Größe 512 MB und mit dem Namen **swap1**. Legen Sie deren Typ auf **linux-swap** fest. Verwenden Sie die Endposition der ersten Partition als Startpunkt. Die Endposition ist 2000 MB + 512 MB = 2512 MB

```
[root@serverb ~]# parted /dev/vdb mkpart swap1 linux-swap 2000MB 2512M
Information: You may need to update /etc/fstab.
```

- 3.3. Erstellen Sie die zweite Partition in der Größe 512 MB und mit dem Namen **swap2**. Legen Sie deren Typ auf **linux-swap** fest. Verwenden Sie die Endposition der vorherigen Partition als Startpunkt: **2512M**. Die Endposition ist 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. Zeigen Sie die Partitionstabelle an, um Ihre Arbeit zu überprüfen.

```
[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       swap1  swap
3      2512MB 3024MB 512MB       swap2  swap

```

- 3.5. Führen Sie den Befehl **udevadm settle** aus. Dieser Befehl wartet, bis das System die neuen Partitionen registriert und die Gerätedateien erstellt hat.

```
[root@serverb ~]# udevadm settle
[root@serverb ~]#
```

4. Initialisieren Sie die beiden 512 MiB großen Partitionen als Swap-Speicher und konfigurieren Sie sie so, dass sie beim Booten aktiviert werden. Legen Sie den Swap-Speicher auf der Partition **swap2** als bevorzugten Swap-Speicher fest.

- 4.1. Initialisieren Sie mit dem Befehl **mkswap** die Swap-Partitionen.

```
[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
```

Notieren Sie sich die UUIDs der beiden Swap-Speicher. Sie verwenden diese Informationen im nächsten Schritt. Wenn die **mkswap**-Ausgabe nicht mehr angezeigt wird, rufen Sie mit dem Befehl **lsblk --fs** die UUIDs ab.

- 4.2. Bearbeiten Sie **/etc/fstab** und definieren Sie die neuen Swap-Speicher. Um den Swap-Speicher auf der Partition **swap2** als vor **swap1** bevorzugten Swap-Speicher festzulegen, vergeben Sie dafür mit der Option **pri** eine höhere Priorität.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4fbf-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. Erzwingen Sie, dass **systemd** die Datei **/etc/fstab** erneut liest.

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

- 4.4. Aktivieren Sie mit dem Befehl **swapon -a** die neuen Swap-Speicher. Überprüfen Sie mit dem Befehl **swapon --show** die korrekte Aktivierung der Swap-Speicher.

```
[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. Booten Sie **serverb** neu, um Ihre Arbeit zu überprüfen. Überprüfen Sie, ob das System die erste Partition automatisch unter **/backup** mountet. Vergewissern Sie sich außerdem, dass das System die beiden Swap-Speicher aktiviert.

Melden Sie sich von **serverb** ab, wenn Sie fertig sind.

- 5.1. Booten Sie **serverb** neu.

```
[root@serverb ~]# systemctl reboot
[root@serverb ~]#
Connection to serverb closed by remote host.
Connection to serverb closed.
[student@workstation ~]$
```

- 5.2. Warten Sie ein paar Minuten, bis **serverb** neu gebootet ist, und melden Sie sich als Benutzer **student** an.

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

- 5.3. Überprüfen Sie, ob das System automatisch **/dev/vdb1** unter **/backup** mountet.

```
[student@serverb ~]$ mount | grep /backup
/dev/vdb1 on /backup type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

- 5.4. Überprüfen Sie mit dem Befehl **swapon --show**, ob das System beide Swap-Speicher aktiviert.

```
[student@serverb ~]$ swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2 partition 489M   0B   10
/dev/vdb3 partition 488M   0B   20
```

- 5.5. Melden Sie sich von **serverb** ab.

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

Bewertung

Führen Sie auf **workstation** das Skript **lab storage-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab storage-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Sie verwenden den Befehl **parted**, um Partitionen auf Festplatten mit dem MBR- oder dem GPT-Partitionierungsschema hinzuzufügen, zu ändern oder zu entfernen.
- Sie verwenden den Befehl **mkfs.xfs**, um XFS-Dateisysteme auf Festplattenpartitionen zu erstellen.
- Sie müssen **/etc/fstab** Befehle zum Mounten des Dateisystems hinzufügen, um diese Mounts dauerhaft zu machen.
- Sie verwenden den Befehl **mkswap**, um Swap-Speicher zu initialisieren.

Kapitel 7

Verwalten logischer Volumes

Ziel

Logische Volumes, die Dateisysteme und Swap-Speicher enthalten, über die Befehlszeile erstellen und verwalten

Ziele

- Logische Volumes auf Speichergeräten erstellen und verwalten, mit Dateisystemen formatieren oder mit Swap-Speicher vorbereiten
- Volume-Gruppen zugewiesenen Speicher hinzufügen und entfernen sowie zerstörungsfrei die Größe eines mit einem Dateisystem formatierten logischen Volumes erweitern

Abschnitte

- Erstellen logischer Volumes (und angeleitete Übung)
- Erweitern logischer Volumes (und angeleitete Übung)

Praktische Übung

Verwalten logischer Volumes

Erstellen logischer Volumes

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Komponenten und Konzepte des Logical Volume Management beschreiben
- LVM-Speicher implementieren
- LVM-Komponenteninformationen anzeigen

Konzepte des Logical Volume Management (LVM)

Logische Volumes und Logical Volume Management vereinfachen die Verwaltung des auf Datenträgern vorhandenen Speicherplatzes. Wenn ein Dateisystem, das ein logisches Volume hostet, mehr Platz benötigt, kann seinem logischen Volume freier Platz aus der Volume-Gruppe zugewiesen werden, wodurch eine Vergrößerung des Dateisystems möglich wird. Droht ein Datenträger auszufallen, kann ein Ersatzdatenträger als physisches Volume für die Volume-Gruppe registriert werden, sodass die Extents des logischen Volumes auf den neuen Datenträger migriert werden können.

LVM-Definitionen

Physische Geräte

Physische Geräte sind Speichergeräte, die zum Speichern von auf einem logischen Volume gespeicherten Daten verwendet werden. Es handelt sich um Blockgeräte. Sie können Datenträgerpartitionen, ganze Datenträger, RAID-Arrays oder SAN-Datenträger sein. Ein Gerät muss als physisches LVM-Volume initialisiert werden, damit es mit LVM verwendet werden kann. Das ganze Gerät wird als physisches Volume verwendet.

Physische Volumes (PVs)

Physische Volumes sind der LVM zugrunde liegende „physische“ Speicher. Sie müssen ein Gerät als physisches Volume initialisieren, bevor Sie es in einem LVM-System verwenden. LVM-Tools segmentieren physische Volumes in *physische Extents (PEs)*. Dabei handelt es sich um kleine Datenblöcke („Chunks“), die als kleinster Speicherblock auf einem physischen Volume fungieren.

Volume-Gruppen (VGs)

Volume-Gruppen sind Speicherpools, die aus mindestens einem physischen Volume bestehen. Dies ist das funktionale Äquivalent eines gesamten Datenträgers im Basisspeicher. Ein PV kann nur einer einzelnen VG zugewiesen werden. Eine VG kann aus ungenutztem Speicherplatz und einer beliebigen Anzahl logischer Volumes bestehen.

Logische Volumes (LVs)

Logische Volumes werden aus freien physischen Extents in einer Volume-Gruppe erstellt und stellen das „Speicher“-Gerät bereit, das von Anwendungen, Benutzern und dem Betriebssystem verwendet wird. LVs sind eine Sammlung *logischer Extents (LEs)*, die auf physische Extents verweisen, dem kleinsten Daten-Chunk eines PV. Standardmäßig ist jedes LE einem PE zugeordnet. Durch die Festlegung spezifischer LV-Optionen wird diese Zuordnung geändert; durch Spiegelung wird z. B. bewirkt, dass jedes LE zwei PEs zugeordnet wird.

Implementieren von LVM-Speicher

Das Erstellen eines LVM-Speichers erfordert mehrere Schritte. Im ersten Schritt müssen Sie feststellen, welche physischen Geräte verwendet werden sollen. Nachdem ein Satz geeigneter Geräte zusammengestellt wurde, werden diese als physische Volumes initialisiert, sodass sie als zu LVM gehörig erkannt werden. Die physischen Volumes werden dann zu einer Volume-Gruppe zusammengefasst. Dadurch wird ein Pool mit Datenträgerspeicher erstellt, aus dem logische Volumes zugewiesen werden können. Logische Volumes, die aus dem verfügbaren Speicherplatz in einer Volume-Gruppe erstellt werden, können mit einem Dateisystem formatiert, als Swap-Speicher aktiviert und dauerhaft gemountet oder aktiviert werden.

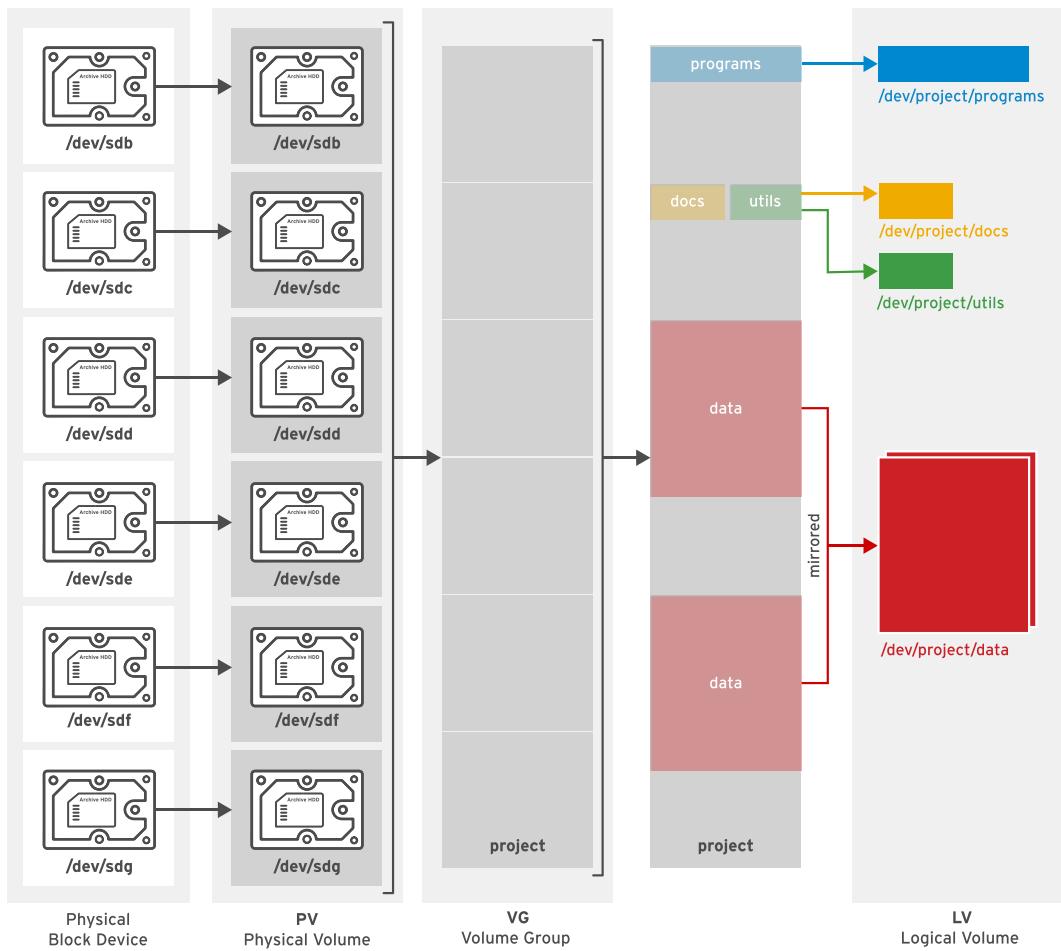


Abbildung 7: Komponenten des Logical Volume Management

LVM bietet zahlreiche Befehlszeilentools für die Implementierung und Verwaltung von LVM-Speicher. Die Tools können in Skripten verwendet werden, sodass sie sich für die Automatisierung eignen.



Wichtig

In den folgenden Beispielen werden das Gerät **vdb** und seine Partitionen verwendet, um LVM-Befehle zu illustrieren. In der Praxis müssten diese Beispiele die richtigen Geräte für den Datenträger und die Datenträgerpartitionen verwenden, die vom System verwendet werden. Mit den Befehlen **lsblk**, **blkid** oder **cat /proc/partitions** identifizieren Sie die Geräte in Ihrem System.

Erstellen eines logischen Volumes

Führen Sie die folgenden Schritte aus, um ein logisches Volume zu erstellen:

Vorbereiten des physischen Geräts

Verwenden Sie **parted**, **gdisk** oder **fdisk** zum Erstellen einer neuen Partition zur Verwendung mit LVM zu erstellen. Legen Sie den Partitionstyp bei LVM-Partitionen immer auf **Linux LVM** fest. Für MBR-Partitionen sollten Sie **0x8e** verwenden. Falls notwendig, können Sie die neue Partition mit **partprobe** beim Kernel registrieren.

Alternativ können Sie einen gesamten Datenträger, ein RAID-Array oder einen SAN-Datenträger verwenden.

Ein physisches Gerät muss nur vorbereitet werden, wenn bisher keine Geräte vorbereitet wurden und ein neues physisches Volume für die Erweiterung oder Erstellung einer Volume-Gruppe erforderlich ist.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1MiB 769MiB
[root@host ~]# parted -s /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted -s /dev/vdb set 1 lvm on
[root@host ~]# parted -s /dev/vdb set 2 lvm on
```

Erstellen eines physischen Volumes

Mit **pvccreate** benennen Sie die Partition (oder ein anderes physisches Gerät) als physisches Volume. Der Befehl **pvccreate** teilt das physische Volume in physische Extents (PEs) mit fester Größe auf, zum Beispiel in Blöcke mit 4 MiB. Sie können mehrere Geräte gleichzeitig benennen, indem Sie durch Leerzeichen getrennte Gerätenamen als Argumente für **pvccreate** angeben.

```
[root@host ~]# pvccreate /dev/vdb2 /dev/vdb1
```

Mit diesem Befehl werden die Geräte **/dev/vdb2** und **/dev/vdb1** als PVs benannt, die anschließend einer Volume-Gruppe zugewiesen werden können.

Ein PV muss nur erstellt werden, wenn keine PVs zum Erstellen oder Erweitern einer VG zur Verfügung stehen.

Erstellen einer Volume-Gruppe

Mit **vgcreate** stellen Sie mindestens ein physisches Volume zu einer Volume-Gruppe zusammen. Eine Volume-Gruppe ist das funktionale Äquivalent einer Festplatte. Sie erstellen logische Volumes aus dem Pool der freien physischen Extents in der Volume-Gruppe.

Die Befehlszeile **vgcreate** besteht aus einem Volume-Gruppennamen gefolgt von einem oder mehreren physischen Volumes, die dieser Volume-Gruppe zugeordnet werden.

```
[root@host ~]# vgcreate vg01 /dev/vdb2 /dev/vdb1
```

Durch diesen Befehl wird eine VG namens **vg01** erstellt, die die PE-Einheiten der beiden PVs **/dev/vdb2** und **/dev/vdb1** kombiniert.

Eine VG muss nur erstellt werden, wenn noch keine vorhanden ist. Zusätzliche VGs können aus administrativen Gründen zur Verwaltung der Verwendung von PVs und LVs erstellt werden.

Andernfalls können vorhandene VGs erweitert werden, sodass sie, falls erforderlich, Platz für neue LVs bieten.

Erstellen eines logischen Volumes

Mit **lvcreate** wird aus den verfügbaren physischen Extents in einer Volume-Gruppe ein neues logisches Volume erstellt. Der Befehl **lvcreate** enthält mindestens die Option **-n** zum Festlegen des LV-Namens, entweder die Option **-L** zum Festlegen der LV-Größe in Byte oder die Option **-1** zum Festlegen der LV-Größe in Extents und den Namen der Volume-Gruppe, die dieses logische Volume hostet.

```
[root@host ~]# lvcreate -n lv01 -L 700M vg01
```

Dadurch wird das LV **lv01** mit einer Größe von 700 MiB in der VG **vg01** erstellt. Dieser Befehl schlägt fehl, wenn die Volume-Gruppe nicht über genügend freie physische Extents für die angeforderte Größe verfügt. Beachten Sie auch, dass die Größe auf einen Faktor der physischen Extent-Größe gerundet wird, wenn die Größe nicht genau übereinstimmt.

Sie können die Größe mit der Option **-L** angeben, wobei die Größe in Byte, Mebibyte (binäre Megabyte, 1048576 Byte), Gibibyte (binäre Gigabyte) oder ähnliches angegeben wird. Alternativ können Sie auch die Option **-1** verwenden, die die Größe als Anzahl physischer Extents erwartet.

Die folgende Liste enthält einige Beispiele für das Erstellen von LVs:

- **lvcreate -L 128M**: Dimensionieren des logischen Volumes auf genau 128 MiB
- **lvcreate -1 128** : Dimensionieren des logischen Volumes auf genau 128 Extents Die Gesamtanzahl an Bytes hängt von der Größe der Blöcke der physischen Extents auf dem zugrunde liegenden physischen Volume ab.



Wichtig

Verschiedene Tools zeigen den Namen des logischen Volumes an und zwar entweder den herkömmlichen Namen **/dev/vgname/lvname** oder den Kernel-Gerätezuordnungsnamen **/dev/mapper/vgname-lvname**.

Hinzufügen des Dateisystems

Verwenden Sie **mkfs**, um ein **XFS**-Dateisystem auf dem neuen logischen Volume zu erstellen. Alternativ können Sie auch ein Dateisystem basierend auf Ihrem bevorzugten Dateisystem erstellen, zum Beispiel **ext4**.

```
[root@host ~]# mkfs -t xfs /dev/vg01/lv01
```

Führen Sie die folgenden Schritte aus, um das Dateisystem für sämtliche Reboots zur Verfügung zu stellen:

- Erstellen Sie mit **mkdir** einen Mount-Punkt.

```
[root@host ~]# mkdir /mnt/data
```

- Fügen Sie der Datei **/etc/fstab** einen Eintrag hinzu:

```
/dev/vg01/lv01 /mnt/data xfs defaults 1 2
```



Anmerkung

Das Mounten eines logischen Volumes nach Namen entspricht dem Mounten nach UUID, da LVM seine physischen Volumes basierend auf einer UUID ermittelt, selbst wenn Sie sie der Volume-Gruppe ursprünglich nach Namen hinzugefügt haben.

- Führen Sie **mount /mnt/data** aus, um das gerade hinzugefügte Dateisystem unter **/etc/fstab** zu mounten.

```
[root@host ~]# mount /mnt/data
```

Entfernen eines logischen Volumes

Führen Sie die folgenden Schritte aus, um *alle* logischen Volumes zu entfernen:

Vorbereiten des Dateisystems

Verschieben Sie alle Daten, die aufbewahrt werden müssen, in ein anderes Dateisystem. Unmounten Sie anschließend mit dem Befehl **umount** das Dateisystem und entfernen Sie dann alle **/etc/fstab**-Einträge, die diesem Dateisystem zugeordnet sind.

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



Warnung

Durch das Entfernen eines logischen Volumes werden sämtliche Daten, die darauf gespeichert sind, zerstört. Sichern oder Verschieben Sie die Daten, bevor Sie das logische Volume entfernen.

Entfernen des logischen Volumes

Mit **lvremove DEVICE_NAME** entfernen Sie ein logisches Volume, das Sie nicht mehr benötigen.

```
[root@host ~]# lvremove /dev/vg01/lv01
```

Bevor Sie diesen Befehl ausführen, müssen Sie das LV-Dateisystem unmounten. Der Befehl fordert Sie zur Bestätigung auf, bevor das LV entfernt wird.

Die physischen Extents des LV werden freigegeben und können somit vorhandenen oder neuen LVs in der Volume-Gruppe zugewiesen werden.

Entfernen der Volume-Gruppe

Mit **vgremove VG_NAME** entfernen Sie eine Volume-Gruppe, die Sie nicht mehr benötigen.

```
[root@host ~]# vgremove vg01
```

Die physischen Volumes der VG werden freigegeben und können somit vorhandenen oder neuen VGs im System zugewiesen werden.

Entfernen der physischen Volumes

Kapitel 7 | Verwalten logischer Volumes

Mit **pvremove** entfernen Sie physische Volumes, die Sie nicht mehr benötigen. Mit einer durch Leerstellen getrennten Liste von PV-Geräten können Sie mehrere PVs gleichzeitig entfernen. Dieser Befehl löscht die PV-Metadaten von der Partition (oder von dem Datenträger). Die Partition steht nun für die Neuzuweisung oder Neuformatierung zur Verfügung.

```
[root@host ~]# pvremove /dev/vdb2 /dev/vdb1
```

Überprüfen der LVM-Statusinformationen

Physische Volumes

Mit **pvdisplay** zeigen Sie Informationen zu physischen Volumes an. Verwenden Sie den Befehl ohne Argumente, um Informationen zu allen physischen Volumes aufzulisten. Um Informationen zu einem bestimmten physischen Volume aufzulisten, übergeben Sie den Namen dieses Geräts an den Befehl.

```
[root@host ~]# pvdisplay /dev/vdb1
--- Physical volume ---
PV Name           /dev/vdb1
VG Name           vg01
PV Size          768.00 MiB / not usable 4.00 MiB
Allocatable       yes
PE Size          4.00 MiB
Total PE         191
Free PE          16
Allocated PE     175
PV UUID          JWzDpn-LG3e-n2oi-9EtD-VT2H-PMem-1ZXwP1
```

- ① Mit **PV Name** wird eine Zuordnung zu dem Gerätenamen vorgenommen.
- ② Mit **VG Name** wird die Volume-Gruppe angezeigt, der das PV zugewiesen ist.
- ③ Mit **PV Size** wird die physische Größe des PV angezeigt, einschließlich des nicht verwendbaren Speicherplatzes.
- ④ Bei **PE Size** handelt es sich um die Größe des physischen Extents, das gleichzeitig die kleinste Größe ist, die einem logischen Volume zugewiesen werden kann.

Außerdem ist es der Multiplikationsfaktor für die Berechnung der Größe eines jeden Wertes, der in PE-Einheiten angegeben wird, wie *Free PE*. 26 PEs x 4 MiB (die *PE-Größe*) entspricht beispielsweise 104 MiB freiem Speicherplatz. Die Größe eines logischen Volumes wird auf einen Faktor von PE-Einheiten gerundet.

- LVM legt die PE-Größe automatisch fest. Es ist allerdings auch möglich, sie manuell anzugeben.
- ⑤ **Free PE** zeigt an, wie viele PE-Einheiten für die Zuweisung zu neuen logischen Volumes zur Verfügung stehen.

Volume-Gruppen

Mit **vgdisplay** zeigen Sie Informationen zu Volume-Gruppen an. Verwenden Sie den Befehl ohne Argumente, um Informationen zu allen Volume-Gruppen aufzulisten. Um Informationen zu einer bestimmten Volume-Gruppe aufzulisten, übergeben Sie den Namen der VG an den Befehl.

```
[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	1016.00 MiB ②
PE Size	4.00 MiB
Total PE	254 ③
Alloc PE / Size	175 / 700.00 MiB
Free PE / Size	79 / 316.00 MiB ④
VG UUID	3snNw3-CF71-CcYG-Llk1-p6EY-rHEv-xfUSez

- ① **VG Name** ist der Name der Volume-Gruppe.
- ② **VG Size** steht für die Gesamtgröße des Speicherpools, der für die Zuweisung zu logischen Volumes zur Verfügung steht.
- ③ **Total PE** ist die Gesamtgröße, angegeben in PE-Einheiten.
- ④ **Free PE / Size** zeigt an, wie viel Speicherplatz in der VG für die Zuweisung zu neuen LVs oder die Erweiterung vorhandener LVs zur Verfügung steht.

Logische Volumes

Mit **lvdisplay** zeigen Sie Informationen zu logischen Volumes an. Wenn Sie kein Argument für den Befehl angeben, werden Informationen zu allen LVs angezeigt. Wenn Sie einen LV-Gerätenamen als Argument angeben, zeigt der Befehl Informationen zu diesem speziellen Gerät an.

[root@host ~]# lvdisplay /dev/vg01/lv01	
--- Logical volume ---	
LV Path	/dev/vg01/lv01 ①
LV Name	lv01
VG Name	vg01 ②
LV UUID	5IyRea-W8Zw-xLhk-3h2a-IuVN-YaeZ-i3IRrN
LV Write Access	read/write
LV Creation host, time	host.lab.example.com, 2019-03-28 17:17:47 -0400
LV Status	available
# open	1
LV Size	700 MiB ③
Current LE	175 ④
Segments	1
Allocation	inherit
Read ahead sectors	auto
- current set to	256
Block device	252:0

- ① **LV Path** zeigt den Gerätenamen des logischen Volumes an.

Einige Tools geben den Gerätenamen unter Umständen als **/dev/mapper/vgname-lvname** an; beide stehen für das gleiche LV.

- ② **VG Name** zeigt die Volume-Gruppe an, aus der das LV zugewiesen wurde.
- ③ **LV Size** zeigt die Gesamtgröße des LV an. Überprüfen Sie mit den Dateisystemtools, welcher Speicherplatz belegt ist und welcher für die Datenspeicherung zur Verfügung steht.
- ④ **Current LE** zeigt die Anzahl der logischen Extents an, die von diesem LV genutzt werden. Ein LE lässt sich in der Regel einem physischen Extent in der VG und somit auch dem physischen Volume zuordnen.



Literaturhinweise

Manpages **lvm(8)**, **pvcreate(8)**, **vgcreate(8)**, **lvcreate(8)**, **pvremove(8)**, **vgremove(8)**, **lvremove(8)**, **pvdisplay(8)**, **vgdisplay(8)**, **lvdisk(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)** und **mkfs(8)**

► Angeleitete Übung

Erstellen logischer Volumes

In dieser praktischen Übung erstellen Sie ein physisches Volume, eine Volume-Gruppe, ein logisches Volume und ein XFS-Dateisystem. Außerdem mounten Sie das Dateisystem des logischen Volumes dauerhaft.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Physische Volumes, Volume-Gruppen und logische Volumes mit LVM-Tools erstellen
- Neue Dateisysteme auf logischen Volumes erstellen und dauerhaft mounten

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab lvm-creating start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem wird überprüft, ob Speicher verfügbar ist und die entsprechenden Softwarepakete installiert sind.

```
[student@workstation ~]$ lab lvm-creating start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Erstellen Sie die physischen Ressourcen auf dem Gerät **/dev/vdb**.

- 3.1. Erstellen Sie mit **parted** zwei 256 MiB große Partitionen und legen Sie sie auf den Typ „Linux LVM“ fest.

```
[root@servera ~]# parted -s /dev/vdb mklabel gpt
[root@servera ~]# parted -s /dev/vdb mkpart primary 1MiB 257MiB
[root@servera ~]# parted -s /dev/vdb set 1 lvm on
[root@servera ~]# parted -s /dev/vdb mkpart primary 258MiB 514MiB
[root@servera ~]# parted -s /dev/vdb set 2 lvm on
```

- 3.2. Verwenden Sie **udevadm settle**, damit das System die neuen Partitionen registriert.

```
[root@servera ~]# udevadm settle
```

- 4. Fügen Sie mit **pvcreate** die beiden neuen Partitionen als PVs hinzu.

```
[root@servera ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
```

- 5. Erstellen Sie mit **vgcreate** eine neue VG mit dem Namen **servera_01_vg** aus den beiden PVs.

```
[root@servera ~]# vgcreate servera_01_vg /dev/vdb1 /dev/vdb2
Volume group "servera_01_vg" successfully created
```

- 6. Erstellen Sie mit **lvcreate** ein 400 MiB großes LV mit dem Namen **servera_01_lv** aus der VG **servera_01_vg**.

```
[root@servera ~]# lvcreate -n servera_01_lv -L 400M servera_01_vg
Logical volume "servera_01_lv" created.
```

Dadurch wird ein Gerät mit dem Namen **/dev/servera_01_vg/servera_01_lv**, aber ohne Dateisystem erstellt.

- 7. Fügen Sie ein dauerhaftes Dateisystem hinzu.

- 7.1. Fügen Sie auf dem LV **servera_01_lv** mit dem Befehl **mkfs** ein **XFS**-Dateisystem hinzu.

```
[root@servera ~]# mkfs -t xfs /dev/servera_01_vg/servera_01_lv
...output omitted...
```

- 7.2. Erstellen Sie einen Mount-Punkt bei **/data**.

```
[root@servera ~]# mkdir /data
```

- 7.3. Fügen Sie auf **servera** die folgende Zeile am Ende von **/etc/fstab** hinzu:

```
/dev/servera_01_vg/servera_01_lv    /data    xfs    defaults    1 2
```

- 7.4. Aktualisieren Sie mit **systemctl daemon-reload systemd** mit der neuen **/etc/fstab**-Konfiguration.

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

- 7.5. Überprüfen Sie den **/etc/fstab**-Eintrag und mounten Sie mit dem Befehl **mount** das neue LV-Gerät **servera_01_lv**.

```
[root@servera ~]# mount /data
```

► 8. Testen und überprüfen Sie Ihre Arbeit.

- 8.1. Kopieren Sie als letzten Test einige Dateien auf **/data** und prüfen Sie, wie viele Dateien kopiert wurden.

```
[root@servera ~]# cp -a /etc/*.* /data
```

```
[root@servera ~]# ls /data | wc -l
```

34

In der nächsten angeleiteten Übung überprüfen Sie, ob immer noch dieselbe Anzahl Dateien vorhanden ist.

- 8.2. **parted /dev/vdb print** listet die Partitionen auf, die auf **/dev/vdb** vorhanden sind.

```
[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  269MB  268MB          primary  lvm
 2      271MB   539MB  268MB          primary  lvm
```

Beachten Sie die Spalte **Number**, die die Werte **1** und **2** enthält. Diese entsprechen **/dev/vdb1** beziehungsweise **/dev/vdb2**. Beachten Sie auch die Spalte **Flags**, die den Partitionstyp angibt.

- 8.3. **pvdisplay** zeigt Informationen zu jedem physicalen Volume an. Fügen Sie optional den Gerätenamen hinzu, um die Details auf ein bestimmtes PV zu beschränken.

```
[root@servera ~]# pvdisplay /dev/vdb2
--- Physical volume ---
PV Name           /dev/vdb2
VG Name           servera_01_vg
PV Size          256.00 MiB / not usable 4.00 MiB
Allocatable       yes
PE Size          4.00 MiB
Total PE         63
```

Free PE	26
Allocated PE	37
PV UUID	2z0Cf3-99YI-w9ny-a1EW-wwhL-S8RJ-M2rfZk

Dies zeigt, dass das PV der VG **servera_01_vg** zugewiesen ist, 256 MiB groß ist (auch wenn 4 MiB nicht nutzbar sind) und die physische Extent-Größe (**PE Size**) 4 MiB (kleinste zuweisbare LV-Größe) beträgt.

Es gibt 63 PEs, von denen 26 frei sind und zukünftig LVs zugewiesen werden können und 37 momentan LVs zugewiesen sind. Sie lassen sich wie folgt in MiB-Werte übersetzen:

- Gesamtgröße: 252 MiB (63 PEs x 4 MiB); denken Sie daran, dass 4 MiB nicht nutzbar sind.
- Frei: 104 MiB (26 PEs x 4 MiB)
- Zugewiesen: 148 MiB (37 PEs x 4 MiB)

- 8.4. **vgdisplay vgname** zeigt Informationen über die Volume-Gruppe namens **vgname** an.

```
[root@servera ~]# vgdisplay servera_01_vg
```

Überprüfen Sie die folgenden Werte:

- **VG Size** beträgt **504,00 MiB**.
- **Total PE** beträgt **126**.
- **Allocated PE / Size** beträgt **100/400,00 MiB**.
- **Free PE / Size** beträgt **26/104,00 MiB**.

- 8.5. **lvdisplay /dev/vgname/lvname** zeigt Informationen über das logische Volume namens **lvname** an.

```
[root@servera ~]# lvdisplay /dev/servera_01_vg/servera_01_lv
```

Überprüfen Sie **LV Path**, **LV Name**, **VG Name**, **LV Status**, **LV Size** und **Current LE** (logische Extents, die physischen Extents zugeordnet sind).

- 8.6. Der Befehl **mount** zeigt alle gemounteten Geräte und alle Mount-Optionen an. Er sollte **/dev/servera_01_vg/servera_01_lv** enthalten.



Anmerkung

Viele Tools zeigen stattdessen den Namen der Gerätezuweisung **/dev/mapper/servera_01_vg-servera_01_lv** an. Dabei handelt es sich um dasselbe logische Volume.

```
[root@servera ~]# mount
```

Sie müssten (vermutlich in der letzten Zeile) **/dev/mapper/servera_01_vg-servera_01_lv** gemountet auf **/data** und die zugehörigen Mount-Informationen sehen.

- 8.7. **df -h** zeigt den freien Speicher auf dem Datenträger in lesbarer Form an. Fügen Sie optional einen Mount-Punkt hinzu, um die Details auf dieses Dateisystem zu begrenzen.

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  395M   24M  372M   6% /data
```

Wenn Dateisystem-Metadaten zugelassen sind, wären diese Werte die erwarteten.

- 9. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** das Skript **lab lvm-creating finish** aus, um diese Übung abzuschließen. Dieses Skript entfernt den während der Übung auf **servera** konfigurierten Speicher.

```
[student@workstation ~]$ lab lvm-creating finish
```

Hiermit ist die angeleitete Übung beendet.

Erweitern logischer Volumes

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Eine Volume-Gruppe (VG) mit **pvccreate** und **vgextend** erweitern und die Ergebnisse mit **vgdisplay** überprüfen
- Eine VG mit **pvmove** und **vgreduce** verkleinern
- Ein logisches Volume (LV) mit **lvextend** erweitern
- Die Größe von **XFS**-Dateisystemen mit **xfs_growfs** ändern
- Die Größe von **ext4**-Dateisystemen mit **resize2fs** ändern

Erweitern und Verkleinern einer Volume-Gruppe

Sie können den Datenträgerspeicher für eine Volume-Gruppe vergrößern, indem Sie weitere physische Volumes hinzufügen. Dies wird als *Erweitern der Volume-Gruppe* bezeichnet. Anschließend können Sie die neuen physischen Extents von den zusätzlichen physischen Volumes logischen Volumes zuweisen.

Sie können nicht verwendete physische Volumes aus einer Volume-Gruppe entfernen. Dies wird als *Verkleinern der Volume-Gruppe* bezeichnet. Verschieben Sie zuerst mit dem Befehl **pvmove** Daten von Extents auf einem physischen Volume auf Extents anderer physischer Volumes in der Volume-Gruppe. So kann ein neuer Datenträger zu einer vorhandenen Volume-Gruppe hinzugefügt, Daten können von einem älteren und langsameren Datenträger auf einen neuen Datenträger verschoben und der alte Datenträger kann aus der Volume-Gruppe entfernt werden. Sie können diese Vorgänge durchführen, während die logischen Volumes in der Volume-Gruppe verwendet werden.



Wichtig

In den folgenden Beispielen werden das Gerät **vdb** und seine Partitionen verwendet, um LVM-Befehle zu illustrieren. Verwenden Sie in der Praxis die entsprechenden Geräte für die Datenträger- und Datenträgerpartitionen auf Ihrem System.

Erweitern einer Volume-Gruppe

Führen Sie die folgenden Schritte aus, um eine Volume-Gruppe zu erweitern:

Vorbereiten des physischen Geräts und Erstellen des physischen Volumes

Wie beim Erstellen einer neuen Volume-Gruppe müssen Sie eine neue Partition für die Verwendung als physisches Volume erstellen und vorbereiten, wenn noch keine vorbereiteten vorhanden sind.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1027MiB 1539MiB
[root@host ~]# parted -s /dev/vdb set 3 lvm on
[root@host ~]# pvcreate /dev/vdb3
```

Ein PV muss nur erstellt werden, wenn keine PVs für die Erweiterung der VG frei sind.

Erweitern der Volume-Gruppe

Mit **vgextend** fügen Sie der Volume-Gruppe ein neues physisches Volume hinzu. Geben Sie den VG-Namen und den PV-Gerätenamen als Argumente für **vgextend** an.

```
[root@host ~]# vgextend vg01 /dev/vdb3
```

Dadurch wird die VG **vg01** um die Größe des PV **/dev/vdb3** erweitert.

Überprüfen, ob der neue Speicherplatz verfügbar ist

Überprüfen Sie mit **vgdisplay**, ob die weiteren physischen Extents verfügbar sind. Prüfen Sie **Free PE / Size** in der Ausgabe. Der Wert sollte ungleich null sein.

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name           vg01
...output omitted...
Free PE / Size    178 / 712.00 MiB
...output omitted...
```

Verkleinern einer Volume-Gruppe

Führen Sie die folgenden Schritte aus, um eine Volume-Gruppe zu verkleinern:

Verschieben der physischen Extents

Verschieben Sie mit **pvmove PV_DEVICE_NAME** alle physische Extents von dem physischen Volume, das Sie entfernen möchten, auf andere physische Volumes in der Volume-Gruppe. Die anderen physischen Volumes müssen über eine ausreichende Anzahl freier Extents verfügen, um die verschobenen Extents aufnehmen zu können. Dies ist nur möglich, wenn genügend freie Extents in der VG verfügbar sind und alle von anderen PVs stammen.

```
[root@host ~]# pvmove /dev/vdb3
```

Dieser Befehl verschiebt die PEs von **/dev/vdb3** zu anderen PVs mit freien PEs in derselben VG.



Warnung

Sichern Sie vor der Verwendung von **pvmove** alle auf allen logischen Volumes gespeicherten Daten in der Volume-Gruppe. Durch einen Stromausfall während des Vorgangs könnte die Volume-Gruppe in einem inkonsistenten Zustand verbleiben. Das kann zu Datenverlust auf logischen Volumes in der Volume-Gruppe führen.

Verkleinern der Volume-Gruppe

Mit **vgreduce VG_NAME PV_DEVICE_NAME** entfernen Sie ein physisches Volume aus einer Volume-Gruppe.

```
[root@host ~]# vgreduce vg01 /dev/vdb3
```

Damit wird das PV **/dev/vdb3** aus der VG **vg01** entfernt und es kann jetzt einer anderen VG hinzugefügt werden. Alternativ kann mit **pvremove** die Verwendung des Geräts als PV dauerhaft beendet werden.

Erweitern eines logischen Volumes und des XFS-Dateisystems

Ein Vorteil von logischen Volumes ist die Möglichkeit, sie ohne Ausfallzeiten zu erweitern. In einer Volume-Gruppe können freie physische Extents einem logischen Volume hinzugefügt werden, um dessen Kapazität zu erweitern. Auf diese Weise lässt sich auch das darin enthaltene Dateisystem erweitern.

Erweitern eines logischen Volumes

Führen Sie die folgenden Schritte aus, um ein logisches Volume zu erweitern:

Überprüfen, ob in der Volume-Gruppe genügend Speicherplatz verfügbar ist

Überprüfen Sie mit **vgdisplay**, ob genügend physische Extents verfügbar sind.

```
[root@host ~]# vgdisplay vg01
--- Volume group ---
VG Name           vg01
...output omitted...
Free PE / Size    178 / 712.00 MiB
...output omitted...
```

Prüfen Sie **Free PE / Size** in der Ausgabe. Überprüfen Sie, ob die Volume-Gruppe über ausreichend freien Speicherplatz für die LV-Erweiterung verfügt. Wenn nicht genügend Speicherplatz verfügbar ist, erweitern Sie die Volume-Gruppe entsprechend. Siehe „*Erweitern und Verkleinern einer Volume-Gruppe*“.

Erweitern des logischen Volumes

Mit **lvextend LV_DEVICE_NAME** erweitern Sie das logische Volume auf eine neue Größe.

```
[root@host ~]# lvextend -L +300M /dev/vg01/lv01
```

Dadurch wird die Größe des logischen Volumes **lv01** um 300 MiB erhöht. Beachten Sie das Pluszeichen (+) vor der Größe. Das bedeutet, dass dieser Wert der vorhandenen Größe hinzugefügt wird, andernfalls gibt der Wert die endgültige Größe des LV an.

Wie bei **lvcreate** gibt es verschiedene Methoden, um die Größe anzugeben: Die Option **-l** erwartet die Anzahl der physischen Extents als Argument. Die Option **-L** erwartet Größen in Byte, Mebibyte, Gibibyte usw.

Die folgende Liste enthält einige Beispiele für das Erweitern von LVs:

Beispiele für das Erweitern von LVs

Befehl	Ergebnisse
lvextend -l 128	Ändern der Größe des logischen Volumes auf eine Größe von genau 128 Extents
lvextend -l +128	Hinzufügen von 128 Extents zu der aktuellen Größe des logischen Volumes
lvextend -L 128M	Ändern der Größe des logischen Volumes auf genau 128 MiB
lvextend -L +128M	Hinzufügen von 128 MiB zu der aktuellen Größe des logischen Volumes
lvextend -l +50%FREE	Hinzufügen von 50 % des aktuellen freien Speicherplatzes in der VG zu dem LV.

Erweitern des Dateisystems

Mit **xfs_growfs mountpoint** erweitern Sie das Dateisystem, um das erweiterte LV zu belegen. Das Zieldateisystem muss gemountet sein, wenn Sie den Befehl **xfs_growfs** verwenden. Sie können das Dateisystem während der Größenänderung weiterhin verwenden.

```
[root@host ~]# xfs_growfs /mnt/data
```



Anmerkung

Ein häufiger Fehler besteht darin, **lvextend** auszuführen, nicht aber **xfs_growfs**. Eine Alternative zur aufeinanderfolgenden Ausführung der zwei Schritte besteht darin, die Option **-r** in den Befehl **lvextend** einzuschließen. Dadurch wird die Größe des Dateisystems angepasst, nachdem das LV erweitert wurde. Dafür wird **fsadm(8)** verwendet. Der Befehl funktioniert mit einer Reihe unterschiedlicher Dateisysteme.

Überprüfen Sie die neue Größe des gemounteten Dateisystems.

```
[root@host ~]# df -h /mountpoint
```

Erweitern eines logischen Volumes und des ext4-Dateisystems

Die Schritte zum Erweitern eines **ext4**-basierten logischen Volumes sind im Wesentlichen die gleichen wie für ein **XFS**-basiertes LV, außer dem Schritt zur Größenänderung des Dateisystems. Sehen Sie sich „Erweitern eines logischen Volumes und des XFS-Dateisystems“ nochmals an.

Überprüfen, ob in der Volume-Gruppe genügend Speicherplatz verfügbar ist

Überprüfen Sie mit **vgdisplay VGNAME**, ob für die Volume-Gruppe genügend physische Extents verfügbar sind.

Erweitern des logischen Volumes

Erweitern Sie mit **lvextend -l +extents /dev/vgname/lvname** das logische Volume */dev/vgname/lvname* um den Wert *extents*.

Erweitern des Dateisystems

Mit **resize2fs /dev/vgname/lvname** erweitern Sie das Dateisystem, um das neue erweiterte LV zu belegen. Das Dateisystem kann gemountet sein und verwendet werden, während der Erweiterungsbefehl ausgeführt wird. Sie können die Option **-p** angeben, um den Fortschritt des Größenänderungsvorgangs anzuzeigen.

```
[root@host ~]# resize2fs /dev/vg01/lv01
```



Anmerkung

Der Hauptunterschied zwischen **xfs_growfs** und **resize2fs** ist das Argument, das zum Identifizieren des Dateisystems übergeben wird. **xfs_growfs** übernimmt den Mount-Punkt und **resize2fs** den Namen des logischen Volumes.

Erweitern eines logischen Volumes und des Swap-Speichers

Logische Datenträger, die als Swap-Speicher formatiert sind, können ebenfalls erweitert werden. Der Vorgang unterscheidet sich jedoch von dem für die Erweiterung eines Dateisystems, wie **ext4** oder **XFS**. Mit einem Dateisystem formatierte logische Volumes können ohne Ausfallzeiten dynamisch erweitert werden. Logische Volumes, die mit Swap-Speicher formatiert sind, müssen offline sein, um sie zu erweitern.

Überprüfen, ob in der Volume-Gruppe genügend Speicherplatz verfügbar ist

Überprüfen Sie mit **vgdisplay vgname**, ob genügend freie physische Extents verfügbar sind.

Deaktivieren des Swap-Speichers

Mit **swapoff -v /dev/vgname/lvname** deaktivieren Sie den Swap-Speicher auf dem logischen Volume.



Warnung

Ihr System muss über genügend freien Arbeitsspeicher oder Swap-Speicher verfügen, um alles aufnehmen zu können, wenn der Swap-Speicher auf dem logischen Datenträger deaktiviert ist.

Erweitern des logischen Volumes

lvextend -l +extents /dev/vgname/lvname erweitert das logische Volume */dev/vgname/lvname* um den Wert *extents*.

Formatieren des logischen Volumes als Swap-Speicher

mkswap /dev/vgname/lvname formatiert das gesamte logische Volume als Swap-Speicher.

Aktivieren des Swap-Speichers

Mit **swapon -va /dev/vgname/lvname** aktivieren Sie den Swap-Speicher auf dem logischen Volume.



Literaturhinweise

Manpages **lvm(8)**, **pvccreate(8)**, **pvmove(8)**, **vgdisplay(8)**, **vgextend(8)**, **vgreduce(8)**, **vgdisplay(8)**, **vgextend(8)**, **vgreduce(8)**, **lvextend(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)**, **xfs_growfs(8)** und **resize2fs(8)** **swapoff(8)** **swapon(8)** **mkswap(8)**

► Angeleitete Übung

Erweitern logischer Volumes

In dieser praktischen Übung erweitern Sie das logische Volume, das in der vorherigen praktischen Übung hinzugefügt wurde.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Die Volume-Gruppe um ein zusätzliches physisches Volume erweitern
- Die Größe des logischen Volumes ändern, während das Dateisystem weiterhin gemountet ist und verwendet wird

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab lvm-extending start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Host **servera** im Netzwerk erreichbar ist und stellt sicher, dass der Speicher aus der vorherigen angeleiteten Übung verfügbar ist.

```
[student@workstation ~]$ lab lvm-extending start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an.

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

- 2. Wechseln Sie mit dem Befehl **sudo -i** an der Eingabeaufforderung der Shell zu **root**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Prüfen Sie mit **vgdisplay**, ob die VG über ausreichend freien Speicherplatz verfügt, um das LV auf eine Gesamtgröße von 700 MiB zu erweitern.

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name          servera_01_vg
System ID
Format           lvm2
...output omitted...
VG Size          504.00 MiB
PE Size          4.00 MiB
```

Kapitel 7 | Verwalten logischer Volumes

```
Total PE           126
Alloc PE / Size   100 / 400.00 MiB
Free  PE / Size   26 / 104.00 MiB
VG UUID          0BBATU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

Es sind nur 104 MiB verfügbar (26 PEs x 4 MiB Extents) und Sie benötigen mindestens 300 MiB, um insgesamt über 700 MiB zu verfügen. Sie müssen die VG erweitern.

Erfassen Sie zum späteren Vergleich mit **df** den aktuellen freien Speicherplatz:

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  395M   24M  372M   6% /data
```

▶ **4.** Erstellen Sie die physische Ressource.

- 4.1. Erstellen Sie mit **parted** eine weitere Partition mit 512 MiB und legen Sie sie auf den Typ „Linux LVM“ fest.

```
[root@servera ~]# parted -s /dev/vdb mkpart primary 515MiB 1027MiB
[root@servera ~]# parted -s /dev/vdb set 3 lvm on
```

- 4.2. Verwenden Sie **udevadm settle**, damit das System die neue Partition registriert.

```
[root@servera ~]# udevadm settle
```

▶ **5.** Fügen Sie mit **pvccreate** die neue Partition als PV hinzu.

```
[root@servera ~]# pvccreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

▶ **6.** Erweitern Sie die Volume-Gruppe.

- 6.1. Erweitern Sie mit **vgextend** die VG **servera_01_vg** und verwenden Sie dazu das neue PV **/dev/vdb3**.

```
[root@servera ~]# vgextend servera_01_vg /dev/vdb3
Volume group "servera_01_vg" successfully extended
```

- 6.2. Untersuchen Sie mit **vgdisplay** erneut den freien Speicherplatz der VG **servera_01_vg**. Es sollte jetzt ausreichend freier Speicherplatz vorhanden sein.

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name           servera_01_vg
System ID         lvm2
Format            lvm2
...output omitted...
VG Size           1012.00 MiB
PE Size           4.00 MiB
Total PE          253
```

```
Alloc PE / Size      100 / 400.00 MiB
Free  PE / Size     153 / 612.00 MiB
VG UUID             0BBATU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

Jetzt sind 612 MiB freier Speicherplatz verfügbar (153 PEs x 4 MiB Extents).

- 7. Erweitern Sie mit **lvextend** das vorhandene LV auf 700 MiB.

```
[root@servera ~]# lvextend -L 700M /dev/servera_01_vg/servera_01_lv
Size of logical volume servera_01_vg/servera_01_lv changed from 400.00 MiB (100
extents) to 700.00 MiB (175 extents).
Logical volume servera_01_vg/servera_01_lv successfully resized.
```



Anmerkung

In dem Beispiel ist die genaue Größe für die endgültige LV angegeben. Sie haben jedoch möglicherweise den gewünschten zusätzlichen Speicherplatz angegeben:

- **-L +300M**, um den neuen Speicherplatz unter Angabe der Größe in MiB hinzuzufügen.
- **-l 175**, um die Gesamtzahl der Extents anzugeben (175 PEs x 4 MiB).
- **-l +75**, um die benötigten zusätzlichen Extents hinzuzufügen.

- 8. Erweitern Sie das XFS-Dateisystem mit **xfs_growfs** auf den restlichen freien Speicherplatz auf dem LV.

```
[root@servera ~]# xfs_growfs /data
meta-data=/dev/mapper/servera_01_vg-servera_01_lv isize=512    agcount=4,
agsize=25600 blks
...output omitted...
```

- 9. Überprüfen Sie mit **df** und **ls | wc** die Größe des neuen Dateisystems und verifizieren Sie, ob die zuvor bestehenden Dateien noch vorhanden sind.

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  695M   26M  670M   4% /data
[root@servera ~]# ls /data | wc -l
34
```

Die Dateien sind noch vorhanden und das Dateisystem hat ungefähr die angegebene Größe.

- 10. Melden Sie sich von **servera** ab.

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

Beenden

Führen Sie auf **workstation** den Befehl **lab lvm-extending finish** aus, um diese Übung abzuschließen. Dieses Skript entfernt den während der Übung auf **servera** konfigurierten Speicher.

```
[student@workstation ~]$ lab lvm-extending finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Verwalten logischer Volumes

Leistungscheckliste

In dieser praktischen Übung ändern Sie die Größe eines vorhandenen logischen Volumes, fügen nach Bedarf LVM-Ressourcen hinzu und fügen dann ein neues logisches Volume mit einem dauerhaft gemounteten XFS-Dateisystem hinzu.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Größe des logischen Volumes **serverb_01_lv** in 768 MiB ändern
- Ein neues 128 MiB großes Volume mit dem Namen **serverb_02_lv** mit einem unter **/storage/data2** dauerhaft gemounteten XFS-Dateisystem erstellen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab lvm-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Der Befehl bereitet auch den Speicher auf **serverb** für die Übung vor.

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

Auf **serverb** verfügt das unter **/storage/data1** gemountete logische Volume **serverb_01_lv** nicht über genügend Speicherplatz und Sie wurden gebeten, den Speicherplatz auf 768 MiB zu erweitern. Sie müssen sicherstellen, dass **serverb_01_lv** dauerhaft unter **/storage/data1** gemountet bleibt.

Sie wurden zudem gebeten, ein neues, 128 MiB großes logisches Volume mit dem Namen **serverb_02_lv** zu erstellen, dass unter **/storage/data2** gemountet ist. Sie wurden angewiesen, das neue logische Volume mit dem XFS-Dateisystem zu formatieren.

Die logischen Volumes sind in der Volume-Gruppe **serverb_01_vg** enthalten. Leider ist nicht genügend Speicherplatz vorhanden, um das vorhandene logische Volume zu erweitern und das neue hinzuzufügen. Auf **/dev/vdb** wurde zuvor eine Partition mit 512 MiB erstellt. Sie wurden angewiesen, weitere 512 MiB auf **/dev/vdb** zu verwenden. Sie müssen die neue Partition erstellen.

1. Erstellen Sie eine 512 MiB große Partition auf **/dev/vdb**, initialisieren Sie sie als physisches Volume und erweitern Sie die Volume-Gruppe **serverb_01_vg** mit der Partition.
2. Erweitern Sie das logische Volume **serverb_01_lv** auf 768 MiB, einschließlich des Dateisystems.
3. Erstellen Sie in der vorhandenen Volume-Gruppe das neue logische Volume **serverb_02_lv** mit einer Größe von 128 MiB. Fügen Sie ein XFS-Dateisystem hinzu und mounten Sie es persistent unter **/storage/data2**.

Kapitel 7 | Verwalten logischer Volumes

4. Wenn Sie fertig sind, booten Sie den Rechner **serverb** neu. Führen Sie dann den Befehl **lab lvm-review grade** von Ihrem Rechner **workstation** aus, um Ihre Arbeit zu überprüfen.
Warten Sie, bis **serverb** vollständig gebootet ist, und fahren Sie dann mit der Bewertung fort.

Bewertung

Führen Sie auf **workstation** das Skript **lab lvm-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab lvm-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

► Lösung

Verwalten logischer Volumes

Leistungscheckliste

In dieser praktischen Übung ändern Sie die Größe eines vorhandenen logischen Volumes, fügen nach Bedarf LVM-Ressourcen hinzu und fügen dann ein neues logisches Volume mit einem dauerhaft gemounteten XFS-Dateisystem hinzu.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Größe des logischen Volumes **serverb_01_lv** in 768 MiB ändern
- Ein neues 128 MiB großes Volume mit dem Namen **serverb_02_lv** mit einem unter **/storage/data2** dauerhaft gemounteten XFS-Dateisystem erstellen

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab lvm-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Der Befehl bereitet auch den Speicher auf **serverb** für die Übung vor.

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

Auf **serverb** verfügt das unter **/storage/data1** gemountete logische Volume **serverb_01_lv** nicht über genügend Speicherplatz und Sie wurden gebeten, den Speicherplatz auf 768 MiB zu erweitern. Sie müssen sicherstellen, dass **serverb_01_lv** dauerhaft unter **/storage/data1** gemountet bleibt.

Sie wurden zudem gebeten, ein neues, 128 MiB großes logisches Volume mit dem Namen **serverb_02_lv** zu erstellen, dass unter **/storage/data2** gemountet ist. Sie wurden angewiesen, das neue logische Volume mit dem XFS-Dateisystem zu formatieren.

Die logischen Volumes sind in der Volume-Gruppe **serverb_01_vg** enthalten. Leider ist nicht genügend Speicherplatz vorhanden, um das vorhandene logische Volume zu erweitern und das neue hinzuzufügen. Auf **/dev/vdb** wurde zuvor eine Partition mit 512 MiB erstellt. Sie wurden angewiesen, weitere 512 MiB auf **/dev/vdb** zu verwenden. Sie müssen die neue Partition erstellen.

1. Erstellen Sie eine 512 MiB große Partition auf **/dev/vdb**, initialisieren Sie sie als physisches Volume und erweitern Sie die Volume-Gruppe **serverb_01_vg** mit der Partition.
 - 1.1. Melden Sie sich bei **serverb** als Benutzer **student** an.

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

Kapitel 7 | Verwalten logischer Volumes

- 1.2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 1.3. Erstellen Sie die 512 MiB große Partition mit **parted** und legen Sie sie auf den Typ „Linux LVM“ fest.

```
[root@serverb ~]# parted -s /dev/vdb mkpart primary 514MiB 1026MiB
[root@serverb ~]# parted -s /dev/vdb set 2 lvm on
```

- 1.4. Verwenden Sie **udevadm settle**, damit das System die neue Partition registriert.

```
[root@servera ~]# udevadm settle
```

- 1.5. Initialisieren Sie die neue Partition mit **pvcreate** als PV.

```
[root@serverb ~]# pvcreate /dev/vdb2
Physical volume "/dev/vdb2" successfully created.
```

- 1.6. Erweitern Sie mit **vgextend** die VG **serverb_01_vg** und verwenden Sie dazu das neue PV **/dev/vdb2**.

```
[root@serverb ~]# vgextend serverb_01_vg /dev/vdb2
Volume group "serverb_01_vg" successfully extended
```

2. Erweitern Sie das logische Volume **serverb_01_lv** auf 768 MiB, einschließlich des Dateisystems.

- 2.1. Erweitern Sie mit **lvextend** das LV **serverb_01_lv** auf 768 MiB.

```
[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.
```

**Anmerkung**

Alternativ können Sie für die Größenänderung des LV auch **-L +512M** verwenden.

- 2.2. Erweitern Sie das XFS-Dateisystem mit **xfs_growfs** auf den restlichen freien Speicherplatz auf dem 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...
```



Anmerkung

In diesem Beispiel wird der **xfs_growfs**-Schritt zur Erweiterung des Dateisystems verwendet. Eine Alternative wäre, dem Befehl **lvextend** die Option **-r** hinzuzufügen.

3. Erstellen Sie in der vorhandenen Volume-Gruppe das neue logische Volume **serverb_02_lv** mit einer Größe von 128 MiB. Fügen Sie ein XFS-Dateisystem hinzu und mounten Sie es persistent unter **/storage/data2**.

- 3.1. Erstellen Sie mit **lvcreate** eine 128 MiB große LV mit dem Namen **serverb_02_lv** aus der VG **serverb_01_vg**.

```
[root@serverb ~]# lvcreate -n serverb_02_lv -L 128M serverb_01_vg  
Logical volume "serverb_02_lv" created
```

- 3.2. Fügen Sie mit **mkfs** auf dem LV **serverb_02_lv** ein **xfs**-Dateisystem hinzu. Verwenden Sie den LV-Gerätenamen.

```
[root@serverb ~]# mkfs -t xfs /dev/serverb_01_vg/serverb_02_lv  
meta-data=/dev/serverb_01_vg/serverb_02_lv isize=512    agcount=4, agsize=8192  
blks  
...output omitted...
```

- 3.3. Erstellen Sie mit **mkdir** einen Mount-Punkt bei **/storage/data2**.

```
[root@serverb ~]# mkdir /storage/data2
```

- 3.4. Fügen Sie auf **serverb** die folgende Zeile am Ende von **/etc/fstab** hinzu:

```
/dev/serverb_01_vg/serverb_02_lv /storage/data2 xfs defaults 1 2
```

- 3.5. Aktualisieren Sie mit **systemctl daemon-reload** **systemd** mit der neuen **/etc/fstab**-Konfiguration.

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

- 3.6. Überprüfen Sie mit **mount** den Eintrag **/etc/fstab** und mounten Sie das neue LV-Gerät **serverb_02_lv**.

```
[root@serverb ~]# mount /storage/data2
```

4. Wenn Sie fertig sind, booten Sie den Rechner **serverb** neu. Führen Sie dann den Befehl **lvm-review grade** von Ihrem Rechner **workstation** aus, um Ihre Arbeit zu überprüfen.

```
[root@serverb ~]# systemctl reboot
```

Warten Sie, bis **serverb** vollständig gebootet ist, und fahren Sie dann mit der Bewertung fort.

Bewertung

Führen Sie auf **workstation** das Skript **lab lvm-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab lvm-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Mit LVM können Sie flexiblen Speicher erstellen, indem Sie Speicherplatz auf mehreren Speichergeräten zuweisen.
- Physische Volumes, Volume-Gruppen und logische Volumes werden von verschiedenen Tools, wie z. B. **pvccreate**, **vgreduce** und **lvextend**, verwaltet.
- Logische Volumes können mit einem Dateisystem oder mit Swap-Speicher formatiert und dauerhaft gemountet werden.
- Volume-Gruppen kann zusätzlicher Speicher hinzugefügt und logische Volumes können dynamisch erweitert werden.

Kapitel 8

Implementieren erweiterter Storage-Features

Ziel

Verwalten Sie den Storage mithilfe des lokalen Speicherverwaltungssystems von Stratis, und verwenden Sie die VDO-Volumes, um den verwendeten Speicherplatz zu optimieren.

Ziele

- Verwalten von mehreren Storage-Ebenen mit der lokalen Speicherverwaltung von Stratis.
- Optimieren der Speicherplatznutzung mittels VDO zum Komprimieren und Deduplizieren von Daten auf Storage-Geräten.

Abschnitte

- Verwalten von Storage-Ebenen mit Stratis (mit angeleiteter Übung)
- Komprimieren und Deduplizieren von Storage mit VDO (mit angeleiteter Übung)

Praktische Übung

Implementieren erweiterter Storage-Features

Verwalten von Storage-Schichten mit Stratis

Ziele

In diesem Abschnitt wird beschrieben, wie mehrere Storage-Ebenen mit der lokalen Speicherverwaltung von Stratis verwaltet werden.

Beschreiben der Stratis-Architektur

Stratis ist eine neue lokale Storage-Managementlösung für Linux. Stratis wurde im Hinblick auf vereinfachte anfängliche Konfiguration des Storage, Änderungen an der Storage-Konfiguration und erweiterte Storage-Funktionen konzipiert.



Wichtig

Stratis ist als Technologievorschau verfügbar. Informationen zum Geltungsbereich des Red Hat-Supports für Technologievorschaufunktionen finden Sie im Dokument Geltungsbereich des Supports für Technologiefunktionen [<https://access.redhat.com/support/offers/techpreview>].

Kunden, die Stratis bereitstellen, werden gebeten, Feedback an Red Hat zu senden.

Stratis wird als Service ausgeführt, der Pools physischer Speichergeräte verwaltet. Er erstellt und verwaltet transparent Volumes für die neu erstellten Dateisysteme.

In Stratis werden Dateisysteme unter Verwendung des sog. *Thin Provisioning*-Konzepts in freigegebenen *Pools* von Disk-Geräten erstellt. Statt dem Dateisystem physischen Speicherplatz sofort bei Erstellung zuzuweisen, weist Stratis diesen Speicherplatz dynamisch aus dem Pool zu, da das Dateisystem mehr Daten speichert. Aus diesem Grund scheint das Dateisystem möglicherweise 1 TiB groß zu sein, hat jedoch möglicherweise nur 100 GiB des tatsächlichen Speichers aus dem Pool zugewiesen.

Sie können mehrere Pools von unterschiedlichen Speichergeräten erstellen. Sie können aus jedem Pool ein oder mehrere Dateisysteme erstellen. Derzeit können Sie bis zu 2^{24} Dateisysteme pro Pool erstellen.

Die Komponenten, aus denen ein von Stratis verwaltetes Dateisystem besteht, basieren auf standardmäßigen Linux-Komponenten. Intern wird Stratis mit der Device Mapper-Infrastruktur implementiert, die auch für die Implementierung von LVM verwendet wird. Von Stratis verwaltete Dateisysteme werden mit XFS formatiert.

Das folgende Diagramm zeigt, wie die Elemente der Storage-Verwaltungslösung von Stratis zusammengestellt sind. Block-Storage-Geräte, wie z. B. Festplatten oder SSDs, werden Pools zugewiesen, die jeweils physischen Speicher zum Pool beisteuern. Dateisysteme werden aus den Pools erstellt, und der physische Speicher wird jedem Dateisystem nach Bedarf zugeordnet.

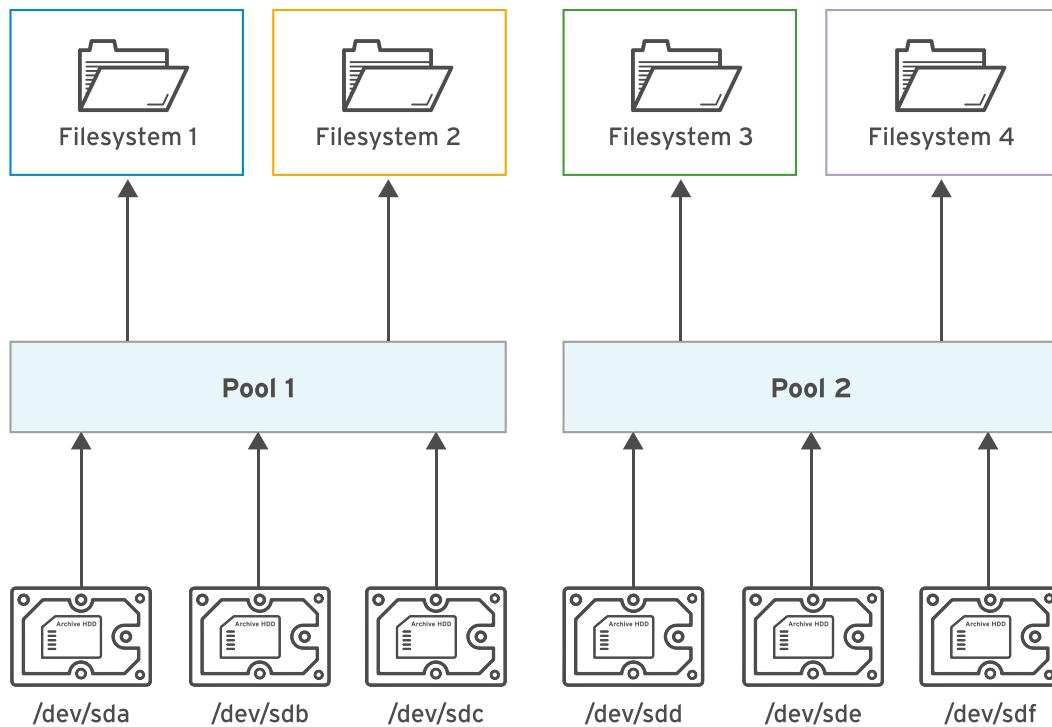


Abbildung 8.1: Elemente von Stratis

Arbeiten mit Stratis-Storage

Installieren Sie die Pakete `stratis-cli` und `stratisd`, um Dateisysteme mithilfe der Storage-Verwaltungslösung von Stratis zu verwalten. Das Paket `stratis-cli` stellt den Befehl **stratis** bereit, der Neukonfigurationsanforderungen an den System-Daemon **stratisd** sendet. Das Paket `stratisd` stellt den Service **stratisd** bereit, der die Neukonfigurationsanforderungen verarbeitet und Blockgeräte, Pools, und Dateisysteme, die Stratis verwendet, verwaltet und überwacht.

!
Warnung

Von Stratis erstellte Dateisysteme sollten nur mit Stratis-Tools und -Befehlen neu konfiguriert werden.

Stratis verwendet gespeicherte Metadaten, um verwaltete Pools, Volumes und Dateisysteme zu erkennen. Die manuelle Konfiguration von Stratis-Dateisystemen mit anderen Befehlen als den von Stratis kann zum Verlust dieser Metadaten führen und verhindern, dass Stratis die erstellten Dateisysteme erkennt.

Installieren und Aktivieren von Stratis

Für die Verwendung von Stratis müssen Sie sicherstellen, dass die Software installiert ist und der Service **stratisd** ausgeführt wird.

- Führen Sie den Befehl `yum install` aus, um `stratis-cli` und `stratisd` zu installieren.

```
[root@host ~]# yum install stratis-cli stratisd  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

- Führen Sie den Befehl **systemctl** aus, um den Service **stratisd** zu aktivieren.

```
[root@host ~]# systemctl enable --now stratisd
```

Zusammenstellen des Block-Storage in Stratis-Pools

Nachfolgend finden Sie allgemeine Verwaltungsvorgänge, die mit der Storage-Verwaltungslösung von Stratis ausgeführt werden.

- Führen Sie den Befehl **stratis pool create** aus, um Pools aus mindestens einem Blockgerät zu erstellen.

```
[root@host ~]# stratis pool create pool1 /dev/vdb
```

Jeder Pool ist ein Unterverzeichnis unter dem Verzeichnis **/stratis**.

- Führen Sie den Befehl **stratis pool list** aus, um die Liste der verfügbaren Pools anzuzeigen.

```
[root@host ~]# stratis pool list  
Name      Total Physical Size  Total Physical Used  
pool1          5 GiB           52 MiB
```



Warnung

Der Befehl **stratis pool list** ist sehr wichtig, da er zeigt, wie viel Speicherplatz in den Pools verwendet wird (und somit wie viel Speicherplatz noch verfügbar ist).

Wenn ein Pool nicht mehr genügend Speicherplatz aufweist, gehen weitere Daten, die in die zu diesem Pool gehörenden Dateisysteme geschrieben werden, verloren.

- Führen Sie den Befehl **stratis pool add-data** aus, um einem Pool zusätzliche Blockgeräte hinzuzufügen.

```
[root@host ~]# stratis pool add-data pool1 /dev/vdc
```

- Führen Sie den Befehl **stratis blockdev list** aus, um die Blockgeräte eines Pools anzuzeigen.

```
[root@host ~]# stratis blockdev list pool1  
Pool Name  Device Node    Physical Size   State  Tier  
pool1      /dev/vdb        5 GiB     In-use  Data  
pool1      /dev/vdc        5 GiB     In-use  Data
```

Verwalten von Stratis-Dateisystemen

- Führen Sie den Befehl **stratis filesystem create** aus, um ein Dateisystem aus einem Pool zu erstellen.

```
[root@host ~]# stratis filesystem create pool1 fs1
```

Die Links zu den Stratis-Dateisystemen befinden sich im Verzeichnis **/stratis/pool1**.

- Führen Sie den Befehl **stratis filesystem list** aus, um die Liste der verfügbaren Dateisysteme anzuzeigen.

```
[root@host ~]# stratis filesystem list
Pool Name  Name  Used   Created      Device        UUID
pool1      fs1   546 MiB Sep 23 2020 13:11 /stratis/pool1/fs1  31b9363badd...
```



Warnung

Der Befehl **df** meldet, dass alle neuen von Stratis verwalteten XFS-Dateisysteme 1 TiB groß sind, unabhängig davon, wie viel physischer Speicher derzeit dem Dateisystem zugewiesen ist. Da das Dateisystem per Thin Provisioning bereitgestellt wurde, verfügt der Pool möglicherweise nicht über genügend echten Speicher, um das gesamte Dateisystem zu sichern, insbesondere dann, wenn andere Dateisysteme im Pool den gesamten verfügbaren Speicher nutzen.

Daher ist es möglich, dass der gesamte Speicherplatz im Storage Pool verbraucht ist, obwohl **df** weiterhin meldet, dass das Dateisystem verfügbaren Speicherplatz aufweist. Wenn für den Pool kein Storage für das Dateisystem zur Verfügung steht, können weitere Versuche, in dieses Dateisystem zu schreiben, fehlschlagen, was zu Datenverlust führt.

Verwenden Sie den Befehl **stratis pool list**, um den verbleibenden tatsächlichen Speicherplatz in den Stratis-Pools zu überwachen.

- Mit dem Befehl **stratis filesystem snapshot** können Sie einen Snapshot eines von Stratis verwalteten Dateisystems erstellen. Snapshots sind von den Quell-Dateisystemen unabhängig.

```
[root@host ~]# stratis filesystem snapshot pool1 fs1 snapshot1
```

Dauerhaftes Mounten von Stratis-Dateisystemen

Bearbeiten Sie **/etc/fstab**, und geben Sie die Details des Dateisystems an, um sicherzustellen, dass die Stratis-Dateisysteme dauerhaft bereitgestellt sind. Wenn der folgende Befehle ausgeführt wird, wird die UUID des Dateisystems angezeigt, das Sie in **/etc/fstab** verwenden sollten, um das Dateisystem zu identifizieren.

```
[root@host ~]# lsblk --output=UUID /stratis/pool1/fs1
UUID
31b9363b-add8-4b46-a4bf-c199cd478c55
```

Kapitel 8 | Implementieren erweiterter Storage-Features

Im Folgenden finden Sie einen Beispieleintrag in der Datei **/etc/fstab**, um ein Stratis-Dateisystem dauerhaft bereitzustellen. Bei diesem Beispieleintrag handelt es sich um eine einzelne lange Zeile in der Datei.

```
UUID=31b9363b-add8-4b46-a4bf-c199cd478c55 /dir1 xfs defaults,x-
systemd.requires=stratisd.service 0 0
```

Die Bereitstellungsoption **x-systemd.requires=stratisd.service** verzögert die Bereitstellung des Dateisystems, bis **systemd** während des Bootvorgangs **stratisd.service** gestartet hat.



Wichtig

Wenn die Mount-Option **x-systemd.requires=stratisd.service** nicht in **/etc/fstab** für jedes Stratis-Dateisystem enthalten ist, kann der Rechner nicht ordnungsgemäß gestartet werden und bricht beim nächsten Neustart **emergency.target** ab.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Managing layered local storage with Stratis* im Handbuch *Red Hat Enterprise Linux 8 Configuring and Managing File Systems Guide* unter
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_file_systems/

Stratis-Storage

<https://stratis-storage.github.io/>

Was Stratis von ZFS, Btrfs und Linux Volume Manager gelernt hat

<https://opensource.com/article/18/4/stratis-lessons-learned>

► Angeleitete Übung

Verwalten von Storage-Schichten mit Stratis

In dieser Übung erstellen Sie mit Stratis Dateisysteme aus Speicherpools, die von physischen Storage-Geräten bereitgestellt werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines Dateisystems mit Thin Provisioning mithilfe der Storage-Verwaltungslösung von Stratis.
- Verifizieren des dynamischen Wachstums der Stratis-Volumes, um das Wachstum in Echtzeit zu unterstützen.
- Zugreifen auf Daten aus dem Snapshot eines Dateisystems mit Thin Provisioning.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** den Befehl **lab advstorage-stratis start** aus, um die Übung zu starten. Dieses Skript richtet die Umgebung korrekt ein und stellt sicher, dass die zusätzlichen Disks auf **servera** bereinigt sind.

```
[student@workstation ~]$ lab advstorage-stratis start
```

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Wechseln Sie zum Benutzer **root**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Installieren Sie die Pakete **stratisd** und **stratis-cli** mit dem Befehl **yum**

```
[root@servera ~]# yum install stratisd stratis-cli  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

- 4. Aktivieren Sie mit dem Befehl **systemctl** den Service **stratisd**.

```
[root@servera ~]# systemctl enable --now stratisd
```

- 5. Stellen Sie sicher, dass der Stratis-Pool **stratispool1** mit dem Blockgerät **/dev/vdb** vorhanden ist.

- 5.1. Erstellen Sie mit dem Befehl **stratis pool create** einen Stratis-Pool mit dem Namen **stratispool1**.

```
[root@servera ~]# stratis pool create stratispool1 /dev/vdb
```

- 5.2. Führen Sie den Befehl **stratis pool list** aus, um die Verfügbarkeit von **stratispool1** zu verifizieren.

```
[root@servera ~]# stratis pool list  
Name          Total Physical  
stratispool1  5 GiB / 37.63 MiB / 4.96 GiB
```

Notieren Sie die Größe des Pools in der vorhergehenden Ausgabe.

- 6. Erweitern Sie die Kapazität von **stratispool1** mit dem Blockgerät **/dev/vdc**.

- 6.1. Führen Sie den Befehl **stratis pool add-data** aus, um **stratispool1** das Blockgerät **/dev/vdc** hinzuzufügen.

```
[root@servera ~]# stratis pool add-data stratispool1 /dev/vdc
```

- 6.2. Führen Sie den Befehl **stratis pool list** aus, um die Größe von **stratispool1** zu verifizieren.

```
[root@servera ~]# stratis pool list  
Name          Total Physical  
stratispool1  10 GiB / 41.63 MiB / 9.96 GiB
```

Wie oben gezeigt, wird die Größe des Pools **stratispool1** erhöht, wenn Sie das Blockgerät hinzufügen.

- 6.3. Führen Sie den Befehl **stratis blockdev list** aus, um die Blockgeräte zu verifizieren, die aktuell Member von **stratispool1** sind.

```
[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
```

- 7. Fügen Sie im Pool **stratispool1** ein Dateisystem mit Thin Provisioning namens **stratis-fs1** hinzu. Stellen Sie das Dateisystem in **/stratisvol** bereit. Erstellen Sie eine Datei auf dem Dateisystem **stratis-fs1** namens **file1**, die den **Hello World!** enthält.

- 7.1. Führen Sie den Befehl **stratis filesystem create** aus, um das Dateisystem mit Thin Provisioning **stratis-fs1** auf **stratispool1** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
[root@servera ~]# stratis filesystem create stratispool1 stratis-fs1
```

- 7.2. Führen Sie den Befehl **stratis filesystem list** aus, um die Verfügbarkeit von **stratis-fs1** zu verifizieren.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
                  UUID
stratispool1  stratis-fs1  546 MiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-fs1  8714...e7db
```

Beachten Sie die aktuelle Nutzung von **stratis-fs1**. Diese Nutzung des Dateisystems nimmt in den folgenden Schritten nach Bedarf zu.

- 7.3. Führen Sie den Befehl **mkdir** aus, um ein Verzeichnis mit dem Namen **/stratisvol** zu erstellen.

```
[root@servera ~]# mkdir /stratisvol
```

- 7.4. Führen Sie den Befehl **mount** aus, um **stratis-fs1** auf **/stratisvol** bereitzustellen.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-fs1 /stratisvol
```

- 7.5. Führen Sie den Befehl **mount** aus, um sicherzustellen, dass das Volume **stratis-fs1** auf **/stratisvol** bereitgestellt wurde.

```
[root@servera ~]# mount
...output omitted...
/dev/mapper/stratis-1-5c0e...12b9-thin-fs-8714...e7db on /stratisvol type xfs
(rw,relatime,seclabel,attr2,inode64,sunit=2048,swidth=2048,noquota)
```

- 7.6. Führen Sie den Befehl **echo** aus, um die Textdatei **/stratisvol/file1** zu erstellen.

```
[root@servera ~]# echo "Hello World!" > /stratisvol/file1
```

- 8. Verifizieren Sie, dass das Dateisystem mit Thin Provisioning **stratis-fs1** entsprechend dem Dateisystemwachstum dynamisch wächst.

- 8.1. Führen Sie den Befehl **stratis filesystem list** aus, um die aktuelle Nutzung von **stratis-fs1** anzuzeigen.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name           Used     Created          Device
              UUID
stratispool1  stratis-filesystem1  546 MiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-filesystem1  8714...e7db
```

- 8.2. Führen Sie den Befehl **dd** aus, um eine Datei mit 2 GiB auf **stratis-filesystem1** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
[root@servera ~]# dd if=/dev/urandom of=/stratisvol/file2 bs=1M count=2048
```

- 8.3. Führen Sie den Befehl **stratis filesystem list** aus, um die Nutzung von **stratis-filesystem1** zu verifizieren.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name           Used     Created          Device
              UUID
stratispool1  stratis-filesystem1  2.53 GiB  Mar 29 2019 07:48  /stratis/
stratispool1/stratis-filesystem1  8714...e7db
```

Die vorstehende Ausgabe zeigt, dass sich die Nutzung von **stratis-filesystem1** erhöht hat. Die erhöhte Nutzung bestätigt, dass das Dateisystem mit Thin Provisioning dynamisch erweitert wurde, um das Echtzeit-Datenwachstum zu berücksichtigen, das Sie durch die Erstellung von **/stratisvol/file2** verursacht haben.

- 9. Erstellen Sie einen Snapshot von **stratis-filesystem1** namens **stratis-filesystem1-snap**. Durch den Snapshot erhalten Sie Zugriff auf Dateien, die vom **stratis-filesystem1** gelöscht wurden.

- 9.1. Führen Sie den Befehl **stratis filesystem snapshot** aus, um ein Snapshot von **stratis-filesystem1** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

Der folgende Befehl ist lang und sollte als eine einzelne Zeile eingegeben werden.

```
[root@servera ~]# stratis filesystem snapshot stratispool1 stratis-filesystem1
stratis-filesystem1-snap
```

- 9.2. Führen Sie den Befehl **stratis filesystem list** aus, um die Verfügbarkeit des Snapshots zu verifizieren.

```
[root@servera ~]# stratis filesystem list
...output omitted...
stratispool1  stratis-filesystem1-snap  2.53 GiB  Mar 29 2019 10:28  /stratis/
stratispool1/stratis-filesystem1-snap  291d...8a16
```

- 9.3. Entfernen Sie die Datei **/stratisvol/file1**.

```
[root@servera ~]# rm /stratisvol/file1  
rm: remove regular file '/stratisvol/file1'? y
```

- 9.4. Führen Sie den Befehl **mkdir** aus, um ein Verzeichnis mit dem Namen **/stratisvol-snap** zu erstellen.

```
[root@servera ~]# mkdir /stratisvol-snap
```

- 9.5. Führen Sie den Befehl **mount** aus, um den Snapshot **stratis-filesystem1-snap** auf **/stratisvol-snap** bereitzustellen.

Der folgende Befehl ist lang und sollte als eine einzelne Zeile eingegeben werden.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-filesystem1-snap /  
stratisvol-snap
```

- 9.6. Bestätigen Sie, dass Sie weiterhin auf die Datei zugreifen können, die Sie mit dem Snapshot **stratis-filesystem1-snap** aus **stratis-filesystem1** gelöscht haben.

```
[root@servera ~]# cat /stratisvol-snap/file1  
Hello World!
```

- 10. Heben Sie mit dem Befehl **umount** die Bereitstellung von **/stratisvol** und **/stratisvol-snap** auf.

```
[root@servera ~]# umount /stratisvol-snap  
[root@servera ~]# umount /stratisvol
```

- 11. Entfernen Sie das Dateisystem mit Thin Provisioning **stratis-filesystem1** und den zugehörigen Snapshot **stratis-filesystem1-snap** vom System.

- 11.1. Führen Sie den Befehl **stratis filesystem destroy** aus, um **stratis-filesystem1-snap** zu vernichten.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1-snap
```

- 11.2. Führen Sie den Befehl **stratis filesystem destroy** aus, um **stratis-filesystem1** zu vernichten.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratis-filesystem1
```

- 11.3. Beenden Sie die Shell des Benutzers **root** und melden Sie sich bei **servera** ab.

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

Beenden

Führen Sie auf der **workstation** den Befehl **lab advstorage-stratis finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Partitionen und Dateien und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab advstorage-stratis finish
```

Hiermit ist die angeleitete Übung beendet.

Komprimieren und Deduplizieren von Storage mit VDO

Ziele

In diesem Abschnitt wird beschrieben, wie die Speicherplatznutzung mittels VDO zum Komprimieren und Deduplizieren von Daten auf Storage-Geräten optimiert wird.

Beschreiben von Virtual Data Optimizer

Red Hat Enterprise Linux 8 enthält den Virtual Data Optimizer-Treiber (VDO), der den von Daten auf Blockgeräten belegten Speicherplatz optimiert. VDO ist ein Linux Device Mapper-Treiber, der den Speicherplatzbedarf auf Blockgeräten reduziert, die Replikation von Daten minimiert, Speicherplatz spart und sogar den Datendurchsatz erhöht. VDO enthält zwei Kernel-Module, das Modul **kvdo** zur transparenten Steuerung der Datenkomprimierung und das Modul **uds** zur Deduplizierung.

Die VDO-Ebene wird auf einem vorhandenen Blockspeichergerät (z. B. einem RAID-Gerät oder einer lokalen Disk) platziert. Diese Blockgeräte können auch verschlüsselte Geräte sein. Die Storage-Ebenen, z. B. logische LVM-Volumes und Dateisysteme, werden auf einem VDO-Gerät platziert. Das folgende Diagramm zeigt die Platzierung von VDO in einer Infrastruktur, die aus virtuellen KVM-Rechnern besteht, die optimierte Storage-Geräte verwenden.

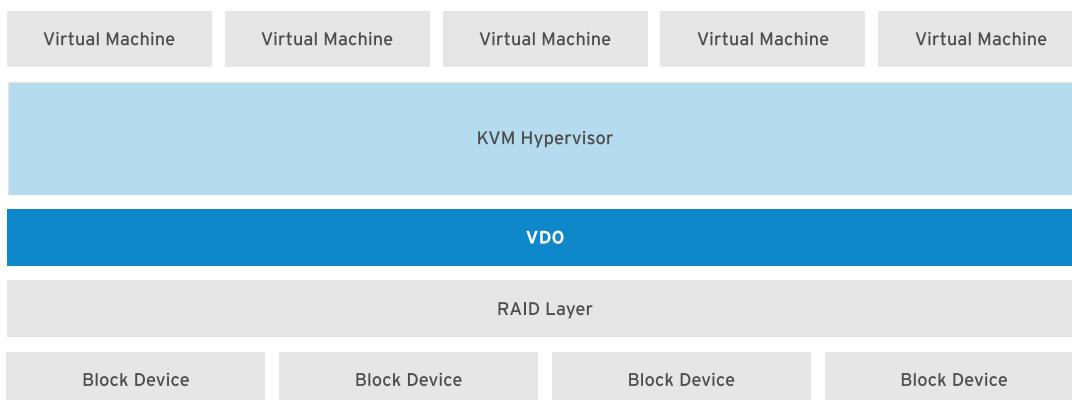


Abbildung 8.2: VDO-basierte virtuelle Rechner

VDO wendet drei Phasen in der folgenden Reihenfolge auf Daten an, um den Platzbedarf auf Storage-Geräten zu reduzieren:

1. Die *Nullblock-Eliminierung* filtert Datenblöcke heraus, die nur Nullen (0) enthalten, und zeichnet die Informationen dieser Blöcke nur in den Metadaten auf. Die Nicht-Null-Datenblöcke werden dann zur nächsten Verarbeitungsphase weitergeleitet. Diese Phase ermöglicht das Thin-Provisioning-Feature auf den VDO-Geräten.
2. Die *Deduplizierung* beseitigt redundante Datenblöcke. Wenn Sie mehrere Kopien derselben Daten erstellen, erkennt VDO die doppelten Datenblöcke und aktualisiert die Metadaten, um diese doppelten Blöcke als Verweise auf den ursprünglichen Datenblock zu verwenden, ohne redundante Datenblöcke zu erstellen. Das UDS-Kernelmodul (Universal Deduplication Service) überprüft die Redundanz der Daten durch die Metadaten, die es verwaltet. Dieses Kernelmodul ist standardmäßig in VDO enthalten.

Kapitel 8 | Implementieren erweiterter Storage-Features

3. Die **Komprimierung** ist die letzte Phase. Das Kernelmodul **kvdvdo** komprimiert Datenblöcke mittels LZ4-Komprimierung und gruppiert sie in 4 KB-Blöcke.

Implementieren von Virtual Data Optimizer

Die logischen Geräte, die Sie mit VDO erstellen, werden als **VDO-Volumes** bezeichnet. VDO-Volumes ähneln Disk-Partitionen. Sie können die Volumes mit dem gewünschten Dateisystemtyp formatieren und wie ein gewöhnliches Dateisystem bereitstellen. Sie können ein VDO-Volume auch als physisches LVM-Volume verwenden.

Geben Sie zum Erstellen eines VDO-Volumes ein Blockgerät und den Namen des logischen Geräts an, das VDO dem Benutzer anzeigt. Sie können optional die logische Größe des VDO-Volumes angeben. Die logische Größe des VDO-Volumes kann größer sein als die physische Größe des tatsächlichen Blockgeräts.

Da die VDO-Volumes mittels Thin Provisioning bereitgestellt werden, können Benutzer nur den verwendeten logischen Speicherplatz sehen. Daher wissen sie nicht, welcher physische Speicherplatz tatsächlich zur Verfügung steht. Wenn Sie beim Erstellen des Volumes keine logische Größe angeben, nimmt VDO die tatsächliche physische Größe als logische Größe des Volumes an. Dieses 1:1-Verhältnis bei der Zuordnung der logischen zur physischen Größe führt zu einer besseren Leistung, spart jedoch weniger Speicherplatz. Basierend auf Ihren Infrastrukturanforderungen sollten Sie entweder der Leistung oder der Speicherplatzeffizienz Priorität einräumen.

Wenn die logische Größe eines VDO-Volumes größer als die tatsächliche physische Größe ist, sollten Sie durch Ausführung des Befehls **vdostats --verbose** die Volume-Statistik proaktiv überwachen, um die tatsächliche Nutzung anzuzeigen.

Aktivieren von VDO

Installieren Sie die Pakete **vdo** und **kmod-kvdo**, um VDO auf dem System zu aktivieren.

```
[root@host ~]# yum install vdo kmod-kvdo
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

Erstellen eines VDO-Volumes

Führen Sie den Befehl **vdo create** aus, um ein VDO-Volume zu erstellen.

```
[root@host ~]# vdo create --name=vdo1 --device=/dev/vdd --vdoLogicalSize=50G
...output omitted...
```

Wenn Sie die logische Größe nicht angeben, erhält das resultierende VDO-Volume dieselbe Größe wie sein physisches Gerät.

Wenn das VDO-Volume vorhanden ist, können Sie es mit dem gewünschten Dateisystemtyp formatieren und es unter der Dateisystemhierarchie in Ihrem System bereitstellen.

Analysieren eines VDO-Volumes

Führen Sie den Befehl **vdo status** aus, um ein VDO-Volume zu analysieren. Dieser Befehl zeigt einen Bericht über das VDO-System und den Status des VDO-Volumes im YAML-Format

an. Er zeigt auch Attribute des VDO-Volumes an. Verwenden Sie die Option **--name=**, um den Namen eines bestimmten Volumes anzugeben. Wenn Sie den Namen des bestimmten Volumes weglassen, zeigt die Ausgabe des Befehls **vdo status** den Status sämtlicher VDO-Volumes an.

```
[root@host ~]# vdo status --name=vdo1  
...output omitted...
```

Der Befehl **vdo list** zeigt die Liste der VDO-Volumes an, die aktuell gestartet werden. Mit den Befehlen **vdo start** bzw. **vdo stop** können Sie ein VDO-Volume starten und stoppen.



Literaturhinweise

Weitere Informationen finden Sie im Kapitel *Getting started with VDO* im Handbuch *Red Hat Enterprise Linux 8 Deduplication and Compression Storage Guide* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/deduplicating_and_compressing_storage/

Einführung in Virtual Data Optimizer

<https://rhelblog.redhat.com/2018/04/11/introducing-virtual-data-optimizer-to-reduce-cloud-and-on-premise-storage-costs/>

► Angeleitete Übung

Komprimieren und Deduplizieren von Storage mit VDO

In dieser Übung erstellen Sie ein VDO-Volume, formatieren es mit einem Dateisystem, stellen es bereit, speichern Daten darauf und untersuchen die Auswirkungen von Komprimierung und Deduplizierung auf den tatsächlich verwendeten Speicherplatz.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines Volumes mit Virtual Data Optimizer, Formatieren des Volumes mit einem Dateisystemtyp und Bereitstellen eines Dateisystems auf dem Volume.
- Untersuchen der Auswirkungen der Datendeduplizierung und -komprimierung auf einem Virtual Data Optimizer-Volume.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** den Befehl **lab advstorage-vdo start** aus, um die Übung zu starten. Dieses Skript stellt sicher, dass auf der Disk **/dev/vdd** keine Partitionen vorhanden sind. Zudem richtet es die Umgebung korrekt ein.

```
[student@workstation ~]$ lab advstorage-vdo start
```

- 1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 2. Erstellen Sie das VDO-Volume **vdo1** mithilfe des Geräts **/dev/vdd**. Legen Sie die zugehörige Größe auf 50 GB fest.

2.1. Wechseln Sie zum Benutzer **root**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2.2. Führen Sie den Befehl **yum** aus, um zu bestätigen, dass das Paket **vdo** installiert ist.

```
[root@servera ~]# yum list installed vdo
Installed Packages
vdo.x86_64          6.2.2.117-13.el8          @rhel-8-for-x86_64-baseos-rpms
```

- 2.3. Führen Sie den Befehl **vdo create** aus, um das Volume **vdo1** zu erstellen.

```
[root@servera ~]# vdo create --name=vdo1 \
--device=/dev/vdd --vdoLogicalSize=50G
...output omitted...
```

- 2.4. Führen Sie den Befehl **vdo list** aus, um die Verfügbarkeit des Volumes **vdo1** zu verifizieren.

```
[root@servera ~]# vdo list
vdo1
```

- ▶ 3. Verifizieren Sie, dass die Komprimierungs- und Deduplizierungsfeatures auf dem Volume **vdo1** aktiviert sind.

Verwenden Sie **grep**, um nach den Zeilen zu suchen, in denen die Zeichenfolge **Deduplication** und **Compression** in der Ausgabe des Befehls **vdo status --name=vdo1** enthalten ist.

```
[root@servera ~]# vdo status --name=vdo1 \
| grep -E 'Deduplication|Compression'
Compression: enabled
Deduplication: enabled
```

- ▶ 4. Formatieren Sie das Volume **vdo1** mit dem **XFS**-Dateisystemtyp und stellen Sie es auf **/mnt/vdo1** bereit.

- 4.1. Verwenden Sie den Befehl **udevadm**, um zu verifizieren, dass die neue VDO-Gerätedatei erstellt wurde.

```
[root@servera ~]# udevadm settle
```

- 4.2. Führen Sie den Befehl **mkfs** aus, um das Volume **vdo1** mit dem **XFS**-Dateisystem zu formatieren.

```
[root@servera ~]# mkfs.xfs -K /dev/mapper/vdo1
...output omitted...
```

Die Option **-K** im vorhergehenden Befehl **mkfs.xfs** verhindert, dass nicht verwendete Blöcke im Dateisystem sofort verworfen werden, wodurch der Befehl schneller zurückgegeben wird.

- 4.3. Führen Sie den Befehl **mkdir** aus, um das Verzeichnis **/mnt/vdo1** zu erstellen.

```
[root@servera ~]# mkdir /mnt/vdo1
```

- 4.4. Führen Sie den Befehl **mount** aus, um das Volume **vdo1** auf **/mnt/vdo1** bereitzustellen.

```
[root@servera ~]# mount /dev/mapper/vdo1 /mnt/vdo1
```

- 4.5. Führen Sie den Befehl **mount** aus, um zu verifizieren, dass das Volume **vdo1** erfolgreich bereitgestellt ist.

```
[root@servera ~]# mount
...output omitted...
/dev/mapper/vdo1 on /mnt/vdo1 type xfs
(rw,relatime,seclabel,attr2,inode64,noquota)
```

- 5. Erstellen Sie drei Kopien derselben Datei namens **/root/install.img** auf dem Volume **vdo1**. Vergleichen Sie die Statistiken des Volumes, um die Datendeduplizierung und -komprimierung des Volumes zu verifizieren. Die vorherige Ausgabe kann auf Ihrem System abweichen.
- 5.1. Führen Sie den Befehl **vdostats** aus, um die Anfangsstatistiken und den Status des Volumes anzuzeigen.

```
[root@servera ~]# vdostats --human-readable
Device           Size     Used   Available  Use% Space saving%
/dev/mapper/vdo1    5.0G   3.0G      2.0G   60%       99%
```

Beachten Sie, dass 3 GB des Volumes bereits bei der Erstellung verwendet werden, da das VDO-Volume 3-4 GB für sich selbst reserviert. Beachten Sie zudem, dass der Wert **99%** im Feld **Space saving%** angibt, dass Sie noch keinen Inhalt auf dem Volume erstellt haben, der zu dem gesamten gespeicherten Volume-Speicherplatz beiträgt.

- 5.2. Kopieren Sie **/root/install.img** in **/mnt/vdo1/install.img.1**, und überprüfen Sie die Statistiken des Volume. Das Kopieren der Datei kann bis zu einer Minute dauern.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.1
[root@servera ~]# vdostats --human-readable
Device           Size     Used   Available  Use% Space saving%
/dev/mapper/vdo1    5.0G   3.4G      1.6G   68%       5%
```

Beachten Sie, dass der Wert des Felds **Used** von **3.0G** auf **3.4G** erhöht wird, weil Sie eine Datei auf das Volume kopiert haben, die etwas Speicherplatz beansprucht. Beachten Sie außerdem, dass sich der Wert des Felds **Space saving%** von **99%** auf **5%** verringert, weil sich anfangs kein Inhalt im Volume befand, was zu einer geringen Speicherplatzauslastung und zu einer hohen Speicherplatzersparnis beitrug, bis Sie eine Datei erstellt haben. Der Speicherplatz auf dem Volume ist vergleichsweise gering, da Sie eine eindeutige Kopie der Datei auf dem Volume erstellt haben und nichts zu deduplizieren ist.

- 5.3. Kopieren Sie **/root/install.img** in **/mnt/vdo1/install.img.2** und überprüfen Sie die Statistiken des Volume. Das Kopieren der Datei kann bis zu einer Minute dauern.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.2
[root@servera ~]# vdostats --human-readable
Device           Size     Used   Available  Use% Space saving%
/dev/mapper/vdo1    5.0G   3.4G      1.6G   68%       51%
```

Beachten Sie, dass sich der verwendete Speicherplatz auf dem Volume nicht geändert hat. Stattdessen stieg der Prozentsatz des gespeicherten Volume-Speicherplatzes an, was belegt, dass die Datendeduplizierung stattgefunden hat, um den Speicherplatzbedarf für die redundanten Kopien derselben Datei zu reduzieren. Der Wert von **Space saving%** in der vorherigen Ausgabe kann auf Ihrem System abweichen.

- 5.4. Beenden Sie die Shell des Benutzers **root** und melden Sie sich bei **servera** ab.

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

Beenden

Führen Sie auf der **workstation** den Befehl **lab advstorage-vdo finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab advstorage-vdo finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Implementieren erweiterter Storage-Features

In dieser Übung verwenden Sie die Stratis-Speicherverwaltungslösung zum Erstellen von Dateisystemen, die mit dem steigenden Datenbedarf wachsen, und Virtual Data Optimizer zum Erstellen von Datenträgern, um Speicherplatz effizient zu nutzen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines Dateisystems mit Thin Provisioning mithilfe der Storage-Verwaltungslösung von Stratis.
- Verifizieren des dynamischen Wachstums der Stratis-Volumes, um das Wachstum in Echtzeit zu unterstützen.
- Zugreifen auf Daten aus dem Snapshot eines Dateisystems mit Thin Provisioning.
- Erstellen eines Volumes mit Virtual Data Optimizer und Bereitstellen dieses auf einem Dateisystem.
- Untersuchen der Auswirkungen der Datendeduplizierung und -komprimierung auf einem Virtual Data Optimizer-Volume.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** den Befehl **lab advstorage-review start** aus, um die praktische Übung zu starten. Dieses Skript richtet die Umgebung korrekt ein und stellt sicher, dass die zusätzlichen Disks auf **serverb** bereinigt sind.

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

1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **serverb**.
2. Wechseln Sie zum Benutzer **root**.
3. Installieren Sie die Pakete **stratisd** und **stratis-cli** mit **yum**.
4. Starten Sie und aktivieren Sie den Service **stratisd** mit dem Befehl **systemctl**.
5. Erstellen Sie den Stratis-Pool **labpool1**, der das Blockgerät **/dev/vdb** enthält.
6. Erweitern Sie die Kapazität von **labpool1** um die im System **/dev/vdc** verfügbare Disk.
7. Erstellen Sie ein Dateisystem mit Thin Provisioning namens **labfs** im Pool **labpool1**. Stellen Sie dieses Dateisystem auf **/labstratisvol** bereit, sodass es auch nach Neustarts erhalten bleibt. Erstellen Sie auf dem Dateisystem **labfs** eine Datei namens **labfile1**, die den Text **Hello World!** enthält. Vergessen Sie nicht, die Bereitstellungsoption **x-systemd.requires=stratisd.service** in **/etc/fstab** zu verwenden.

8. Verifizieren Sie, dass das Dateisystem mit Thin Provisioning **labfs** entsprechend dem Dateisystemwachstum dynamisch wächst, indem Sie dem Dateisystem eine **labfile2** mit 2 GiB hinzufügen.
9. Erstellen Sie einen Snapshot namens **labfs-snap** des Dateisystems **labfs**. Mit dem Snapshot können Sie auf eine beliebige Datei zugreifen, die aus **labfs** gelöscht wird.
10. Erstellen Sie das VDO-Volume **labvdo** mit dem Gerät **/dev/vdd**. Legen Sie die logische Größe auf **50 GB** fest.
11. Stellen Sie das Volume **labvdo** auf **/labvdovol** mit dem **XFS**-Dateisystem bereit, sodass es auch nach erneuten Bootvorgängen erhalten bleibt. Vergessen Sie nicht, die Bereitstellungsoption **x-systemd.requires=vdo.service** in **/etc/fstab** zu verwenden.
12. Erstellen Sie drei Kopien derselben Datei namens **/root/install.img** auf dem Volume **labvdo**. Vergleichen Sie die Statistiken des Volumes, um die Datendeduplizierung und -komprimierung des Volumes zu verifizieren.
13. Booten Sie **serverb** neu. Überprüfen Sie, ob Ihr Volume **labvdo** auf **/labvdovol** bereitgestellt wurde, nachdem das System wieder gestartet wurde.

Bewertung

Führen Sie auf **workstation** den Befehl **lab advstorage-review grade** aus, um den Erfolg dieser Übung zu überprüfen.

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

Beenden

Führen Sie auf der **workstation** den Befehl **lab advstorage-review finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Partitionen und Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Implementieren erweiterter Storage-Features

In dieser Übung verwenden Sie die Stratis-Speicherverwaltungslösung zum Erstellen von Dateisystemen, die mit dem steigenden Datenbedarf wachsen, und Virtual Data Optimizer zum Erstellen von Datenträgern, um Speicherplatz effizient zu nutzen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines Dateisystems mit Thin Provisioning mithilfe der Storage-Verwaltungslösung von Stratis.
- Verifizieren des dynamischen Wachstums der Stratis-Volumes, um das Wachstum in Echtzeit zu unterstützen.
- Zugreifen auf Daten aus dem Snapshot eines Dateisystems mit Thin Provisioning.
- Erstellen eines Volumes mit Virtual Data Optimizer und Bereitstellen dieses auf einem Dateisystem.
- Untersuchen der Auswirkungen der Datendeduplizierung und -komprimierung auf einem Virtual Data Optimizer-Volume.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie auf **workstation** den Befehl **lab advstorage-review start** aus, um die praktische Übung zu starten. Dieses Skript richtet die Umgebung korrekt ein und stellt sicher, dass die zusätzlichen Disks auf **serverb** bereinigt sind.

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

1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **serverb**.

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

2. Wechseln Sie zum Benutzer **root**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. Installieren Sie die Pakete **stratisd** und **stratis-cli** mit **yum**.

```
[root@serverb ~]# yum install stratisd stratis-cli  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

4. Starten Sie und aktivieren Sie den Service **stratisd** mit dem Befehl **systemctl**.

```
[root@serverb ~]# systemctl enable --now stratisd
```

5. Erstellen Sie den Stratis-Pool **labpool**, der das Blockgerät **/dev/vdb** enthält.

- 5.1. Führen Sie den Befehl **stratis pool create** aus, um den Stratis-Pool **labpool** zu erstellen.

```
[root@serverb ~]# stratis pool create labpool /dev/vdb
```

- 5.2. Führen Sie den Befehl **stratis pool list** aus, um die Verfügbarkeit von **labpool** zu verifizieren.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical  
labpool  5 GiB / 37.63 MiB / 4.96 GiB
```

Notieren Sie die Größe des Pools in der vorhergehenden Ausgabe.

6. Erweitern Sie die Kapazität von **labpool** um die im System **/dev/vdc** verfügbare Disk.

- 6.1. Führen Sie den Befehl **stratis pool add-data** aus, um **labpool** das Blockgerät **/dev/vdc** hinzuzufügen.

```
[root@serverb ~]# stratis pool add-data labpool /dev/vdc
```

- 6.2. Führen Sie den Befehl **stratis pool list** aus, um die Größe von **labpool** zu verifizieren.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical  
labpool  10 GiB / 41.63 MiB / 9.96 GiB
```

Die vorstehende Ausgabe zeigt, dass die Größe von **labpool** zugenommen hat, nachdem dem Pool eine neue Disk hinzugefügt wurde.

- 6.3. Führen Sie den Befehl **stratis blockdev list** aus, um die Blockgeräte aufzulisten, die nun Member von **labpool** sind.

```
[root@serverb ~]# stratis blockdev list labpool  
Pool Name  Device Node  Physical Size  Tier  
labpool    /dev/vdb           5 GiB  Data  
labpool    /dev/vdc           5 GiB  Data
```

7. Erstellen Sie ein Dateisystem mit Thin Provisioning namens **labfs** im Pool **labpool**. Stellen Sie dieses Dateisystem auf **/labstratisvol** bereit, sodass es auch nach Neustarts erhalten bleibt. Erstellen Sie auf dem Dateisystem **labfs** eine Datei namens **labfile1**, die den Text **Hello World!** enthält. Vergessen Sie nicht, die Bereitstellungsoption **x-systemd.requires=stratisd.service** in **/etc/fstab** zu verwenden.
- 7.1. Führen Sie den Befehl **stratis filesystem create** aus, um das Dateisystem **labfs** mit Thin Provisioning in **labpool** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
[root@serverb ~]# stratis filesystem create labpool labfs
```

- 7.2. Führen Sie den Befehl **stratis filesystem list** aus, um die Verfügbarkeit von **labfs** zu verifizieren.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used       Created        Device
                  UUID
labpool  labfs  546 MiB  Mar 29 2019 07:48  /stratis/labpool/labfs  9825...d6ca
```

Beachten Sie die aktuelle Nutzung von **labfs**. Diese Nutzung des Dateisystems nimmt in den folgenden Schritten nach Bedarf zu.

- 7.3. Führen Sie den Befehl **lsblk** aus, um die UUID von **labfs** zu bestimmen.

```
[root@serverb ~]# lsblk --output=UUID /stratis/labpool/labfs
UUID
9825e289-fb08-4852-8290-44d1b8f0d6ca
```

- 7.4. Bearbeiten Sie **/etc/fstab**, sodass das Dateisystem mit Thin Provisioning **labfs** zur Laufzeit bereitgestellt wird. Verwenden Sie die UUID, die Sie im vorherigen Schritt ermittelt haben. Im Folgenden wird die Zeile gezeigt, die Sie **/etc/fstab** hinzufügen sollten. Sie können den Befehl **vi /etc/fstab** verwenden, um die Datei zu bearbeiten.

```
UUID=9825...d6ca /labstratisvol xfs defaults,x-systemd.requires=stratisd.service
0 0
```

- 7.5. Führen Sie den Befehl **mkdir** aus, um ein Verzeichnis mit dem Namen **/labstratisvol** zu erstellen.

```
[root@serverb ~]# mkdir /labstratisvol
```

- 7.6. Führen Sie zum Bereitstellen des Dateisystems mit Thin Provisioning **labfs** den Befehl **mount** aus, um zu bestätigen, dass in der Datei **/etc/fstab** die entsprechenden Einträge vorhanden sind.

```
[root@serverb ~]# mount /labstratisvol
```

Wenn der vorhergehende Befehl Fehler verursacht, sollten Sie die Datei **/etc/fstab** erneut aufrufen und sicherstellen, dass sie die entsprechenden Einträge enthält.

Kapitel 8 | Implementieren erweiterter Storage-Features

- 7.7. Führen Sie den Befehl **echo** aus, um die Textdatei **/labstratisvol/labfile1** zu erstellen.

```
[root@serverb ~]# echo "Hello World!" > /labstratisvol/labfile1
```

8. Verifizieren Sie, dass das Dateisystem mit Thin Provisioning **labfs** entsprechend dem Dateisystemwachstum dynamisch wächst, indem Sie dem Dateisystem eine **labfile2** mit 2 GiB hinzufügen.
- 8.1. Führen Sie den Befehl **stratis filesystem list** aus, um die aktuelle Nutzung von **labfs** anzuzeigen.

stratis filesystem list					
Pool	Name	Name	Used	Created	Device
				UUID	
labpool	labfs	546 MiB	Mar 29 2019 07:48	/stratis/labpool/labfs	9825...d6ca

- 8.2. Führen Sie den Befehl **dd** aus, um eine Datei mit 2 GiB in **labfs** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
[root@serverb ~]# dd if=/dev/urandom of=/labstratisvol/labfile2 bs=1M count=2048
```

- 8.3. Führen sie den Befehl **stratis filesystem list** aus, um zu verifizieren, dass die Nutzung von **labfs** zugenommen hat.

stratis filesystem list					
Pool	Name	Name	Used	Created	Device
				UUID	
labpool	labfs	2.53 GiB	Mar 29 2019 07:48	/stratis/labpool/labfs	9825...d6ca

9. Erstellen Sie einen Snapshot namens **labfs-snap** des Dateisystems **labfs**. Mit dem Snapshot können Sie auf eine beliebige Datei zugreifen, die aus **labfs** gelöscht wird.
- 9.1. Führen Sie den Befehl **stratis filesystem snapshot** aus, um ein Snapshot von **labfs** zu erstellen. Es kann bis zu einer Minute dauern, bis der Befehl abgeschlossen ist.

```
[root@serverb ~]# stratis filesystem snapshot labpool \
labfs labfs-snap
```

- 9.2. Führen Sie den Befehl **stratis filesystem list** aus, um die Verfügbarkeit des Snapshots zu verifizieren.

```
[root@serverb ~]# stratis filesystem list
...output omitted...
labpool labfs-snap 2.53 GiB Mar 29 2019 10:28 /stratis/labpool/labfs-snap
291d...8a16
```

- 9.3. Entfernen Sie die Datei **/labstratisvol/labfile1**.

Kapitel 8 | Implementieren erweiterter Storage-Features

```
[root@serverb ~]# rm /labstratisvol/labfile1
rm: remove regular file '/labstratisvol/labfile1'? y
```

- 9.4. Führen Sie den Befehl **mkdir** aus, um das Verzeichnis **/labstratisvol-snap** zu erstellen.

```
[root@serverb ~]# mkdir /labstratisvol-snap
```

- 9.5. Führen Sie den Befehl **mount** aus, um den Snapshot **labfs-snap** auf **/labstratisvol-snap** bereitzustellen.

```
[root@serverb ~]# mount /stratis/labpool/labfs-snap \
/labstratisvol-snap
```

- 9.6. Bestätigen Sie, dass Sie mit dem Snapshot **labfs-snap** weiterhin auf die Datei zugreifen können, die Sie von **labfs** gelöscht haben.

```
[root@serverb ~]# cat /labstratisvol-snap/labfile1
Hello World!
```

- 10.** Erstellen Sie das VDO-Volume **labvdo** mit dem Gerät **/dev/vdd**. Legen Sie die logische Größe auf **50 GB** fest.

- 10.1. Führen Sie den Befehl **vdo create** aus, um das Volume **labvdo** zu erstellen.

```
[root@serverb ~]# vdo create --name=labvdo --device=/dev/vdd --vdoLogicalSize=50G
...output omitted...
```

- 10.2. Führen Sie den Befehl **vdo list** aus, um die Verfügbarkeit des Volumes **labvdo** zu verifizieren.

```
[root@serverb ~]# vdo list
labvdo
```

- 11.** Stellen Sie das Volume **labvdo** auf **/labvdovol** mit dem **XFS**-Dateisystem bereit, sodass es auch nach erneuten Bootvorgängen erhalten bleibt. Vergessen Sie nicht, die Bereitstellungsoption **x-systemd.requires=vdo.service** in **/etc/fstab** zu verwenden.

- 11.1. Führen Sie den Befehl **mkfs** aus, um das Volume **labvdo** mit dem **XFS**-Dateisystem zu formatieren.

```
[root@serverb ~]# mkfs.xfs -K /dev/mapper/labvdo
...output omitted...
```

- 11.2. Führen Sie den Befehl **udevadm** aus, um den neuen Geräteknoten zu registrieren.

```
[root@serverb ~]# udevadm settle
```

- 11.3. Führen Sie den Befehl **mkdir** aus, um das Verzeichnis **/labvdovol** zu erstellen.

```
[root@serverb ~]# mkdir /labvdovol
```

- 11.4. Führen Sie den Befehl **lsblk** aus, um die UUID von **labvdo** zu bestimmen.

```
[root@serverb ~]# lsblk --output=UUID /dev/mapper/labvdo
UUID
ef8cce71-228a-478d-883d-5732176b39b1
```

- 11.5. Bearbeiten Sie **/etc/fstab**, sodass **labvdo** zur Startzeit bereitgestellt wird. Verwenden Sie die UUID des Volumes, die Sie im vorherigen Schritt ermittelt haben. Im Folgenden wird die Zeile gezeigt, die Sie **/etc/fstab** hinzufügen sollten. Sie können den Befehl **vi /etc/fstab** verwenden, um die Datei zu bearbeiten.

```
UUID=ef8c...39b1 /labvdovol xfs defaults,x-systemd.requires=vdo.service 0 0
```

- 11.6. Führen Sie den Befehl **mount** aus, um das Volume **labvdo** bereitzustellen, um zu bestätigen, dass die Datei **/etc/fstab** die entsprechenden Einträge aufweist.

```
[root@serverb ~]# mount /labvdovol
```

Wenn der vorhergehende Befehl Fehler verursacht, sollten Sie die Datei **/etc/fstab** erneut aufrufen und sicherstellen, dass sie die entsprechenden Einträge enthält.

12. Erstellen Sie drei Kopien derselben Datei namens **/root/install.img** auf dem Volume **labvdo**. Vergleichen Sie die Statistiken des Volumes, um die Datendeduplikierung und -komprimierung des Volumes zu verifizieren.

- 12.1. Führen Sie den Befehl **vdostats** aus, um die Anfangsstatistiken und den Status des Volumes anzuzeigen.

```
[root@serverb ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.0G      2.0G  60%      99%
```

Beachten Sie, dass 3 GB des Volumes bereits bei der Erstellung verwendet werden, da das VDO-Volume 3–4 GB für sich selbst reserviert. Beachten Sie zudem, dass der Wert **99%** im Feld **Space saving%** angibt, dass Sie noch keinen Inhalt auf dem Volume erstellt haben, der zu dem gesamten gespeicherten Volume-Speicherplatz beiträgt.

- 12.2. Kopieren Sie **/root/install.img** in **/labvdovol/install.img.1**, und überprüfen Sie die Statistiken des Volume. Das Kopieren der Datei kann bis zu einer Minute dauern.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.1
[root@serverb ~]# vdostats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.4G      1.6G  68%      5%
```

Beachten Sie, dass der Wert des Felds **Used** von **3.0G** auf **3.4G** erhöht wird, da Sie eine Datei auf das Volume kopiert haben, die etwas Speicherplatz beansprucht. Beachten Sie außerdem, dass sich der Wert des Felds **Space saving%** von **99%** auf **5%** verringert, weil sich anfangs kein Inhalt im Volume befand, was zu einer geringen

Speicherplatzauslastung und zu einer hohen Speicherplatzersparnis beitrug, bis Sie darin eine Datei erstellt haben. Der Speicherplatz auf dem Volume ist ziemlich gering, da Sie eine eindeutige Kopie der Datei auf dem Volume erstellt haben und nichts zu deduplizieren ist.

- 12.3. Kopieren Sie **/root/install.img** in **/labvdovol/install.img.2**, und überprüfen Sie die Statistiken des Volume. Das Kopieren der Datei kann bis zu einer Minute dauern.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.2
[root@serverb ~]# vdfstats --human-readable
Device           Size     Used Available Use% Space saving%
/dev/mapper/labvdo    5.0G   3.4G     1.6G  68%      51%
```

Beachten Sie, dass sich der verwendete Speicherplatz auf dem Volume nicht geändert hat. Stattdessen stieg der Prozentsatz des gespeicherten Volumen-Speicherplatzes an, was belegt, dass die Datendeduplikierung stattgefunden hat, um den Speicherplatzbedarf für die redundanten Kopien derselben Datei zu reduzieren. Der Wert von **Space saving%** in der vorherigen Ausgabe kann auf Ihrem System abweichen.

13. Booten Sie **serverb** neu. Überprüfen Sie, ob Ihr Volume **labvdo** auf **/labvdovol** bereitgestellt wurde, nachdem das System wieder gestartet wurde.

- 13.1. Booten Sie den Rechner „serverb“ neu.

```
[root@serverb ~]# systemctl reboot
```



Anmerkung

Hinweis: Wenn beim erneuten Booten „serverb“ nicht über eine reguläre Anmeldeaufforderung gebootet wird, sondern stattdessen „Give root password for maintenance (or press Control-D to continue):“ angezeigt wird, haben Sie wahrscheinlich einen Fehler in **/etc/fstab** gemacht. Nach Eingabe des Root-Passworts **redhat** müssen Sie das Root-Dateisystem mit Lese-/Schreibzugriff mit dem folgenden Befehl erneut mounten:

```
[root@serverb ~]# mount -o remount,rw /
```

Überprüfen Sie, ob **/etc/fstab** entsprechend den Lösungen ordnungsgemäß konfiguriert ist. Achten Sie besonders auf die Bereitstellungsoptionen in den Zeilen für **/labstratisvol** und **/labvdovol**.

Bewertung

Führen Sie auf **workstation** den Befehl **lab advstorage-review grade** aus, um den Erfolg dieser Übung zu überprüfen.

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

Beenden

Führen Sie auf der **workstation** den Befehl **lab advstorage-review finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Partitionen und Dateien und sorgt für eine bereinigte Umgebung.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die Storage-Verwaltungslösung von Stratis implementiert flexible Dateisysteme, die dynamisch entsprechend den Daten wachsen.
- Die Storage-Verwaltungslösung von Stratis unterstützt Thin Provisioning, Snapshots und Überwachung.
- Mit Virtual Data Optimizer (VDO) sollen die Datenspeicherungskosten reduziert werden.
- Virtual Data Optimizer wendet die Nullblock-Eliminierung, Datendeduplizierung und die Datenkomprimierung an, um die Effizienz des Disk-Speichers zu optimieren.

Kapitel 9

Zugreifen auf Network-Attached Storage

Ziel

Zugreifen auf Network-Attached Storage mit dem NFS-Protokoll.

Ziele

- Einhängen, Verwenden und Aushängen eines NFS-Exports über die Befehlszeile und zur Startzeit.
- Konfiguration des Automounters mit direkten und indirekten Zuordnungen, um bei Bedarf automatisch ein NFS-Dateisystem einzubinden (mount) und es zu entfernen (umount), wenn es nicht mehr verwendet wird.

Abschnitte

- Mounten von Network-Attached Storage mit NFS (und angeleitete Übung)
- Automatisches Mounten von Network-Attached Storage (und angeleitete Übung)

Praktische Übung

Zugreifen auf Network-Attached Storage

Mounten von Network-Attached Storage mit NFS

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Ermitteln von NFS-Freigabeinformationen.
- Erstellen Sie ein Verzeichnis zur Verwendung als Bereitstellungspunkt.
- Stellen Sie eine NFS-Freigabe durch Ausführen des Befehls **mount** oder durch Konfigurieren der Datei **/etc/fstab** bereit.
- Führen Sie den Befehl **umount** aus, um die Bereitstellung einer NFS-Freigabe aufzuheben.

Bereitstellen und Aufheben der Bereitstellung von NFS-Freigaben

Das NFS (*Network File System*) ist ein Internet-Standardprotokoll, das von Linux, UNIX und ähnlichen Betriebssystemen als natives Network File System verwendet wird. Es handelt sich um einen offenen Standard, der aktiv erweitert wird und native Linux-Berechtigungen und Dateisystemfeatures unterstützt.

Die NFS-Standardversion in Red Hat Enterprise Linux 8 lautet 4.2. Unterstützt werden NFSv4- und NFSv3-Hauptversionen. NFSv2 wird nicht mehr unterstützt. NFSv4 verwendet ausschließlich das TCP-Protokoll für die Kommunikation mit dem Server. Frühere NFS-Versionen konnten TCP oder UDP verwenden.

NFS-Server exportieren Freigaben (Verzeichnisse). NFS-Clients stellen eine exportierte Freigabe auf einem lokalen Bereitstellungspunkt (Verzeichnis) bereit, der vorhanden sein muss. Um NFS-Freigaben einzuhängen, gibt es verschiedene Vorgehensweisen:

- Manuell, durch Ausführen des Befehls **mount**.
- Automatisch zur Startzeit mittels **/etc/fstab**-Einträgen.
- Nach Bedarf, entweder durch den Service **autofs** oder durch die Funktion **systemd.automount**.

Mounten von NFS-Freigaben

Führen Sie die folgenden drei Schritte aus, um eine NFS-Freigabe bereitzustellen:

1. **Identifizieren:** Der Administrator des NFS-Clientsystems kann verfügbare NFS-Freigaben auf verschiedene Arten ermitteln:

Der für den NFS-Server zuständige Administrator kann Exportdetails, einschließlich Sicherheitsanforderungen, zur Verfügung stellen.

Alternativ kann der Clientadministrator NFSv4-Freigaben ermitteln, indem das Verzeichnis „root“ des NFS-Servers bereitgestellt wird und die exportierten Verzeichnisse untersucht werden. Erledigen Sie dies als der Benutzer **root**. Der Zugriff auf Freigaben, die Kerberos-Sicherheit verwenden, wird verweigert, aber der Name der Freigabe (Verzeichnis) ist sichtbar. Andere freigegebene Verzeichnisse sind durchsuchbar.

```
[user@host ~]$ sudo mkdir mountpoint
[user@host ~]$ sudo mount server:/share mountpoint
[user@host ~]$ sudo ls mountpoint
```

2. **Bereitstellungspunkt:** Verwenden Sie **mkdir**, um an einer geeigneten Stelle einen Bereitstellungspunkt zu erstellen.

```
[user@host ~]$ mkdir -p mountpoint
```

3. **Mounten:** Wie bei Dateisystemen auf Partitionen müssen NFS-Freigaben gemountet werden, um verfügbar zu sein. Wählen Sie eine der folgenden Optionen aus, um eine NFS-Freigabe bereitzustellen. In jedem Fall müssen Sie diese Befehle als Superuser ausführen, indem Sie sich als **root** anmelden oder den Befehl **sudo** ausführen.
- Temporäre Bereitstellung: Führen Sie den Befehl **mount** aus, um die NFS-Freigabe bereitzustellen.

```
[user@host ~]$ sudo mount -t nfs -o rw,sync server:/share mountpoint
```

Die Option **-t nfs** ist der Dateisystemtyp für NFS-Freigaben (nicht zwingend erforderlich, aufgeführt aus Gründen der Vollständigkeit). Die Option **-o sync** veranlasst **mount**, Schreiboperationen sofort mit dem NFS-Server zu synchronisieren (die Standardeinstellung ist **asynchron**).

Dieser Befehl stellt die Freigabe sofort, aber nicht dauerhaft bereit. Beim nächsten Systemstart ist diese NFS-Freigabe nicht verfügbar. Dies ist nützlich für den einmaligen Zugriff auf Daten. Es ist auch nützlich, um eine NFS-Freigabe zu testen, bevor die Freigabe dauerhaft verfügbar gemacht wird.

- Dauerhafte Bereitstellung: Bearbeiten Sie zum Hinzufügen des Bereitstellungseintrags die Datei **/etc/fstab**, um sicherzustellen, dass die NFS-Freigabe zur Startzeit bereitgestellt wird.

```
[user@host ~]$ sudo vim /etc/fstab
...
server:/share mountpoint nfs rw,soft 0 0
```

Stellen Sie anschließend die NFS-Freigabe bereit:

```
[user@host ~]$ sudo mount mountpoint
```

Da der NFS-Clientservice den NFS-Server und die Bereitstellungsoptionen in der Datei **/etc/fstab** findet, müssen Sie diese nicht an der Befehlszeile angeben.

Aufheben der Bereitstellung von NFS-Freigaben

Führen Sie den Befehl **umount** aus, um als **root** (oder mit **sudo**) die Bereitstellung einer NFS-Freigabe aufzuheben.

```
[user@host ~]$ sudo umount mountpoint
```



Anmerkung

Wenn die Bereitstellung einer Freigabe aufgehoben wird, wie ihr zugehöriger /etc/fstab-Eintrag nicht entfernt. Wenn Sie den Eintrag nicht entfernen oder auskommentieren, wird die NFS-Freigabe entweder beim nächsten Systemstart oder beim Neustart des NFS-Clientservices erneut bereitgestellt.



Literaturhinweise

Manpages **mount(8)**, **umount(8)**, **fstab(5)**, **mount.nfs(8)** und **nfsconf(8)**

► Angeleitete Übung

Verwalten des NAS mit NFS

Leistungscheckliste

In dieser Übung ändern Sie die Datei **/etc/fstab**, um einen NFS-Export dauerhaft beim Starten bereitzustellen.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Ausführen des Befehls **mount**, um einen NFS-Server zu testen.
- Konfigurieren der NFS-Freigaben in der Konfigurationsdatei **/etc/fstab**, um die Änderungen auch nach einem Neustart des Systems zu speichern.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie den Befehl **lab netstorage-nfs start** auf **workstation** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob die Rechner **servera** und **serverb** im Netzwerk erreichbar ist. Das Skript benachrichtigt Sie, wenn die Server nicht verfügbar sind. Das Startskript konfiguriert **serverb** als einen NFSv4-Server, richtet Berechtigungen ein und exportiert Verzeichnisse. Es erstellt Benutzer und Gruppen, die auf **servera** und **serverb** erforderlich sind.

```
[student@workstation ~]$ lab netstorage-nfs start
```

Eine Reederei verwendet einen zentralen Server, **serverb**, um mehrere freigegebene Dokumente und Verzeichnisse zu hosten. Benutzer auf **servera**, die allesamt Mitglieder der Gruppe **admin** sind, benötigen Zugriff auf die dauerhaft bereitgestellte NFS-Freigabe.

Wichtige Informationen:

- **serverb** gibt das Verzeichnis **/shares/public** frei, in dem einige Textdateien enthalten sind.
 - Mitglieder der Gruppe **admin** (**admin1**, **sysmanager1**) haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/public**.
 - Der wichtigste Bereitstellungspunkt für **servera** ist **/public**.
 - Alle Benutzerpasswörter lauten **redhat**.
- 1. Melden Sie sich bei **servera** als der Benutzer **student** an, und wechseln Sie zum Benutzer **root**.
- 1.1. Melden Sie sich unter **servera** als Benutzer **student** an.

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

- 1.2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- 2. Testen Sie den NFS-Server auf **serverb** mit **servera** als NFS-Client.

- 2.1. Erstellen Sie den Bereitstellungspunkt **/public** auf **servera**.

```
[root@servera ~]# mkdir /public
```

- 2.2. Führen Sie auf **servera** den Befehl **mount** aus, um zu verifizieren, dass die durch **serverb** exportierte NFS-Freigabe **/share/public** ordnungsgemäß auf dem Bereitstellungspunkt **/public** bereitgestellt wird.

```
[root@servera ~]# mount -t nfs \  
serverb.lab.example.com:/shares/public /public
```

- 2.3. Listen Sie den Inhalt der bereitgestellten NFS-Freigabe auf.

```
[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. Untersuchen Sie die **mount**-Befehloptionen für die bereitgestellte NFS-Freigabe.

```
[root@servera ~]# mount | grep public  
serverb.lab.example.com:/shares/public on /public type nfs4  
(rw,relatime,vers=4.2,rsize=262144,wszie=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)
```

- 2.5. Heben Sie die Bereitstellung der NFS-Freigabe auf.

```
[root@servera ~]# umount /public
```

- 3. Konfigurieren Sie **servera**, um sicherzustellen, dass die oben verwendete Freigabe dauerhaft bereitgestellt wird.

- 3.1. Öffnen Sie die Datei **/etc/fstab** zur Bearbeitung.

```
[root@servera ~]# vim /etc/fstab
```

Fügen Sie am Ende der Datei die folgende Zeile hinzu:

```
serverb.lab.example.com:/shares/public /public nfs rw,sync 0 0
```

- 3.2. Führen Sie den Befehl **mount** aus, um das freigegebene Verzeichnis bereitzustellen.

```
[root@servera ~]# mount /public
```

- 3.3. Listen Sie den Inhalt dieses freigegebenen Verzeichnisses auf.

```
[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. Starten Sie den Rechner **servera** neu.

```
[root@servera ~]# systemctl reboot
```

- 4. Melden Sie sich nach Abschluss des **servera**-Bootvorgangs als der Benutzer **admin1** bei **servera** an und testen Sie die dauerhaft bereitgestellte NFS-Freigabe.

- 4.1. Melden Sie sich bei **servera** als der Benutzer **admin1** an.

```
[student@workstation ~]$ ssh admin1@servera
[admin1@servera ~]$
```

- 4.2. Testen Sie die auf **/public** bereitgestellte NFS-Freigabe.

```
[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. Melden Sie sich von **servera** ab.

```
[admin1@servera ~]$ exit  
logout  
Connection to servera closed.
```

Beenden

Führen Sie auf **workstation** das Skript **lab netstorage-nfs finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab netstorage-nfs finish
```

Hiermit ist die angeleitete Übung beendet.

Automatisches Mounten von Network-Attached Storage

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Beschreiben der Vorteile der Verwendung des Automounters
- Automatisches Mounten von NFS-Freigaben anhand von direkten und indirekten Zuordnungen, einschließlich Platzhaltern.

Bereitstellen von NFS-Freigaben mit der automatischen Bereitstellung

Die *automatische Bereitstellung* ist ein Service (**autofs**), mit dem NFS-Freigaben automatisch „auf Abruf“ bereitgestellt werden und die Bereitstellung automatisch wieder aufgehoben wird, wenn die NFS-Freigaben nicht mehr verwendet werden.

Vorteile der automatischen Bereitstellung

- Benutzer benötigen keine root-Berechtigungen, um die Befehle **mount** und **umount** auszuführen.
- Im Automounter konfigurierte NFS-Freigaben stehen allen Benutzern auf dem Rechner vorbehaltlich Zugriffsberechtigungen zur Verfügung.
- NFS-Freigaben sind anders als Einträge in **/etc/fstab** nicht dauerhaft verbunden, wodurch Netzwerk- und Systemressourcen freigesetzt werden.
- Die automatische Bereitstellung ist clientseitig konfiguriert. Es ist keine serverseitige Konfiguration notwendig.
- Die automatische Bereitstellung verwendet dieselben Optionen wie der Befehl **mount**, einschließlich Sicherheitsoptionen.
- Die automatische Bereitstellung unterstützt die direkte und indirekte Zuordnung von Bereitstellungspunkten und sorgt so für Flexibilität an den Speicherorten von Bereitstellungspunkten.
- **autofs** erstellt und entfernt indirekte Bereitstellungspunkte, wodurch sich die manuelle Verwaltung erübrigkt.
- NFS ist das standardmäßige Network File System des Automounters. Andere Network File Systems können jedoch automatisch bereitgestellt werden.
- Der Service **autofs** wird wie andere Systemservices verwaltet.

Erstellen eines Automounters

Die Konfiguration eines Automounters erfolgt in mehreren Schritten:

1. Installieren Sie das Paket **autofs**.

```
[user@host ~]$ sudo yum install autofs
```

Dieses Paket enthält alles, was Sie für die Verwendung des Automounters für NFS-Freigaben benötigen.

2. Fügen Sie **/etc/auto.master.d** eine *Master-Zuordnungsdatei* hinzu. Diese Datei identifiziert das für Bereitstellungspunkte verwendete Basisverzeichnis und die zur Erstellung der Automounter verwendete Zuordnungsdatei.

```
[user@host ~]$ sudo vim /etc/auto.master.d/demo.autofs
```

Der Name der Master-Zuordnungsdatei ist beliebig (wenn auch in der Regel aussagekräftig). Er muss jedoch die Erweiterung **.autofs** aufweisen, damit er vom Subsystem erkannt wird. Sie können mehrere Einträge in einer einzigen Master-Zuordnungsdatei platzieren. Alternativ können Sie mehrere Master-Zuordnungsdateien mit jeweils eigenen Einträgen erstellen, die logisch gruppiert sind.

Fügen Sie den Master-Zuordnungseintrag hinzu, in diesem Fall für indirekt zugeordnete Bereitstellungen:

```
/shares  /etc/auto.demo
```

Dieser Eintrag verwendet das Verzeichnis **/shares** als Grundlage zukünftiger indirekter, automatischer Bereitstellungen. Die Datei **/etc/auto.demo** enthält die Bereitstellungsdetails. Verwenden Sie einen absoluten Dateinamen. Die Datei **auto.demo** muss vor dem Starten des Services **autofs** erstellt werden.

3. Erstellen Sie die Zuordnungsdateien. Jede Zuordnungsdatei identifiziert den Bereitstellungspunkt, Bereitstellungsoptionen und den Quellspeicherort für die Bereitstellung.

```
[user@host ~]$ sudo vim /etc/auto.demo
```

Die Namenskonvention für die Zuordnungsdatei lautet **/etc/auto.name**, wobei *name* für den Inhalt der Zuordnung steht.

```
work  -rw, sync  server:/shares/work
```

Das Format eines Eintrags lautet *Bereitstellungspunkt, Bereitstellungsoptionen und Quellspeicherort*. Dieses Beispiel zeigt einen grundlegenden Eintrag mit einer indirekten Zuordnung. Direkte und indirekte Zuordnungen mit Platzhaltern werden an späterer Stelle dieses Abschnitts behandelt.

- Der *Bereitstellungspunkt*, der in den Manpages als „Schlüssel“ bekannt ist, wird automatisch vom Service **autofs** erstellt und entfernt. In diesem Fall ist der vollqualifizierte Bereitstellungspunkt **/shares/work** (siehe die Master-Zuordnungsdatei). Die Verzeichnisse **/shares** und **/shares/work** werden bei Bedarf vom Service **autofs** erstellt und entfernt.

In diesem Beispiel spiegelt der lokale Bereitstellungspunkt die Verzeichnisstruktur des Servers, dies ist jedoch nicht erforderlich. Der lokale Bereitstellungspunkt kann beliebig benannt werden. Der Service **autofs** erzwingt keine bestimmte Namensstruktur auf dem Client.

- Bereitstellungsoptionen beginnen mit einem „-“ und werden ohne Leerzeichen und getrennt durch ein Komma aufgeführt. Beim automatischen Mounten stehen zum manuellen Mounten eines Dateisystems Mount-Optionen zur Verfügung. In diesem Beispiel stellt die automatische Bereitstellung die Freigabe mit Lese-/Schreibzugriff (Option **rw**) bereit, und der Server wird sofort während der Schreibvorgänge synchronisiert (Option **sync**).

Nützliche Optionen, die spezifisch sind für die automatische Bereitstellung, enthalten - **fstype=** und **-strict**. Verwenden Sie **fstype**, um den Dateisystemtyp anzugeben, beispielsweise **nfs4** oder **xfs**, und verwenden Sie **strict**, um Fehler als schwerwiegend zu behandeln, die während der Bereitstellung von Dateisystemen auftreten.

- Der Quellspeicherort für NFS-Freigaben folgt dem Muster `host:/pathname`, in diesem Beispiel `serverb:/shares/work`. Damit die automatische Bereitstellung erfolgreich ist, muss der NFS-Server **serverb** das Verzeichnis mit Lese-/Schreibzugriff exportieren und der Benutzer, der den Zugriff anfordert, muss über Linux-Standarddateiberechtigungen für das Verzeichnis verfügen. Wenn **serverb** das Verzeichnis nur mit Lesezugriff exportiert, dann erhält der Client nur Lesezugriff, selbst wenn er Lese-/Schreibzugriff angefordert hat.

4. Starten und aktivieren Sie den Service für die automatische Bereitstellung.

Verwenden Sie **systemctl**, um den **autofs**-Dienst zu starten und zu aktivieren.

```
[user@host ~]$ sudo systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /usr/
lib/systemd/system/autofs.service.
```

Direkte Zuordnungen

Direkte Zuordnungen werden verwendet, um eine NFS-Freigabe zum absoluten Pfad eines vorhandenen Bereitstellungspunkts zuzuordnen.

Um direkt zugeordnete Bereitstellungspunkte zu verwenden, kann die Master-Zuordnungsdatei folgendermaßen aussehen:

```
/- /etc/auto.direct
```

Alle direkten Zuordnungseinträge verwenden `/-` als Basisverzeichnis. In diesem Fall enthält die Zuordnungsdatei **/etc/auto.direct** die Bereitstellungsdetails.

Der Inhalt der Datei **/etc/auto.direct** wird möglicherweise wie folgt angezeigt:

```
/mnt/docs -rw, sync serverb:/shares/docs
```

Der Bereitstellungspunkt (oder Schlüssel) ist immer ein absoluter Pfad. Der Rest der Zuordnungsdatei verwendet dieselbe Struktur.

In diesem Beispiel ist das Verzeichnis **/mnt** vorhanden und wird nicht durch **autofs** verwaltet. Das vollständige Verzeichnis **/mnt/docs** wird vom Service **autofs** automatisch erstellt und entfernt.

Indirekte Platzhalterzuordnungen

Wenn ein NFS-Server mehrere Unterverzeichnisse in einem Verzeichnis exportiert, kann die automatische Bereitstellung so konfiguriert werden, dass über einen einzigen Zuordnungseintrag auf alle diese Unterverzeichnisse zugegriffen wird.

Wenden wir uns wieder dem vorherigen Beispiel zu. Wenn `serverb:/shares` zwei oder mehr Unterverzeichnisse exportiert, auf die über dieselben Bereitstellungsoptionen zugegriffen werden kann, könnten die Inhalte der Datei `/etc/auto.demo` folgendermaßen aussehen:

```
* -rw, sync serverb:/shares/&
```

Der Bereitstellungspunkt (oder Schlüssel) ist ein Sternchen (*), und das Unterverzeichnis am Quellspeicherort ist ein Kaufmanns-Und (&). Der übrige Eintrag bleibt gleich.

Versucht ein Benutzer, auf `/shares/work` zuzugreifen, ersetzt der Schlüssel * (in diesem Beispiel **work**) das Kaufmanns-Und am Quellspeicherort, und `serverb:/shares/work` wird bereitgestellt. Wie beim Beispiel zur indirekten Zuordnung wird das Verzeichnis **work** automatisch vom Service **autofs** erstellt und entfernt.



Literaturhinweise

Manpages **autofs(8)**, **automount(5)**, **auto.master(5)** und **mount.nfs(8)**

► Angeleitete Übung

Automatisches Mounten von Network-Attached Storage

Leistungscheckliste

In dieser Übung erstellen Sie direkt zugeordnete und indirekt zugeordnete über das automatische Mounten verwaltete Bereitstellungspunkte, die NFS-Dateisysteme mounten.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Installieren der für die automatische Bereitstellung erforderlichen Pakete.
- Konfigurieren direkter und indirekter Zuordnungen für die automatische Bereitstellung, um Ressourcen von einem vorkonfigurierten NFSv4-Server abzurufen.
- Verstehen des Unterschieds zwischen direkten und indirekten Zuordnungen für die automatische Bereitstellung.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie den Befehl **lab netstorage-autofs start** auf **workstation** aus. Dieses Startskript legt fest, ob über das Netzwerk auf **servera** und **serverb** zugegriffen werden kann. Das Skript benachrichtigt Sie, wenn die Server nicht verfügbar sind. Das Startskript konfiguriert **serverb** als einen NFSv4-Server, richtet Berechtigungen ein und exportiert Verzeichnisse. Darüber hinaus erstellt es Benutzer und Gruppen, die auf **servera** und **serverb** erforderlich sind.

```
[student@workstation ~]$ lab netstorage-autofs start
```

Ein Internet Service Provider verwendet einen zentralen Server, **serverb**, um gemeinsam genutzte Verzeichnisse mit wichtigen Dokumenten zu hosten, die bei Bedarf verfügbar sein müssen. Wenn sich Benutzer bei **servera** anmelden, benötigen sie Zugriff auf die automatisch bereitgestellten freigegebenen Verzeichnisse.

Wichtige Informationen:

- **serverb** exportiert das Verzeichnis **/shares/indirect** als eine NFS-Freigabe, das wiederum die Unterverzeichnisse **west**, **central** und **east** enthält.
- Darüber hinaus exportiert **serverb** das Verzeichnis **/shares/direct/external** als eine NFS-Freigabe.
- Die Gruppe **operators** besteht aus den Benutzern **operator1** und **operator2**. Sie haben Lese- und Schreibzugriff auf die gemeinsam genutzten Verzeichnisse **/shares/indirect/west**, **/shares/indirect/central** und **/shares/indirect/east**.

- Die Gruppe **contractors** besteht aus den Benutzern **contractor1** und **contractor2**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/direct/external**.
- Die erwarteten Bereitstellungspunkte für **servera** sind **/external** und **/internal**.
- Das freigegebene Verzeichnis **/shares/direct/external** sollte mithilfe der *direkten* Zuordnung auf **/external** automatisch auf **servera** bereitgestellt werden.
- Das freigegebene Verzeichnis **/shares/indirect/west** sollte mithilfe einer *indirekten* Zuordnung auf **/internal/west** automatisch auf **servera** bereitgestellt werden.
- Das freigegebene Verzeichnis **/shares/indirect/central** sollte mithilfe einer *indirekten* Zuordnung auf **/internal/central** automatisch auf **servera** bereitgestellt werden.
- Das freigegebene Verzeichnis **/shares/indirect/east** sollte mithilfe einer *indirekten* Zuordnung auf **/internal/east** automatisch auf **servera** bereitgestellt werden.
- Alle Benutzerpasswörter lauten **redhat**.

► 1. Melden Sie sich bei **servera** an, und installieren Sie die erforderlichen Pakete.

- Melden Sie sich unter **servera** als Benutzer **student** an.

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

- Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- Installieren Sie das Paket **autofs**.

```
[root@servera ~]# yum install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
```

► 2. Konfigurieren Sie mit Freigaben von **serverb** eine direkte Zuordnung für die automatische Bereitstellung auf **servera**. Erstellen Sie die direkte Zuordnung mit den Dateien namens **/etc/auto.master.d/direct.autofs** für die Master-Zuordnung und **/etc/auto.direct** für die Zuordnungsdatei. Verwenden Sie das Verzeichnis **/external** als Hauptbereitstellungspunkt auf **servera**.

- Testen Sie den NFS-Server und die -Freigabe, bevor Sie die Konfiguration für die automatische Bereitstellung fortsetzen.

```
[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. Erstellen Sie eine Master-Zuordnungsdatei namens **/etc/auto.master.d/direct.autofs**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
/- /etc/auto.direct
```

- 2.3. Erstellen Sie eine direkte Zuordnungsdatei namens **/etc/auto.direct**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
/external -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/direct/external
```

- 3. Konfigurieren Sie mit Freigaben von **serverb** eine indirekte Automounter-Zuordnung auf **servera**. Erstellen Sie die indirekte Zuordnung mit den Dateien namens **/etc/auto.master.d/indirect.autofs** für die Master-Zuordnung und **/etc/auto.indirect** für die Zuordnungsdatei. Verwenden Sie das Verzeichnis **/internal** als Hauptbereitstellungspunkt auf **servera**.

- 3.1. Testen Sie den NFS-Server und die -Freigabe, bevor Sie die Konfiguration für die automatische Bereitstellung fortsetzen.

```
[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. Erstellen Sie eine Master-Zuordnungsdatei namens **/etc/auto.master.d/indirect.autofs**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
/internal /etc/auto.indirect
```

- 3.3. Erstellen Sie eine indirekte Zuordnungsdatei namens **/etc/auto.indirect**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
* -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/indirect/&
```

- 4. Starten Sie den Service **autofs** auf **servera**, und aktivieren Sie ihn so, dass er automatisch zur Startzeit gestartet wird. Starten Sie **servera** neu, um zu bestimmen, ob der Service **autofs** automatisch gestartet wird.

- 4.1. Starten und aktivieren Sie den Service **autofs** auf **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. Starten Sie den Rechner **servera** neu.

```
[root@servera ~]# systemctl reboot
```

- 5. Testen Sie die direkte Zuordnung für die automatische Bereitstellung als der Benutzer **contractor1**. Wenn Sie fertig sind, beenden Sie die Benutzersitzung **contractor1** auf **servera**.

- 5.1. Nachdem der Rechner **servera** gebootet wurde, melden Sie sich bei **servera** als Benutzer **student** an.

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

- 5.2. Wechseln Sie zum Benutzer **contractor1**.

```
[student@servera ~]$ su - contractor1  
Password: redhat
```

- 5.3. Listen Sie den Bereitstellungspunkt **/external** auf.

```
[contractor1@servera ~]$ ls -l /external  
total 4  
-rw-r--r--. 1 root contractors 22 Apr 7 23:34 README.txt
```

- 5.4. Überprüfen Sie den Inhalt und testen Sie den Zugriff auf den Bereitstellungspunkt **/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. Beenden Sie die Benutzersitzung **contractor1**.

```
[contractor1@servera ~]$ exit  
logout  
[student@servera ~]$
```

- 6. Testen Sie die indirekte Zuordnung für die automatische Bereitstellung als der Benutzer **operator1**. Melden Sie sich von **servera** ab, wenn Sie fertig sind.

- 6.1. Wechseln Sie zum Benutzer **operator1**.

```
[student@servera ~]$ su - operator1  
Password: redhat
```

- 6.2. Listen Sie den Bereitstellungspunkt **/internal** auf.

```
[operator1@servera ~]$ ls -l /internal  
total 0
```



Anmerkung

Sie werden feststellen, dass in einer indirekten Zuordnung für die automatische Bereitstellung, selbst wenn Sie sich im zugeordneten Bereitstellungspunkt befinden, Sie jedes der gemeinsam genutzten Unterverzeichnisse oder Dateien bei Bedarf aufrufen müssen, um auf sie zuzugreifen. In einer direkten Zuordnung für die automatische Bereitstellung erhalten Sie nach dem Öffnen des zugeordneten Bereitstellungspunkts Zugriff auf die Verzeichnisse und Inhalte, die im freigegebenen Verzeichnis konfiguriert sind.

- 6.3. Testen Sie den Zugriff auf freigegebene Verzeichnisse mit dem Befehl für die automatische Bereitstellung **/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. Testen Sie den Zugriff auf freigegebene Verzeichnisse mit dem Befehl für die automatische Bereitstellung **/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. Testen Sie den Zugriff auf freigegebene Verzeichnisse mit dem Befehl für die automatische Bereitstellung **/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. Testen Sie den Zugriff auf freigegebene Verzeichnisse mit dem Befehl für die automatische Bereitstellung **/external**.

```
[operator1@servera ~]$ ls -l /external
ls: cannot open directory '/external': Permission denied
```

- 6.7. Melden Sie sich von **servera** ab.

```
[operator1@servera ~]$ exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
```

Beenden

Führen Sie auf **workstation** das Skript **lab netstorage-autofs finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab netstorage-autofs finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Zugreifen auf Network-Attached Storage

Leistungscheckliste

In dieser Übung richten Sie die automatische Bereitstellung mit einer indirekten Zuordnung ein, wobei Freigaben von einem NFSv4-Server verwendet werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Installieren der zum Einrichten des Automounters erforderlichen Pakete.
- Konfigurieren einer indirekten Automounter-Zuordnung, um Ressourcen von einem vorkonfigurierten NFSv4-Server abzurufen.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie den Befehl **lab netstorage-review start** auf **workstation** aus. Dieses Startskript legt fest, ob über das Netzwerk auf die Systeme **servera** und **serverb** zugegriffen werden kann. Das Startskript konfiguriert **serverb** als einen NFSv4-Server, richtet Berechtigungen ein und exportiert Verzeichnisse. Darüber hinaus erstellt es Benutzer und Gruppen, die auf den Systemen **servera** und **serverb** erforderlich sind.

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

Ein IT-Support-Unternehmen verwendet den zentralen Server **serverb**, um einige freigegebene Verzeichnisse auf **/remote/shares** für ihre Gruppen und Benutzer zu hosten. Benutzer müssen sich anmelden können und ihre freigegebenen Verzeichnisse bei Bedarf im Verzeichnis **/shares** auf **servera** bereitgestellt und einsatzbereit haben.

Wichtige Informationen:

- **serverb** gibt das Verzeichnis **/shares** frei, das wiederum die Unterverzeichnisse **management**, **production** und **operation** enthält.
- Die Gruppe **managers** besteht aus den Benutzern **manager1** und **manager2**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/management**.
- Die Gruppe **production** besteht aus den Benutzern **dbuser1** und **sysadmin1**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/production**.
- Die Gruppe **operators** besteht aus den Benutzern **contractor1** und **consultant1**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/operation**.
- Der Hauptbereitstellungspunkt für **servera** ist das Verzeichnis **/remote**.
- Das freigegebene Verzeichnis **/shares/management** sollte in **/remote/management** auf **servera** automatisch bereitgestellt werden.

Kapitel 9 | Zugreifen auf Network-Attached Storage

- Das freigegebene Verzeichnis **/shares/production** sollte in **/remote/production** auf **servera** automatisch bereitgestellt werden.
 - Das freigegebene Verzeichnis **/shares/operation** sollte in **/remote/operation** auf **servera** automatisch bereitgestellt werden.
 - Alle Benutzerpasswörter lauten **redhat**.
- Melden Sie sich bei **servera** an, und installieren Sie die erforderlichen Pakete.
 - Konfigurieren Sie mit Freigaben von **serverb** eine indirekte Automounter-Zuordnung auf **servera**. Erstellen Sie eine indirekte Zuordnung mit den Dateien namens **/etc/auto.master.d/shares.autofs** für die Master-Zuordnung und **/etc/auto.shares** für die Zuordnungsdatei. Verwenden Sie das Verzeichnis **/remote** als Hauptbereitstellungspunkt auf **servera**. Starten Sie **servera** neu, um zu bestimmen, ob der Service **autofs** automatisch gestartet wird.
 - Testen Sie die Konfiguration **autofs** mit den verschiedenen Benutzern. Melden Sie sich von **servera** ab, wenn Sie fertig sind.

Bewertung

Führen Sie auf **workstation** den Befehl **lab netstorage-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** den Befehl **lab netstorage-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Zugreifen auf Network-Attached Storage

Leistungscheckliste

In dieser Übung richten Sie die automatische Bereitstellung mit einer indirekten Zuordnung ein, wobei Freigaben von einem NFSv4-Server verwendet werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Installieren der zum Einrichten des Automounters erforderlichen Pakete.
- Konfigurieren einer indirekten Automounter-Zuordnung, um Ressourcen von einem vorkonfigurierten NFSv4-Server abzurufen.

Bevor Sie Beginnen

Melden Sie sich bei **workstation** als **student** mit dem Passwort **student** an.

Führen Sie den Befehl **lab netstorage-review start** auf **workstation** aus. Dieses Startskript legt fest, ob über das Netzwerk auf die Systeme **servera** und **serverb** zugegriffen werden kann. Das Startskript konfiguriert **serverb** als einen NFSv4-Server, richtet Berechtigungen ein und exportiert Verzeichnisse. Darüber hinaus erstellt es Benutzer und Gruppen, die auf den Systemen **servera** und **serverb** erforderlich sind.

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

Ein IT-Support-Unternehmen verwendet den zentralen Server **serverb**, um einige freigegebene Verzeichnisse auf **/remote/shares** für ihre Gruppen und Benutzer zu hosten. Benutzer müssen sich anmelden können und ihre freigegebenen Verzeichnisse bei Bedarf im Verzeichnis **/shares** auf **servera** bereitgestellt und einsatzbereit haben.

Wichtige Informationen:

- **serverb** gibt das Verzeichnis **/shares** frei, das wiederum die Unterverzeichnisse **management**, **production** und **operation** enthält.
- Die Gruppe **managers** besteht aus den Benutzern **manager1** und **manager2**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/management**.
- Die Gruppe **production** besteht aus den Benutzern **dbuser1** und **sysadmin1**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/production**.
- Die Gruppe **operators** besteht aus den Benutzern **contractor1** und **consultant1**. Sie haben Lese- und Schreibzugriff auf das freigegebene Verzeichnis **/shares/operation**.
- Der Hauptbereitstellungspunkt für **servera** ist das Verzeichnis **/remote**.
- Das freigegebene Verzeichnis **/shares/management** sollte in **/remote/management** auf **servera** automatisch bereitgestellt werden.

Kapitel 9 | Zugreifen auf Network-Attached Storage

- Das freigegebene Verzeichnis **/shares/production** sollte in **/remote/production** auf **servera** automatisch bereitgestellt werden.
- Das freigegebene Verzeichnis **/shares/operation** sollte in **/remote/operation** auf **servera** automatisch bereitgestellt werden.
- Alle Benutzerpasswörter lauten **redhat**.

1. Melden Sie sich bei **servera** an, und installieren Sie die erforderlichen Pakete.

1.1. Melden Sie sich unter **servera** als Benutzer **student** an.

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

1.2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

1.3. Installieren Sie das Paket **autofs**.

```
[root@servera ~]# yum install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
```

2. Konfigurieren Sie mit Freigaben von **serverb** eine indirekte Automounter-Zuordnung auf **servera**. Erstellen Sie eine indirekte Zuordnung mit den Dateien namens **/etc/auto.master.d/shares.autofs** für die Master-Zuordnung und **/etc/auto.shares** für die Zuordnungsdatei. Verwenden Sie das Verzeichnis **/remote** als Hauptbereitstellungspunkt auf **servera**. Starten Sie **servera** neu, um zu bestimmen, ob der Service **autofs** automatisch gestartet wird.

2.1. Testen Sie den NFS-Server, bevor Sie die Konfiguration für die automatische Bereitstellung fortsetzen.

```
[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. Erstellen Sie eine Master-Zuordnungsdatei namens **/etc/auto.master.d/shares.autofs**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
[root@servera ~]# vim /etc/auto.master.d/shares.autofs  
/remote /etc/auto.shares
```

- 2.3. Erstellen Sie eine indirekte Zuordnungsdatei namens **/etc/auto.shares**, fügen Sie den folgenden Inhalt ein, und speichern Sie die Änderungen.

```
[root@servera ~]# vim /etc/auto.shares  
* -rw, sync, fstype=nfs4 serverb.lab.example.com:/shares/&
```

- 2.4. Starten und aktivieren Sie den Service **autofs** auf **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. Starten Sie den Rechner **servera** neu.

```
[root@servera ~]# systemctl reboot
```

3. Testen Sie die Konfiguration **autofs** mit den verschiedenen Benutzern. Melden Sie sich von **servera** ab, wenn Sie fertig sind.

- 3.1. Nachdem der Rechner **servera** gebootet wurde, melden Sie sich bei **servera** als Benutzer **student** an.

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

- 3.2. Führen Sie den Befehl **su - manager1** aus, um zum Benutzer **manager1** zu wechseln und den Zugriff zu testen.

```
[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. Wechseln Sie zum Benutzer **dbuser1** und testen Sie den Zugriff.

```
[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. Wechseln Sie zum Benutzer **contractor1**, und testen Sie den Zugriff.

```
[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. Untersuchen Sie die **mount**-Optionen für die automatisch bereitgestellte NFS-Freigabe.

```
[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,rsize=262144,wszie=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,rszie=262144,wszie=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,rsize=262144,wszie=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. Melden Sie sich von **servera** ab.

```
[student@servera ~]$ exit  
logout  
[student@workstation ~]$
```

Bewertung

Führen Sie auf **workstation** den Befehl **lab netstorage-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** den Befehl **lab netstorage-review finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Ein- und Aushängen eines NFS-Exports über die Befehlszeile.
- Konfigurieren Sie einen NFS-Export so, dass er beim Start automatisch bereitgestellt wird.
- Konfigurieren Sie den Automounter mit direkten und indirekten Karten und beschreiben Sie deren Unterschiede.

Kapitel 10

Steuern des Boot-Vorgangs

Ziel

Verwalten Sie den Boot-Vorgang, um die angebotenen Services zu steuern und Probleme zu beheben.

Ziele

- Beschreiben des Red Hat Enterprise Linux-Boot-Vorgangs, Festlegen des beim Booten verwendeten Standardziels und Booten eines Systems zu einem nicht standardmäßigen Ziel.
- Anmelden beim System und Ändern des Root-Passworts, wenn das aktuelle Root-Passwort verloren gegangen ist.
- Manuelles Reparieren der Dateisystemkonfiguration oder bei Beschädigungsproblemen, die den Boot-Vorgang stoppen.

Abschnitte

- Auswahl des Boot-Ziels (und angeleitete Übung)
- Zurücksetzen des Root-Passworts (und angeleitete Übung)
- Beheben von Dateisystemproblemen während des Boot-Vorgangs (und angeleitete Übung)

Praktische Übung

Steuern des Boot-Vorgangs

Auswahl des Boot-Ziels

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Beschreiben des Red Hat Enterprise Linux-Boot-Vorgangs.
- Festlegen des beim Booten verwendeten Standardziels.
- Booten eines Systems zu einem nicht standardmäßigen Ziel.

Beschreiben des Red Hat Enterprise Linux 8-Boot-Vorgangs

Bei modernen Computersystemen handelt es sich um komplexe Kombinationen aus Hardware und Software. Um ein System von einem undefinierten, ausgeschalteten Zustand in Betriebszustand mit einer Eingabeaufforderung zur Anmeldung zu überführen, sind zahlreiche Hard- und Softwarekomponenten erforderlich, die zusammenarbeiten. Die folgende Liste bietet einen umfassenden Überblick über die Aufgaben, die durchgeführt werden, wenn ein physisches **x86_64**-System Red Hat Enterprise Linux 8 bootet. Die Liste für virtuelle **x86_64**-Rechner ist ähnlich, jedoch werden einige der hardwarespezifischen Schritte von der Software bzw. vom Hypervisor ausgeführt.

- Das System wird eingeschaltet. Die Systemfirmware, entweder ein modernes UEFI oder ein älteres BIOS, führt einen *Power On Self Test* (POST) durch und beginnt mit der Initialisierung von einigen der Hardware-Komponenten.

Konfiguriert mit dem BIOS- oder UEFI-Konfigurationsbildschirm des Systems, auf die Sie in der Regel frühzeitig während des Boot-Vorgangs über die Eingabe einer bestimmten Tastenkombination zugreifen, beispielsweise **F2**.

- Die System-Firmware sucht nach einem bootfähigen Gerät, das in der UEFI-Boot-Firmware konfiguriert ist, oder es durchsucht alle Festplatten nach einem *Master Boot Record* (MBR) in der im BIOS konfigurierten Reihenfolge.

Konfiguriert mit dem BIOS- oder UEFI-Konfigurationsbildschirm des Systems, auf die Sie in der Regel frühzeitig während des Boot-Vorgangs über die Eingabe einer bestimmten Tastenkombination zugreifen, beispielsweise **F2**.

- Die Systemfirmware liest einen Bootloader von einer Disk und übergibt die Kontrolle über das System anschließend an den Bootloader. Auf einem Red Hat Enterprise Linux 8-System ist die *Grand Unified Bootloader Version 2 (GRUB2)* der Bootloader.

Konfiguriert mit dem Befehl **grub2-install**, der GRUB2 als Bootloader auf der Disk installiert.

- GRUB2 lädt seine Konfiguration aus der Datei **/boot/grub2/grub.cfg** und zeigt ein Menü an, aus dem Sie den zu bootenden Kernel auswählen können.

Konfiguriert mit dem Verzeichnis **/etc/grub.d/**, der Datei **/etc/default/grub** und dem Befehl **grub2-mkconfig** zum Generieren der Datei **/boot/grub2/grub.cfg**.

- Nachdem Sie einen Kernel ausgewählt haben oder der Timeout abgelaufen ist, lädt der Bootloader den Kernel und **initramfs** von der Disk und platziert sie im Arbeitsspeicher. Ein **initramfs** ist ein Archiv, das die Kernelmodule für die gesamte beim Boot erforderliche Hardware, die Initialisierungsskripts und mehr enthält. Auf Red Hat Enterprise Linux 8 enthält das **initramfs** ein gesamtes verwendbares System.

Konfiguriert mit dem Verzeichnis **/etc/dracut.conf.d/**, dem Befehl **dracut** und dem Befehl **lsinitrd** zum Untersuchen der Datei **initramfs**.

- Der Bootloader übergibt die Kontrolle an den Kernel. Optionen, die in der Kernelbefehlszeile angegeben sind, werden an den Bootloader und den Speicherort des **initramfs** im Speicher übergeben.

Konfiguriert mit dem Verzeichnis **/etc/grub.d/**, der Datei **/etc/default/grub** und dem Befehl **grub2-mkconfig** zum Generieren der Datei **/boot/grub2/grub.cfg**.

- Der Kernel initialisiert sämtliche Hardware, für die er einen Treiber im **initramfs** finden kann. Anschließend führt er **/sbin/init** vom **initramfs** als PID 1 aus. Auf Red Hat Enterprise Linux 8 fungiert **/sbin/init** als ein Link zu **systemd**.

Konfiguriert mit dem Befehlszeilenparameter **init=** des Kernels.

- Die Instanz **systemd** vom **initramfs** führt alle Units für das Ziel **initrd.target** aus. Dies beinhaltet auch das Bereitstellen des auf der Disk befindlichen Root-Dateisystems im Verzeichnis **/sysroot**.

Konfiguriert mit **/etc/fstab**.

- Der Kernel wechselt (schwenkt um) das Root-Dateisystem von **initramfs** in das Root-Dateisystem in **/sysroot**. Anschließend führt **systemd** sich selbst mit der Kopie von **systemd** aus, die auf der Disk installiert ist.
- **Systemd** sucht nach einem Standardziel, das entweder über die Kernelbefehlszeile übertragen wird oder auf dem System konfiguriert ist, und startet (bzw. stoppt) Units anschließend (und stoppt sie wieder), um der Konfiguration für das Ziel zu entsprechen. Auf diese Weise werden Abhängigkeiten zwischen Units automatisch behoben. Im Wesentlichen ist ein **Systemd**-Ziel ein Satz von Units, die das System aktivieren sollte, um den gewünschten Zustand zu erreichen. Diese Ziele starten in der Regel eine textbasierte Anmeldung oder einen grafischen Anmeldebildschirm.

Konfiguriert mit **/etc/systemd/system/default.target** und **/etc/systemd/system/**.

Rebooten und Herunterfahren

Zum Ausschalten oder für den Reboot eines aktiven Systems können Sie den Befehl **systemctl** ausführen.

systemctl poweroff hält alle aktiven Services an, hebt die Bereitstellung sämtlicher Dateisysteme auf (oder stellt sie schreibgeschützt erneut bereit, wenn die Aufhebung ihrer Bereitstellung nicht möglich ist), und fährt dann das System herunter.

systemctl reboot hält alle aktiven Services an, hebt die Bereitstellung sämtlicher Dateisysteme auf und rebootet dann das System.

Sie können auch die kürzere Version der Befehle **poweroff** und **reboot** verwenden, die symbolische Links zu ihren **systemctl**-Entsprechungen sind.

**Anmerkung**

Zum Anhalten des Systems sind außerdem **systemctl halt** und **halt** verfügbar. Im Gegensatz **poweroff** sorgen diese Befehle jedoch nicht dafür, dass das System ausgeschaltet wird. Sie überführen das System ausschließlich in einen Zustand, von dem aus es sicher manuell ausgeschaltet werden kann.

Auswählen eines Systemd-Ziels

Bei einem **Systemd**-Ziel handelt es sich um einen Satz von **Systemd**-Units, die das System starten sollte, um einen gewünschten Zustand zu erreichen. In der folgenden Tabelle werden die wichtigsten Ziele aufgelistet.

Häufig verwendete Ziele

Ziel	Zweck
graphical.target	Das System unterstützt mehrere Benutzer sowie grafische und textbasierte Anmeldungen.
multi-user.target	Das System unterstützt mehrere Benutzer und ausschließlich Text-basierte Anmeldung.
rescue.target	sulogin -Eingabeaufforderung, grundlegende Systeminitialisierung ist abgeschlossen.
emergency.target	sulogin -Eingabeaufforderung; initramfs -Pivot abgeschlossen und System-Root schreibgeschützt auf / bereitgestellt.

Ein Ziel kann ein Teil eines anderen Ziels sein. Beispielsweise ist **multi-user.target** in **graphical.target** enthalten, wobei Ersteres von **basic.target** und anderen Zielen abhängt. Sie können diese Abhängigkeiten mit dem folgenden Befehl anzeigen.

```
[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
*   | | ├─local-fs.target
*   | | └─swap.target
...output omitted...
```

Verwenden Sie den folgenden Befehl, um die verfügbaren Ziele aufzulisten.

```
[user@host ~]$ systemctl list-units --type=target --all
UNIT                  LOAD     ACTIVE    SUB      DESCRIPTION
-----                -----   -----   -----
basic.target          loaded   active   active  Basic System
```

```
cryptsetup.target      loaded  active  active Local Encrypted Volumes
emergency.target     loaded  inactive dead   Emergency Mode
getty-pre.target     loaded  inactive dead   Login Prompts (Pre)
getty.target          loaded  active   active Login Prompts
graphical.target     loaded  inactive dead   Graphical Interface
...output omitted...
```

Auswahl eines Ziels während zur Laufzeit

Während ein System ausgeführt wird, können Administratoren über den Befehl **systemctl isolate** zu einem anderen Ziel wechseln.

```
[root@host ~]# systemctl isolate multi-user.target
```

Die Isolierung eines Ziels hat zur Folge, dass alle Services, die das Ziel (und seine Abhängigkeiten) nicht benötigt, gestoppt werden. Erforderliche Services, die noch nicht ausgeführt werden, werden außerdem gestartet.

Nicht alle Ziele können isoliert werden. Sie können nur Ziele isolieren, bei denen **AllowIsolate=yes** in ihren Unit-Files festgelegt ist. Sie können beispielsweise das grafische Ziel isolieren, nicht jedoch das Cryptsetup-Ziel.

```
[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)
```

Festlegen eines Standardziels

Wenn das System startet, aktiviert **systemd** das Ziel **default.target**. Normalerweise ist das Standardziel in **/etc/systemd/system/** ein symbolischer Link zu **graphical.target** oder **multi-user.target**. Anstatt diesen symbolischen Link manuell zu bearbeiten, können Sie den Befehl **systemctl** verwenden, der zwei Sub-Befehle zum Verwalten dieses Links bereitstellt: **get-default** und **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
```

Auswahl eines anderen Ziels während zur Boot-Zeit

Um während des Bootens ein anderes Ziel auszuwählen, hängen Sie die Option **systemd.unit=target.target** an die Kernelbefehlszeile aus dem Bootloader an.

Um das System beispielsweise in einer Rettungs-Shell zu booten, in der Sie die Systemkonfiguration ändern können, während fast keine Services ausgeführt werden, fügen Sie im Bootloader der Kernelbefehlszeile die folgende Option an.

```
systemd.unit=rescue.target
```

Diese Konfigurationsänderung bezieht sich nur auf einen einzelnen Boot-Vorgang, sodass Sie über ein nützliches Tool für die Fehlerbehebung beim Boot-Vorgang verfügen.

Um dieses Verfahren zum Auswählen eines anderen Ziels anzuwenden, befolgen Sie folgende Schritte:

1. Booten oder rebooten Sie das System.
2. Unterbrechen Sie den Zählvorgang des Bootloader-Menüs durch das Drücken einer beliebigen Taste (mit Ausnahme der **Eingabetaste**, durch die ein normaler Boot initiiert werden würde).
3. Bewegen Sie den Cursor zum Kerneleintrag, der gestartet werden soll.
4. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
5. Bewegen Sie den Cursor zu der Zeile, die mit **linux** beginnt. Dies ist die Kernelbefehlszeile.
6. Fügen Sie **systemd.unit=target.target** an. Beispielsweise **systemd.unit=emergency.target**.
7. Drücken Sie **Strg+x**, um mit diesen Änderungen zu booten.



Literaturhinweise

info grub2 (*Handbuch zu GNU GRUB*)

Manpages **bootup(7)**, **dracut .bootup(7)**, **lsinitrd(1)**, **systemd.target(5)**, **systemd.special(7)**, **sulogin(8)** und **systemctl(1)**

Weitere Informationen finden Sie im Kapitel *Managing services with systemd* im *Configuring basic system settings* Guide unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_basic_system_settings/#managing-services-with-systemd

► Angeleitete Übung

Auswahl des Boot-Ziels

In dieser Übung bestimmen Sie das Standardziel, in dem ein System gebootet wird, und booten dieses System in anderen Zielen.

Ergebnisse

Sie sollten das Standardziel des Systems aktualisieren und ein temporäres Ziel vom Bootloader aus verwenden können.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab boot-selecting start** aus. Dieser Befehl führt ein Startskript aus, das **workstation** für die Übung vorbereitet.

```
[student@workstation ~]$ lab boot-selecting start
```

- 1. Öffnen Sie auf **workstation** ein Terminal, und bestätigen Sie, dass das Standardziel **graphical.target** lautet.

```
[student@workstation ~]$ systemctl get-default  
graphical.target
```

- 2. Wechseln Sie auf **workstation** manuell zum Ziel **multi-user**, ohne einen Reboot durchzuführen. Führen Sie den Befehl **sudo** aus. Verwenden Sie bei Aufforderung **student** als Passwort.

```
[student@workstation ~]$ sudo systemctl isolate multi-user.target  
[sudo] password for student: student
```

- 3. Greifen Sie auf die textbasierte Konsole zu. Verwenden Sie die Tastenfolge **Strg+Alt+F1** mit der entsprechenden Taste oder dem entsprechenden Menüeintrag. Melden Sie sich als **root** mit dem Passwort **redhat** an.



Anmerkung

Erinnerung: Wenn Sie das Terminal über eine Webseite verwenden, können Sie auf das Symbol „Show Keyboard“ unter der URL-Leiste Ihres Webbrowsers und dann rechts neben der IP-Adresse des Geräts klicken.

```
workstation login: root  
Password: redhat  
[root@workstation ~]#
```

Kapitel 10 | Steuern des Boot-Vorgangs

- 4. Konfigurieren Sie **workstation** so, dass er automatisch im Ziel **multi-user** gestartet wird, und rebooten Sie anschließend **workstation**, um die Änderung zu verifizieren. Wenn Sie fertig sind, setzen Sie das Standardziel **systemd** auf das **grafische** Ziel zurück.
- 4.1. Führen Sie den Befehl **systemctl set-default** aus, um das Standardziel festzulegen.

```
[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. Rebooten Sie **workstation**.

```
[root@workstation ~]# systemctl reboot
```

Beachten Sie, dass das System nach dem Neustart eine textbasierte Konsole und keine grafische Anmeldung mehr anzeigt.

- 4.3. Melden Sie sich als **root** mit dem Passwort **redhat** an.

```
workstation login: root
Password: redhat
Last login: Thu Mar 28 14:50:53 on tty1
[root@workstation ~]#
```

- 4.4. Setzen Sie das Standardziel **systemd** auf das **grafische** Ziel zurück.

```
[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.
```

Damit ist der erste Teil der Übung abgeschlossen, in der Sie das Einstellen des Standardziels **systemd** üben.

- 5. In diesem zweiten Teil der Übung üben Sie den Rettungsmodus zur Wiederherstellung des Systems.

Greifen Sie auf den Bootloader zu, indem Sie **workstation** rebooten. Booten Sie im Bootloader-Menü das **Rettungsziel**.

- 5.1. Initiiieren Sie den Reboot.

```
[root@workstation ~]# systemctl reboot
```

- 5.2. Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste, um den Zählvorgang zu unterbrechen (mit Ausnahme der **Eingabetaste**, die einen normalen Boot initiiieren würde).
- 5.3. Markieren Sie mit den Cursor-Tasten den standardmäßigen Bootloader-Eintrag.
- 5.4. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
- 5.5. Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt.

- 5.6. Drücken Sie auf **Ende**, um den Cursor zum Ende der Zeile zu bewegen.
- 5.7. Fügen Sie **systemd.unit=rescue.target** an das Ende der Zeile an.
- 5.8. Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.
- 5.9. Melden Sie sich beim Rettungsmodus an. Das **root**-Passwort lautet **redhat**. Möglicherweise müssen Sie die Eingabetaste drücken, um eine leere Eingabeaufforderung zu erhalten.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@workstation ~]#
```

- 6. Überprüfen Sie im Rettungsmodus, ob sich das Root-Dateisystem im Lese-/Schreibmodus befindet.

```
[root@workstation ~]# mount  
...output omitted...  
/dev/vda3 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)  
...output omitted...
```

- 7. Drücken Sie **Strg+d**, um den Boot-Vorgang fortzusetzen.

Das System zeigt eine grafische Anmeldung an. Melden Sie sich als der Benutzer **student** mit dem Passwort **student** an.

Beenden

Führen Sie auf **workstation** das Skript **lab boot-selecting finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab boot-selecting finish
```

Hiermit ist die angeleitete Übung beendet.

Zurücksetzen des Root-Passworts

Ziele

In diesem Abschnitt wird beschrieben, wie Sie sich bei einem System anmelden und das **Root**-Passwort ändern, wenn das aktuelle **Root**-Passwort verloren gegangen ist.

Zurücksetzen des Root-Passworts vom Bootloader

Eine Aufgabe, die jeder Systemadministrator durchführen können sollte, ist das Zurücksetzen eines vergessenen **Root**-Passworts. Wenn der Administrator entweder noch als unprivilegierter Benutzer mit vollständigem **sudo**-Zugriff oder als **Root**-Benutzer angemeldet ist, ist diese Aufgabe hinfällig. Ist der Administrator nicht angemeldet, ist sie jedoch von größerer Bedeutung.

Zum Festlegen eines neuen **Root**-Passworts gibt es verschiedene Möglichkeiten. Der Systemadministrator könnte das System beispielsweise über eine Live-CD booten, das Root-Dateisystem von dort aus bereitstellen und anschließend **/etc/shadow** bearbeiten. In diesem Abschnitt erläutern wir eine Methode, für die keine externen Medien erforderlich sind.



Anmerkung

Auf Red Hat Enterprise Linux 6 und früher können Administratoren das System in Runlevel 1 booten, um eine **root**-Eingabeaufforderung aufzurufen. Die größte Übereinstimmung mit dem Runlevel 1 auf einem Red Hat Enterprise Linux 8-Rechner weisen die Rettungs- und Notfallziele auf. Beide erfordern für die Anmeldung das **Root**-Passwort.

Unter Red Hat Enterprise Linux 8 ist es möglich, die Skripts, die über **initramfs** ausgeführt werden, an bestimmten Stellen pausieren zu lassen. Anschließend wird eine **Root**-Shell bereitgestellt und der Prozess fortgesetzt, sobald die Shell beendet wird. Zwar ist diese Methode hauptsächlich zum Debuggen gedacht, kann aber auch für die Wiederherstellung eines vergessenen **Root**-Passworts verwendet werden.

So greifen Sie auf diese **Root**-Shell zu:

1. Rebooten Sie das System.
2. Unterbrechen Sie den Bootloader-Zählvorgang durch das Drücken einer beliebigen Taste, mit Ausnahme der **Eingabetaste**.
3. Bewegen Sie den Cursor zum Kerneleintrag, der gebootet werden soll.
4. Drücken Sie auf **e**, um den ausgewählten Eintrag zu bearbeiten.
5. Bewegen Sie den Cursor zur Kernelbefehlszeile (Die Zeile beginnt mit **linux**).
6. Hängen Sie **rd.break** an. Bei dieser Option wird das System unterbrochen, kurz bevor das System die Steuerung vom **initramfs** das eigentliche System übergibt.
7. Drücken Sie **Strg+x**, um mit den Änderungen zu booten.

An diesem Punkt zeigt das System eine **Root**-Shell an, wobei das aktuelle Root-Dateisystem auf der Disk auf **/sysroot** schreibgeschützt bereitgestellt wird. Da die Fehlerbehebung häufig Änderungen am Root-Dateisystem erfordert, müssen Sie das Root-Dateisystem in den Lese-/Schreibzugriff ändern. Im folgenden Schritt wird gezeigt, wie die Option **remount**, **rw** zum Befehl **mount** das Dateisystem mit der neuen festgelegten Option (**rw**) erneut bereitstellt.



Anmerkung

Unter Umständen können vorgefertigte Images **console=-**-Argumente zur Unterstützung eines großen Bereichs am Kernel multiplizieren. Diese **console=-**-Argumente geben die Geräte zum Verwenden der Konsolenausgabe an. Die Vorsichtsmaßnahme mit **rd.break** liegt darin, dass, obwohl das System die Kernelmeldungen an alle Konsolen sendet, die Eingabeaufforderung immer die letzte Konsole verwendet. Wenn Sie keine Aufforderung erhalten, möchten Sie möglicherweise die **console=-**-Argumente temporär neu anordnen, wenn Sie die Kernelbefehlszeile im Bootloader bearbeiten.



Wichtig

Das System hat SELinux noch nicht aktiviert, daher hat jede von Ihnen erstellte Datei keinen SELinux-Kontext. Einige Tools, beispielsweise der Befehl **passwd**, erstellen zunächst eine temporäre Datei, mit der die zu bearbeitende Datei ersetzt wird, sodass effektiv eine neue Datei ohne SELinux-Kontext erstellt wird. Daher erhält die Datei **/etc/shadow** keinen SELinux-Kontext, wenn Sie den Befehl **passwd** mit **rd.break** verwenden.

Um das **Root**-Passwort von diesem Punkt aus zurückzusetzen, wenden Sie folgendes Verfahren an:

1. Stellen Sie **/sysroot** erneut mit Lese-/Schreibberechtigungen bereit.

```
switch_root:# mount -o remount,rw /sysroot
```

2. Wechseln Sie in ein **chroot**-Jail, in dem **/sysroot** als Root der Dateisystemstruktur behandelt wird.

```
switch_root:# chroot /sysroot
```

3. Legen Sie ein neues **Root**-Passwort fest.

```
sh-4.4# passwd root
```

4. Stellen Sie sicher, dass alle nicht gekennzeichneten Dateien, einschließlich **/etc/shadow** zu diesem Zeitpunkt, während des Boot-Vorgangs wieder gekennzeichnet werden.

```
sh-4.4# touch /.autorelabel
```

5. Geben Sie zweimal **exit** ein. Der erste Befehl beendet das **chroot**-Jail, und der zweite Befehl beendet die **initramfs**-Debug-Shell.

Kapitel 10 | Steuern des Boot-Vorgangs

An diesem Punkt führt das System weiterhin den Boot-Vorgang aus, führt eine vollständige SELinux-Umbenennung durch und rebootet das System anschließend.

Überprüfen von Protokollen

Es kann nützlich sein, einen Blick auf die Protokolle vorheriger fehlgeschlagener Boot-Vorgänge zu werfen. Wenn die Systemjournale auch nach Reboots bestehen bleiben, können Sie das Tool **journalctl** verwenden, um diese Protokolle zu untersuchen.

Beachten Sie, dass das Systemjournal standardmäßig in Verzeichnis **/run/log/journal** gespeichert wird. Das bedeutet, die Journale werden bei einem Reboot des Systems gelöscht. Legen Sie den Parameter **Storage** in **/etc/systemd/journald.conf** auf **persistent** fest, um Journale im Verzeichnis **/var/log/journal** zu speichern.

```
[root@host ~]# vim /etc/systemd/journald.conf
...output omitted...
[Journal]
Storage=persistent
...output omitted...
[root@host ~]# systemctl restart systemd-journald.service
```

Verwenden Sie die **-b**-Option von **journalctl**, um die Protokolle zu einem vorherigen Boot-Vorgang zu überprüfen. Ohne Argumente zeigt die Option **-b** nur die Meldungen seit dem letzten Boot an. Mit einer negativen Zahl als Argument werden die Protokolle der vorherigen Boots angezeigt.

```
[root@host ~]# journalctl -b -1 -p err
```

Durch diesen Befehl werden alle Benachrichtigungen angezeigt, die im vorherigen Boot-Vorgang als Fehler oder schlimmer bewertet wurden.

Reparieren von Systemd-Boot-Problemen

Red Hat Enterprise Linux 8 stellt die folgenden Tools zur Verfügung, um Probleme beim Servicestart zur Startzeit zu beheben.

Aktivieren der frühen Debug-Shell

Durch die Aktivierung des Services **debug-shell** mit **systemctl enable debug-shell.service** erzeugt das System während der Boot-Sequenz frühzeitig eine **Root**-Shell auf **TTY9 (Strg+Alt+F9)**. Diese Shell wird automatisch als **root** angemeldet, sodass Administratoren das System debuggen können, während das Betriebssystem weiterhin gebootet wird.



Warnung

Vergessen Sie nicht, den Service **debug-shell.service** zu deaktivieren, sobald Sie fertig sind. Andernfalls bleibt eine nicht authentifizierte **Root**-Shell geöffnet, auf die jeder Benutzer mit lokalem Konsolenzugriff zugreifen kann.

Verwenden von Notfall- und Rettungszielen

Durch das Anhängen von **systemd.unit=rescue.target** oder **systemd.unit=emergency.target** an die Kernelbefehlszeile des Bootloaders startet das

System nicht auf normale Art und Weise, sondern in eine Rettungs- oder Notfall-Shell. Beide Shells erfordern das **Root**-Passwort.

Durch das Notfallziel bleibt das Root-Dateisystem weiterhin schreibgeschützt bereitgestellt, während das Rettungsziel auf den Abschluss von **sysinit.target** wartet, sodass weitere Systemkomponenten initialisiert werden, beispielsweise der Protokollierungsservice oder die Dateisysteme. Der Root-Benutzer kann erst Änderungen an /etc/fstab vornehmen, wenn das Laufwerk in einem Lese-/Schreibzustand **mount -o remount, rw** / erneut bereitgestellt ist.

Administratoren können mit diesen Shells alle Fehler beheben, die verhindern, dass das System ordnungsgemäß gebootet wird; zum Beispiel eine Abhängigkeitsschleife zwischen Services oder ein falscher Eintrag in **/etc/fstab**. Nach dem Beenden der Shells wird der reguläre Boot-Vorgang fortgesetzt.

Ermitteln von festgefahrenen Jobs

Während des Starts zeigt **Systemd** eine Reihe von Aufgaben an. Wenn einige der Aufgaben nicht abgeschlossen werden können, wird der Weg für andere Aufgaben versperrt, sodass diese nicht ausgeführt werden können. Zum Überprüfen der aktuellen Aufgabenliste können Administratoren den Befehl **systemctl list-jobs** ausführen. Aufgaben, die als „running“ aufgelistet werden, müssen abgeschlossen werden, bevor mit Aufgaben, die als „waiting“ gekennzeichnet sind, fortgefahrene werden kann.



Literaturhinweise

Manpages **dracut cmdline(7)**, **systemd-journald(8)**, **journald.conf(5)**, **journalctl(1)** und **systemctl(1)**

► Angeleitete Übung

Zurücksetzen des Root-Passworts

In dieser Übung setzen Sie das Passwort **root** auf einem System zurück.

Ergebnisse

Sie sollten in der Lage sein, ein verlorenes **Root**-Passwort zurückzusetzen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab boot-resetting start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Dadurch wird auch das Passwort **root** auf eine zufällige Zeichenfolge zurückgesetzt und ein höherer Timeout für das GRUB2-Menü festgelegt.

```
[student@workstation ~]$ lab boot-resetting start
```

- ▶ 1. Rebooten Sie **servera**, und unterbrechen Sie den Zählvorgang im Bootloader-Menü.
 - 1.1. Suchen Sie entsprechend Ihrer Kursumgebung nach dem Symbol für die **servera**-Konsole. Öffnen Sie die Konsole.
Geben Sie **Strg+Alt+Entf** über die entsprechende Taste oder den entsprechenden Menüeintrag in Ihr System ein.
 - 1.2. Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste, um den Zählvorgang zu unterbrechen, ausgenommen die **Eingabetaste**.
- ▶ 2. Bearbeiten Sie den Bootloader-Standardeintrag im Arbeitsspeicher, um den Boot-Vorgang abzubrechen, nachdem der Kernel alle Dateisysteme bereitgestellt hat, doch bevor er die Kontrolle an **systemd** übergibt.
 - 2.1. Markieren Sie mit den Cursor-Tasten den standardmäßigen Bootloader-Eintrag.
 - 2.2. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
 - 2.3. Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt.
 - 2.4. Drücken Sie auf **Ende**, um den Cursor zum Ende der Zeile zu bewegen.
 - 2.5. Fügen Sie **rd.break** an das Ende der Zeile an.
 - 2.6. Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.
- ▶ 3. Stellen Sie an der Eingabeaufforderung **switch_root** das Dateisystem **/sysroot** erneut mit Lese-/Schreibberechtigungen bereit, und verwenden Sie **chroot**, um in ein **chroot**-Jail unter **/sysroot** zu wechseln.

```
switch_root:/# mount -o remount,rw /sysroot  
switch_root:/# chroot /sysroot
```

- 4. Ändern Sie das **root**-Passwort zurück in **redhat**.

```
sh-4.4# 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. Konfigurieren Sie das System so, dass es nach dem Boot-Vorgang automatisch eine vollständige SELinux-Umbenennung durchführt. Dies ist erforderlich, da der Befehl **passwd** die Datei **/etc/shadow** ohne einen SELinux-Kontext neu erstellt.

```
sh-4.4# touch /.autorelabel
```

- 6. Geben Sie zweimal **exit** ein, um mit dem regulären Boot-Vorgang fortzufahren. Das System führt eine SELinux-Umbenennung durch und wird anschließend eigenständig erneut neu gestartet. Wenn das System in Betrieb ist, überprüfen Sie Ihre Arbeit, indem Sie sich als **root** bei der Konsole anmelden. Verwenden Sie **redhat** als Passwort.

Beenden

Führen Sie auf **workstation** das Skript **lab boot-resetting finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab boot-resetting finish
```

Hiermit ist die angeleitete Übung beendet.

Beheben von Dateisystemproblemen während des Boot-Vorgangs

Ziele

In diesem Abschnitt wird beschrieben, wie Sie die Dateisystemkonfiguration oder Beschädigungsprobleme manuell reparieren, die den Boot-Vorgang stoppen.

Diagnostizieren und Beheben von Dateisystemproblemen

Fehler in **/etc/fstab** und beschädigte Dateisysteme können verhindern, dass ein System gebootet wird. In den meisten Fällen wechselt **systemd** in eine Notfallreparatur-Shell, für die das Passwort **root** erforderlich ist.

In der folgenden Tabelle sind häufige Fehler und ihre Lösungen aufgeführt.

Allgemeine Dateisystemprobleme

Das Problem	Ergebnis
Beschädigtes Dateisystem	systemd versucht, das Dateisystem zu reparieren. Wenn das Problem für eine automatische Korrektur zu schwerwiegend ist, lässt das System den Benutzer in eine Notfall-Shell wechseln.
Nicht vorhandenes Gerät oder nicht vorhandene UUID angegeben in /etc/fstab	systemd wartet für einen festgelegten Zeitraum, ob das Gerät verfügbar wird. Ist dies nicht der Fall, wird der Benutzer vom System nach dem Timeout zu einer Notfall-Shell weitergeleitet.
Kein vorhandener Bereitstellungspunkt in /etc/fstab	Der Benutzer wird vom System zu einer Notfall-Shell weitergeleitet.
Falsche Bereitstellungsoption angegeben in /etc/fstab	Der Benutzer wird vom System zu einer Notfall-Shell weitergeleitet.

In jedem Fall können Administratoren auch das Notfallziel verwenden, um das Problem zu diagnostizieren und zu beheben, da keine Dateisysteme bereitgestellt werden, bis die Notfall-Shell angezeigt wird.



Anmerkung

Vergessen Sie beim Verwenden der Notfall-Shell zum Beheben von Dateisystemproblemen nicht, **systemctl daemon-reload** nach der Bearbeitung von **/etc/fstab** auszuführen. Ohne das erneute Laden verwendet **Systemd** weiterhin die alte Version.

Die Option **nofail** in einem Eintrag in der Datei **/etc/fstab** erlaubt dem System zu booten, selbst wenn die Bereitstellung dieses Dateisystems nicht erfolgreich ist. Sie sollten diese Option unter normalen Umständen *nicht* verwenden. Mit **nofail** kann eine Anwendung mit ihrem fehlenden Storage beginnen, was schwerwiegende Folgen haben kann.



Literaturhinweise

Manpages **systemd-fsck(8)**, **systemd-fstab-generator(8)** und **systemd.mount(5)**

► Angeleitete Übung

Beheben von Dateisystemproblemen während des Boot-Vorgangs

In dieser Übung wird ein System nach einer Fehlkonfiguration in **/etc/fstab** wiederhergestellt, wodurch der Boot-Vorgang fehlschlägt.

Ergebnisse

Sie sollten in der Lage sein, **/etc/fstab**-Probleme zu diagnostizieren und den Notfallmodus zum Wiederherstellen des Systems zu verwenden.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab boot-repairing start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Darüber hinaus führt er zu einem Dateisystemproblem, legt einen höheren Timeout für das GRUB2-Menü fest und rebootet **servera**.

```
[student@workstation ~]$ lab boot-repairing start
```

- ▶ 1. Greifen Sie auf die Konsole **servera** zu, und beachten Sie, dass der Boot-Vorgang frühzeitig hängengeblieben ist.
 - 1.1. Suchen Sie entsprechend Ihrer Kursumgebung nach dem Symbol für die **servera**-Konsole. Öffnen Sie die Konsole.

Beachten Sie, dass ein Startjob scheinbar nicht abgeschlossen ist. Nehmen Sie sich einen Augenblick Zeit, um Vermutungen zu einer möglichen Ursache für dieses Verhalten anzustellen.
 - 1.2. Geben Sie für den Reboot **Strg+Alt+Entf** über die entsprechende Taste oder den entsprechenden Menüeintrag in Ihr System ein. Bei diesem besonderen Boot-Problem bricht diese Schlüsselsequenz möglicherweise nicht sofort den laufenden Job ab, und Sie müssen möglicherweise warten, bis der Timeout überschritten wird, bevor das System rebootet.

Wenn Sie warten, bis ein Timeout für die Aufgabe auftritt, ohne **Strg+Alt+Entf** zu drücken, erzeugt das System von selbst schließlich eine Notfall-Shell.
 - 1.3. Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste, um den Zählvorgang zu unterbrechen, ausgenommen die **Eingabetaste**.
- ▶ 2. Bei der Betrachtung des Fehlers während des vorherigen Boot-Vorgangs erscheint es, dass zumindest noch Teile des Systems funktionieren. Da Sie das **Root**-Passwort kennen (**redhat**), versuchen Sie, einen Notfall-Boot durchzuführen.
 - 2.1. Markieren Sie mit den Cursor-Tasten den standardmäßigen Bootloader-Eintrag.
 - 2.2. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.

- 2.3. Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt.
- 2.4. Drücken Sie auf **Ende**, um den Cursor zum Ende der Zeile zu bewegen.
- 2.5. Fügen Sie **systemd.unit=emergency.target** an das Ende der Zeile an.
- 2.6. Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.

► 3. Melden Sie sich im Notfallmodus an. Das **root**-Passwort lautet **redhat**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@servera ~]#
```

► 4. Bestimmen Sie, welche Dateisysteme aktuell bereitgestellt sind.

```
[root@servera ~]# mount
...output omitted...
/dev/vda1 on / type xfs (ro,relatime,seclabel,attr2,inode64,noquota)
...output omitted...
```

Beachten Sie, dass das Root-Dateisystem schreibgeschützt bereitgestellt ist.

► 5. Stellen Sie das Root-Dateisystem mit Lese- und Schreibzugriff neu bereit.

```
[root@servera ~]# mount -o remount,rw /
```

► 6. Führen Sie den Befehl **mount -a** aus, um zu versuchen, alle anderen Dateisysteme bereitzustellen. Mit der Option **--all** (-a) stellt der Befehl alle in **/etc/fstab** aufgelisteten Dateisysteme bereit, die noch nicht bereitgestellt sind.

```
[root@servera ~]# mount -a
mount: /RemoveMe: mount point does not exist.
```

► 7. Bearbeiten Sie **/etc/fstab**, um das Problem zu beheben.

- 7.1. Entfernen Sie die falsche Zeile, oder kommentieren Sie sie aus.

```
[root@servera ~]# vim /etc/fstab
...output omitted...
# /dev/sdz1  /RemoveMe  xfs  defaults  0 0
```

- 7.2. Aktualisieren Sie **systemd**, damit das System die neue **/etc/fstab**-Konfiguration registriert.

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

► 8. Verifizieren Sie, dass **/etc/fstab** jetzt korrekt ist, indem Sie versuchen, alle Einträge bereitzustellen.

```
[root@servera ~]# mount -a
```

- 9. Rebooten Sie das System, und warten Sie, bis der Boot-Vorgang abgeschlossen ist. Das System sollte nun ordnungsgemäß gebootet werden.

```
[root@servera ~]# systemctl reboot
```

Beenden

Führen Sie auf **workstation** das Skript **lab boot-repairing finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab boot-repairing finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Steuern des Boot-Vorgangs

Leistungscheckliste

In dieser praktischen Übung setzen Sie das Passwort **root** auf einem System zurück, führen eine Wiederherstellung nach einer Fehlkonfiguration durch und legen das standardmäßige boot-Ziel fest.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Zurücksetzen eines verlorenen **Root**-Passworts.
- Diagnostizieren und Beheben von Boot-Problemen.
- Festlegen des standardmäßigen **systemd**-Ziels.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab boot-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Darüber hinaus führt er zu einem Dateisystemproblem, setzt das **Root**-Passwort zurück, legt einen höheren Timeout für das GRUB2-Menü fest und rebootet **serverb**.

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

1. Setzen Sie auf **serverb** das **Root**-Passwort auf **redhat** zurück.
Suchen Sie entsprechend Ihrer Kursumgebung nach dem Symbol für die **serverb**-Konsole.
Arbeiten Sie von dieser Konsole aus.
2. Das System bootet nicht. Ein Startauftrag ist offenbar nicht vollständig. Beheben Sie das Problem an der Konsole.
3. Ändern Sie das Standardziel **systemd** auf **serverb**, damit das System beim Boot automatisch eine grafische Oberfläche startet.
Auf **serverb** ist noch keine grafische Oberfläche installiert. Legen Sie für diese Übung nur das Standardziel fest, und installieren Sie die Pakete nicht.

Bewertung

Führen Sie auf **workstation** das Skript **lab boot-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab boot-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

► Lösung

Steuern des Boot-Vorgangs

Leistungscheckliste

In dieser praktischen Übung setzen Sie das Passwort **root** auf einem System zurück, führen eine Wiederherstellung nach einer Fehlkonfiguration durch und legen das standardmäßige boot-Ziel fest.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Zurücksetzen eines verlorenen **Root**-Passworts.
- Diagnostizieren und Beheben von Boot-Problemen.
- Festlegen des standardmäßigen **systemd**-Ziels.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab boot-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Darüber hinaus führt er zu einem Dateisystemproblem, setzt das **Root**-Passwort zurück, legt einen höheren Timeout für das GRUB2-Menü fest und rebootet **serverb**.

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

1. Setzen Sie auf **serverb** das **Root**-Passwort auf **redhat** zurück.

Suchen Sie entsprechend Ihrer Kursumgebung nach dem Symbol für die **serverb**-Konsole. Arbeiten Sie von dieser Konsole aus.

- 1.1. Geben Sie **Strg+Alt+Entf** über die entsprechende Taste oder den entsprechenden Menüeintrag in Ihr System ein.
- 1.2. Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste, um den Zählvorgang zu unterbrechen, ausgenommen die **Eingabetaste**.
- 1.3. Markieren Sie mit den Cursor-Tasten den standardmäßigen Bootloader-Eintrag.
- 1.4. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
- 1.5. Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt.
- 1.6. Drücken Sie auf **Ende**, um den Cursor zum Ende der Zeile zu bewegen.
- 1.7. Fügen Sie **rd.break** an das Ende der Zeile an.
- 1.8. Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.

Kapitel 10 | Steuern des Boot-Vorgangs

- 1.9. Stellen Sie an der Eingabeaufforderung **switch_root** das Dateisystem **/sysroot** erneut mit Lese-/Schreibberechtigungen bereit, und verwenden Sie **chroot**, um in ein **chroot**-Jail unter **/sysroot** zu wechseln.

```
switch_root:/# mount -o remount,rw /sysroot
switch_root:/# chroot /sysroot
```

- 1.10. Legen Sie das **root**-Passwort auf **redhat** fest.

```
sh-4.4# 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.
```

- 1.11. Konfigurieren Sie das System so, dass es nach dem Boot-Vorgang automatisch eine vollständige SELinux-Umbenennung durchführt.

```
sh-4.4# touch /.autorelabel
```

- 1.12. Geben Sie zweimal **exit** ein, um den Boot Ihres Systems fortzusetzen. Das System kann aufgrund eines Problems nicht gebootet werden, das Sie im nächsten Schritt beheben.

2. Das System bootet nicht. Ein Startauftrag ist offenbar nicht vollständig. Beheben Sie das Problem an der Konsole.

- 2.1. Booten Sie das System im Notfallmodus. Rebooten Sie dazu **serverb**. Geben Sie **Strg+Alt+Entf** über die entsprechende Taste oder den entsprechenden Menüeintrag in Ihr System ein.
- 2.2. Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste, um den Zählvorgang zu unterbrechen, ausgenommen die **Eingabetaste**.
- 2.3. Markieren Sie mit den Cursor-Tasten den standardmäßigen Bootloader-Eintrag.
- 2.4. Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
- 2.5. Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt.
- 2.6. Drücken Sie auf **Ende**, um den Cursor zum Ende der Zeile zu bewegen.
- 2.7. Fügen Sie **systemd.unit=emergency.target** an das Ende der Zeile an.
- 2.8. Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.
- 2.9. Melden Sie sich im Notfallmodus an. Das **root**-Passwort lautet **redhat**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@serverb ~]#
```

- 2.10. Stellen Sie das **/**-Dateisystem im Modus Lesen/Schreiben neu bereit.

```
[root@serverb ~]# mount -o remount,rw /
```

- 2.11. Führen Sie den Befehl **mount -a** aus, um zu versuchen, alle anderen Dateisysteme bereitzustellen.

```
[root@serverb ~]# mount -a
mount: /olddata: can't find UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc.
```

- 2.12. Bearbeiten Sie **/etc/fstab**, um die falsche Zeile zu entfernen oder auszkommentieren.

```
[root@serverb ~]# vim /etc/fstab
...
#UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc  /olddata  xfs  defaults  0 0
```

- 2.13. Aktualisieren Sie **systemd**, damit das System die neue **/etc/fstab**-Konfiguration registriert.

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

- 2.14. Verifizieren Sie, dass **/etc/fstab** jetzt korrekt ist, indem Sie versuchen, alle Einträge bereitzustellen.

```
[root@serverb ~]# mount -a
[root@serverb ~]#
```

- 2.15. Rebooten Sie das System, und warten Sie, bis der Boot-Vorgang abgeschlossen ist. Da Sie im ersten Schritt die Datei **/.autorelabel** erstellt haben, führt das System nach dem Festlegen des Passworts **root** eine SELinux-Umbenennung aus und führt dann selbstständig einen Reboot durch. Das System sollte nun ordnungsgemäß gebootet werden.

```
[root@serverb ~]# systemctl reboot
```

3. Ändern Sie das Standardziel **systemd** auf **serverb**, damit das System beim Boot automatisch eine grafische Oberfläche startet.

Auf **serverb** ist noch keine grafische Oberfläche installiert. Legen Sie für diese Übung nur das Standardziel fest, und installieren Sie die Pakete nicht.

- 3.1. Melden Sie sich als der Benutzer **root** bei **serverb** an. Verwenden Sie **redhat** als Passwort.
- 3.2. Führen Sie den Befehl **systemctl set-default** aus, um **graphical.target** als Standardziel festzulegen.

```
[root@serverb ~]# systemctl set-default graphical.target
```

- 3.3. Führen Sie den Befehl **systemctl get-default** aus, um die Ergebnisse Ihrer Arbeit zu verifizieren.

```
[root@serverb ~]# systemctl get-default  
graphical.target
```

3.4. Melden Sie sich von **serverb** ab.

```
[root@serverb ~]# exit
```

Bewertung

Führen Sie auf **workstation** das Skript **lab boot-review grade** aus, um den Erfolg dieser Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab boot-review finish** aus, um die praktische Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Über **systemctl reboot** und **systemctl poweroff** wird das System gerebootet bzw. ausgeschaltet.
- Mit **systemctl isolate target-name.target** können Sie während des Betriebs zu einem neuen Ziel wechseln.
- **systemctl get-default** und **systemctl set-default** können verwendet werden, um das Standardziel abzufragen und festzulegen.
- Verwenden Sie **rd.break** in der Kernelbefehlszeile, um den Boot-Vorgang zu unterbrechen, bevor die Kontrolle von **initramfs** weitergegeben wird. Das Root-Dateisystem ist schreibgeschützt unter **/sysroot** bereitgestellt.
- Das Notfallziel kann verwendet werden, um Dateisystemprobleme zu diagnostizieren und zuheben.

Kapitel 11

Verwalten der Netzwerksicherheit

Ziel

Steuern Sie mithilfe der System-Firewall und der SELinux-Regeln die Netzwerkverbindungen zu Services.

Ziele

- Akzeptieren oder Ablehnen von Netzwerkverbindungen zu Systemservices mithilfe von Firewalld-Regeln.
- Steuern, ob Netzwerkservices bestimmte Netzwerkports verwenden können, indem Sie SELinux-Port-Labels verwalten.

Abschnitte

- Verwalten von Server-Firewalls (und angeleitete Übung)
- Steuern der SELinux-Portbezeichnung (und angeleitete Übung)

Praktische Übung

Verwalten von Server-Firewalls

Verwalten von Server-Firewalls

Ziele

In diesem Abschnitt wird beschrieben, wie Sie mittels Firewalld-Regeln Netzwerkverbindungen zu Systemservices akzeptieren oder ablehnen.

Konzepte der Firewall-Architektur

Der Linux-Kernel enthält **netfilter**, ein Framework für Netzwerkdatenverkehrsvorgänge, beispielsweise die Paketfilterung, die Netzwerkadressübersetzung und die Portübersetzung. Durch die Implementierung von Handlern im Kernel, die Funktionsaufrufe und Meldungen abfangen, ermöglicht **Netfilter** anderen Kernelmodulen direkt mit dem Networking-Stack des Kernels zu kommunizieren. Firewall-Software verwendet diese Hooks, um Filterregeln und Funktionen zum Ändern von Paketen zu registrieren, wodurch jedes Paket, das den Networking-Stack durchläuft, verarbeitet werden kann. Alle eingehenden, ausgehenden oder weitergeleiteten Netzwerkpakete können programmgesteuert geprüft, geändert, verworfen oder geroutet werden, bevor sie auf Komponenten oder Anwendungen des Benutzerbereichs zugreifen. **Netfilter** ist die primäre Komponente in Red Hat Enterprise Linux 8-Firewalls.

Nftables verbessert Netfilter

Der Linux-Kernel enthält zu dem **nftables**, ein neues Filter- und Paketklassifizierungs-Subsystem, das erweiterte Teilmengen des **Netfilter**-Codes aufweist, jedoch die **Netfilter**-Architektur beibehält, darunter beispielsweise Networking-Stack-Hooks, das Verbindungsverfolgungssystem und die Protokollierungsfunktion. Die Vorteile der **nftables**-Aktualisierung besteht in der schnelleren Paketverarbeitung, in schnelleren Regelsatzaktualisierungen und in der gleichzeitigen IPv4- und IPv6-Verarbeitung nach denselben Regeln. Ein weiterer wesentlicher Unterschied zwischen **nftables** und dem ursprünglichen **Netfilter** besteht in ihren Schnittstellen. **Netfilter** wird durch mehrere Dienstprogramm-Frameworks konfiguriert, einschließlich **iptables**, **ip6tables**, **arptables** und **ebtables**, die mittlerweile veraltet sind. Nftables verwendet das einzelne **nft**-User-Space-Dienstprogramm. Dieses Dienstprogramm ermöglicht die gesamte Protokollverwaltung über eine einzige Schnittstelle, wodurch Verlaufskonflikte durch verschiedene Frontends und mehrere **Netfilter**-Schnittstellen ausgeschlossen werden.

Einführung in firewalld

firewalld ist ein dynamischer Firewall-Manager, ein Frontend zum **nftables**-Framework mit dem Befehl **nft**. Bis zur Einführung von **nftables** wurde von **firewalld** der Befehl **iptables** als eine verbesserte Alternative zum Service **iptables** verwendet, um **Netfilter** direkt zu konfigurieren. In RHEL 8 ist **firewalld** weiterhin das empfohlene Frontend und verwaltet Firewall-Regelsätze mit **nft**. **firewalld** ist weiterhin in der Lage, **iptables**-Konfigurationsdateien und -Regelsätze zu lesen und zu verwalten. Dazu wird **xtables-nft-multi** verwendet, um **iptables**-Objekte direkt in **nftables**-Regeln und -Objekte zu übersetzen. Obwohl dringend davon abgeraten wird, kann **firewalld** für komplexe Anwendungsfälle, wo vorhandene **iptables**-Regelsätze von **nft**-Transaktionen nicht ordnungsgemäß verarbeitet werden können, so konfiguriert werden, dass eine Zurücksetzung auf das **iptables**-Backend vorgenommen wird.

Anwendungen fragen das Subsystem mit der **D-Bus**-Schnittstelle ab. Das im `firewalld`-RPM-Paket verfügbare **firewalld**-Subsystem ist in der minimalen Installation nicht enthalten, jedoch in der Basisinstallation. **firewalld** vereinfacht die Verwaltung der Firewall, indem der gesamte Netzwerdatenverkehr in Zonen klassifiziert wird. Auf der Grundlage von Kriterien, wie etwa der Quell-IP-Adresse eines Pakets oder der eingehenden Netzwerkschnittstelle, wird der Datenverkehr an die Firewall-Regeln für die jeweilige Zone umgeleitet. Jede Zone besitzt ihre eigene Liste von Ports und Services, die offen oder geschlossen sind.



Anmerkung

Für Laptops und andere Rechner, die regelmäßig das Netzwerk wechseln, kann mithilfe von NetworkManager die Firewall-Zone für die einzelnen Verbindungen eingestellt werden. Die Zonen werden mit Regeln, die für die jeweiligen Verbindungen gelten, angepasst.

Dies ist insbesondere beim ständigen Wechseln zwischen drahtlosen Netzwerken in den Zonen `home`, `work` und `public` hilfreich. Ein Benutzer möchte eventuell, dass der `sshd`-Service seines Systems erreichbar ist, wenn er mit dem Heim- und Unternehmensnetzwerk verbunden ist, nicht aber, wenn er eine Verbindung mit dem öffentlichen Drahtlosnetzwerk im Café nebenan herstellt.

Firewalld überprüft die Quelladresse für jedes in das System eingehende Paket. Wenn diese Ausgangsadresse einer bestimmten Zone zugewiesen ist, gelten die Regeln für diese Zone. Wenn die Quelladresse keiner Zone zugewiesen ist, ordnet **firewalld** das Paket der Zone für die eingehende Netzwerkschnittstelle zu, und es gelten die Regeln für diese Zone. Wenn die Netzwerkschnittstelle aus einem bestimmten Grund nicht mit der Zone verknüpft ist, dann verknüpft **firewalld** das Paket mit der Standardzone.

Die Standardzone ist keine separate Zone, sondern eine Bezeichnung für eine vorhandene Zone. Anfänglich legt **firewalld** die Zone **public** als Standard fest und ordnet die Loopback-Schnittstelle **lo** der Zone **trusted** zu.

Die meisten Zonen lassen den Datenverkehr durch die Firewall zu, wenn dieser mit einer Liste mit bestimmten Ports und Protokollen, beispielsweise **631/udp**, oder vordefinierten Services, beispielsweise **ssh**, übereinstimmt. Wenn der Datenverkehr nicht mit einem zugelassenen Port, Protokoll oder Service übereinstimmt, wird er in der Regel abgelehnt. (Die **vertrauenswürdige** Zone, die standardmäßig jeden Datenverkehr zulässt, bildet hierbei eine Ausnahme.)

Vordefinierte Zonen

Firewalld weist vordefinierte Zonen auf, die Sie einzeln anpassen können. Standardmäßig lassen alle Zonen den eingehenden Datenverkehr, der Teil einer vom System angestoßenen Kommunikation ist, sowie sämtlichen ausgehenden Datenverkehr zu. Die folgende Tabelle beschreibt diese anfängliche Zonenkonfiguration.

Standardkonfiguration der Firewalld-Zonen

Zonenname	Standardkonfiguration
trusted	Sämtlichen eingehenden Datenverkehr zulassen.

Zonenname	Standardkonfiguration
home	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit den vordefinierten Services ssh , mdns , ipp-client , samba-client oder dhcpv6-client übereinstimmt.
intern	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit den vordefinierten Services ssh , mdns , ipp-client , samba-client oder dhcpv6-client übereinstimmt (wie bei der home -Zone).
work	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit den vordefinierten Services ssh , ipp-client oder dhcpv6-client übereinstimmt.
öffentlich	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit dem vordefinierten Service ssh oder dhcpv6-client übereinstimmt. <i>Die Standardzone für neu hinzugefügte Netzwerkschnittstellen.</i>
external	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit dem vordefinierten Service ssh übereinstimmt. Der ausgehende IPv4-Datenverkehr, der durch diese Zone weitergeleitet wird, wird <i>maskiert</i> , sodass es den Anschein hat, dieser stamme von der IPv4-Adresse der ausgehenden Netzwerkschnittstelle.
dmz	Eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht oder mit dem vordefinierten Service ssh übereinstimmt.
block	Sämtlicher eingehender Datenverkehr wird abgelehnt, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht.
drop	Sämtlicher eingehender Datenverkehr wird verworfen, sofern er nicht mit ausgehendem Datenverkehr in Beziehung steht (antwortet nicht mit ICMP-Fehlern).

Eine Liste der verfügbaren vordefinierten Zonen und deren vorgesehene Verwendung finden Sie unter **firewalld.zones**(5).

Vordefinierte Services

Firewalld enthält einige vordefinierte Services. Diese Servicedefinitionen helfen Ihnen, bestimmte zu konfigurierende Netzwerkservices zu identifizieren. Anstatt über relevante Ports für den Service **samba-client** zu verfügen, geben Sie beispielsweise den vorgefertigten Service **samba-client** zum Konfigurieren der richtigen Ports und Protokolle an. Die folgende Tabelle enthält die vordefinierten Services, die in der anfänglichen Firewall-Zonenkonfiguration verwendet werden.

Ausgewählte vordefinierte Firewalld-Services

Servicename	Konfiguration
ssh	Lokaler SSH-Server. Datenverkehr an 22/tcp
dhcpv6-client	Lokaler DHCPv6-Client. Datenverkehr an 546/udp an der IPv6-Netzwerkadresse fe80::/64
ipp-client	Lokales Drucken über IPP. Datenverkehr an 631/udp.
samba-client	Lokale Windows-Datei und Druckerfreigabe-Client Datenverkehr an 137/udp und 138/udp.
mdns	Multicast-DNS (mDNS): Lokaler Link zur Namensauflösung. Datenverkehr an 5353/udp für die Multicast-Adressen 224.0.0.251 (IPv4) oder ff02::fb (IPv6).



Anmerkung

Viele vordefinierten Services sind im Paket `firewalld` enthalten. Verwenden Sie `firewall-cmd --get-services`, um sie aufzulisten. Die Konfigurationsdateien für die vordefinierten Services finden Sie in `/usr/lib/firewalld/services` in einem von `firewalld.zone`(5) definierten Format.

Verwenden Sie vordefinierte Services, oder geben Sie den erforderlichen Port und das erforderliche Protokoll direkt an. Die grafische Benutzeroberfläche der Web Console wird verwendet, um die vordefinierten Services zu überprüfen und um zusätzliche Services zu definieren.

Konfigurieren der Firewall

Systemadministratoren interagieren auf drei Arten mit `firewalld`:

- Direktes Bearbeiten der Konfigurationsdateien in `/etc/firewalld/` (wird in diesem Kapitel nicht erläutert)
- Die grafische Benutzeroberfläche der Web Console
- Das Befehlszeilentool `firewall-cmd`

Konfigurieren von Firewall-Services mit der Web Console

Melden Sie sich mit privilegiertem Zugriff an, indem Sie auf die Option **Reuse my password for privileged tasks** klicken, um Firewall-Services mit der Web Console zu konfigurieren. Dies ermöglicht dem Benutzer Befehle mit sudo-Berechtigungen auszuführen, die Firewall-Service zu ändern.

Abbildung 11.1: Die privilegierte Anmeldung bei Web Console

Kapitel 11 | Verwalten der Netzwerksicherheit

Klicken Sie im linken Navigationsmenü auf die Option **Networking**, um den Abschnitt **Firewall** auf der Hauptnetzwerkseite anzuzeigen. Klicken Sie auf den Link **Firewall**, um auf die Liste der zulässigen Services zuzugreifen.

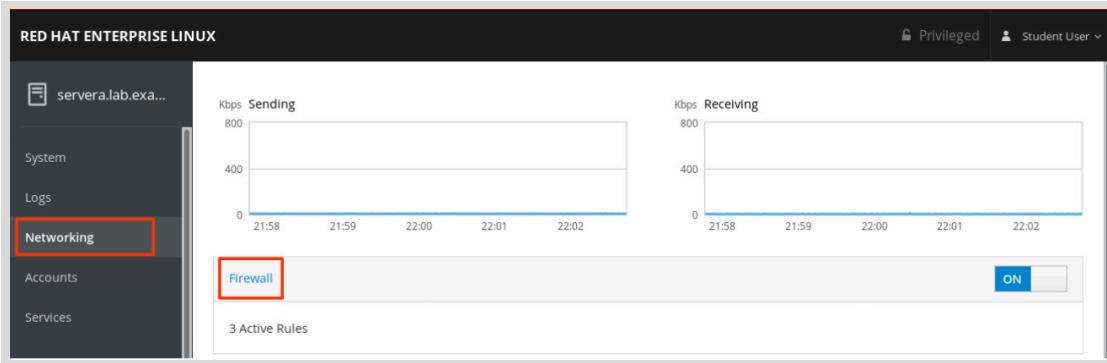


Abbildung 11.2: Das Web Console-Netzwerk

Die aufgeführten zulässigen Services sind derzeit von der Firewall zugelassen. Klicken Sie auf den Pfeil (>) links neben dem Servicenamen, um Servicedetails anzuzeigen. Klicken Sie zum Hinzufügen eines Services in der oberen rechten Ecke der Seite **Firewall Allowed Services** auf die Schaltfläche **Add Services...**

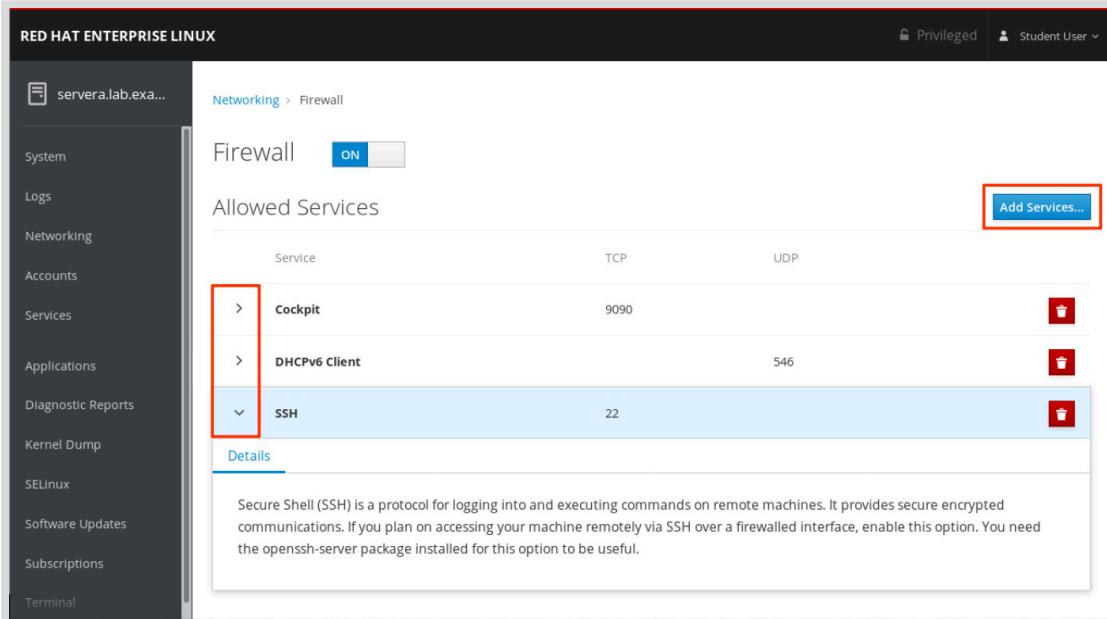


Abbildung 11.3: Die Liste der zulässigen Services der Web Console-Firewall

Auf der Seite **Add Services** werden die verfügbaren vordefinierten Services angezeigt.

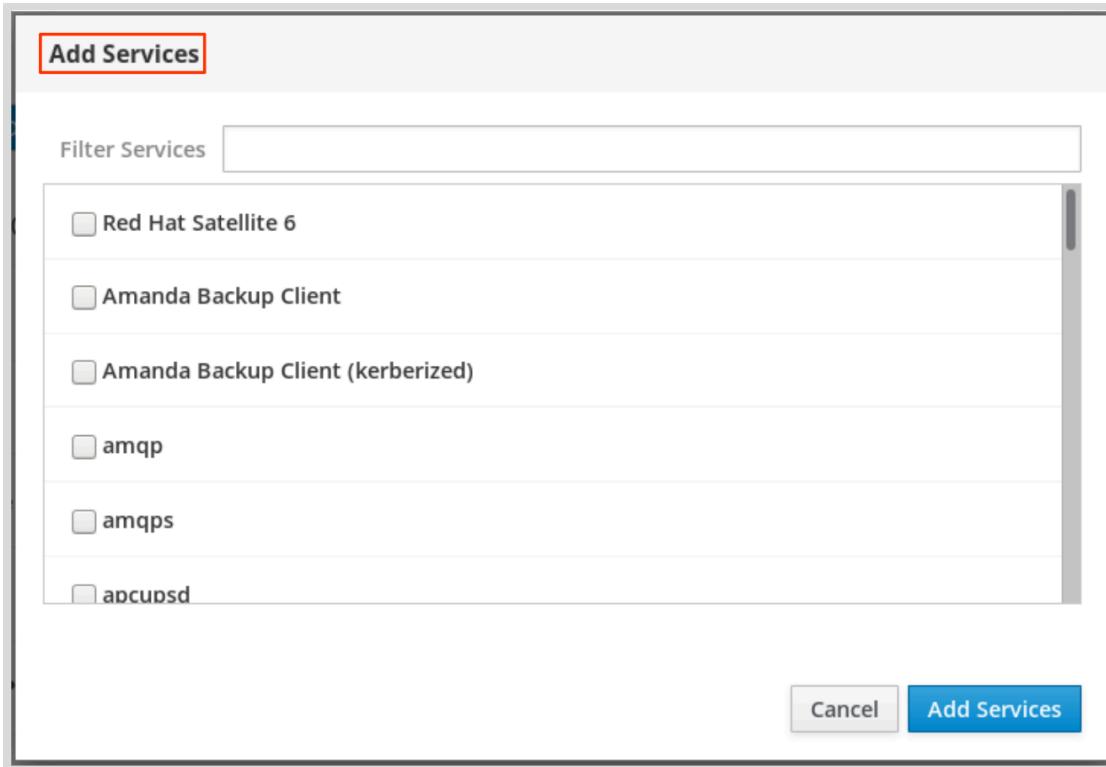


Abbildung 11.4: Die Web Console fügt eine Serviceschnittstelle für die Firewall hinzu.

Blättern Sie zum Hinzufügen eines Services durch die Liste, oder geben Sie im Textfeld **Filter Services** eine Auswahl ein. Im folgenden Beispiel wird die Zeichenfolge **http** in das Suchtextfeld eingegeben, um nach Services zu suchen, die diese Zeichenfolge enthalten, d. h. webbezogene Services. Aktivieren Sie das Kontrollkästchen links neben den Services, um die Firewall zuzulassen. Klicken Sie zum Abschließen des Vorgangs auf die Schaltfläche **Add Services**.

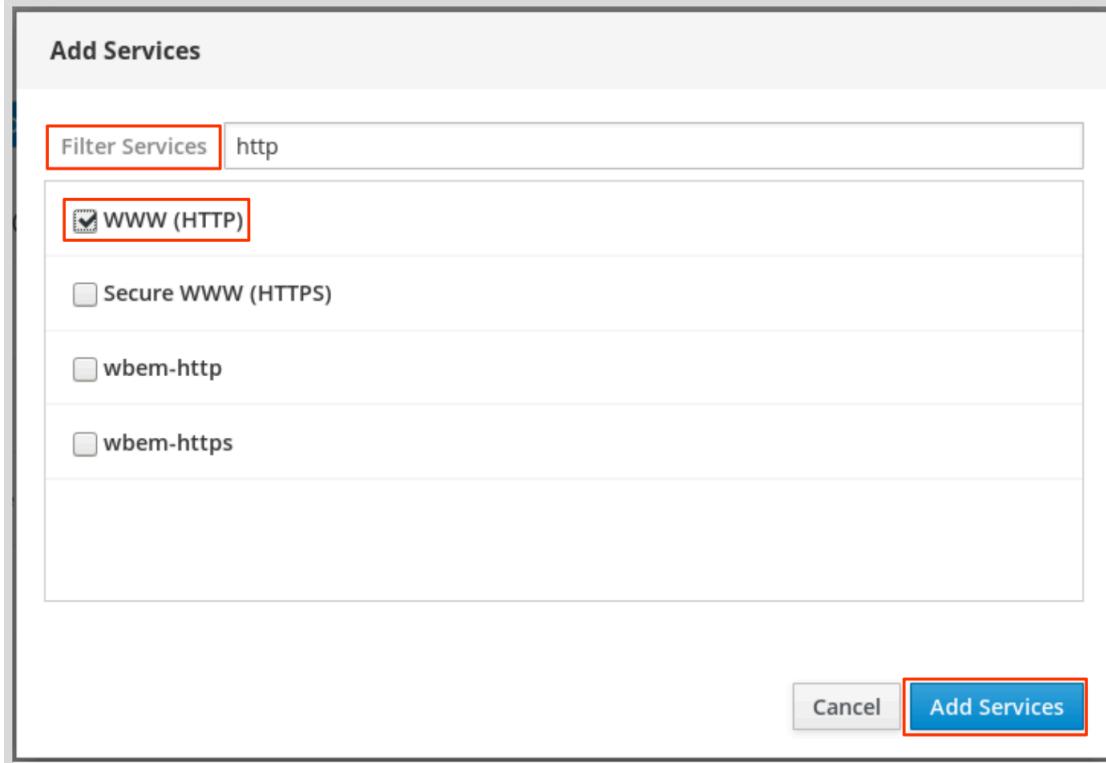


Abbildung 11.5: Die filterbasierte Suche der Web Console-Firewall-Services

Die Oberfläche kehrt zur Seite **Firewall Allowed Services** zurück, wo Sie die aktualisierte Liste der zulässigen Services einsehen können.

Service	TCP	UDP
Cockpit	9090	
DHCPv6 Client	546	
SSH	22	
WWW (HTTP)	80	

Abbildung 11.6: Die Liste der zulässigen Services der Web Console-Firewall

Konfigurieren der Firewall über die Befehlszeile

Der Befehl **firewall-cmd** interagiert mit dem dynamischen Firewall-Manager **firewalld**. Er wird als Teil des **firewalld**-Hauptpakets installiert und ist für Administratoren verfügbar, die bevorzugt an der Befehlszeile arbeiten, auf Systemen ohne grafische Umgebung arbeiten oder Skripts für ein Firewall-Setup erstellen.

Die folgende Tabelle listet eine Reihe häufig verwendeter **firewall-cmd**-Befehle und jeweils eine zugehörige Erläuterung auf. Sofern nichts anderes angegeben ist, funktionieren alle Befehle in der Laufzeitkonfiguration, es sei denn, die Option **--permanent** ist ausgewählt. Wenn die Option **--permanent** angegeben ist, müssen Sie die Einstellung aktivieren, indem Sie auch den Befehl **firewall-cmd --reload** ausführen, der die aktuelle permanente Konfiguration liest und als neue Laufzeitkonfiguration anwendet. Für viele der aufgeführten Befehle gilt die Option **--zone=ZONE**, um zu bestimmen, auf welche Zone sie sich auswirken. Wenn eine Netzmaske erforderlich ist, verwenden Sie die CIDR-Notation, beispielsweise 192.168.1/24.

firewall-cmd-Befehle	Erläuterung
--get-default-zone	Abrufen der aktuellen Standardzone
--set-default-zone=ZONE	Festlegen der Standardzone. Diese Einstellung bezieht sich sowohl auf die Laufzeit- als auch auf die permanente Konfiguration.
--get-zones	Auflisten aller verfügbaren Zonen
--get-active-zones	Auflisten der zurzeit verwendeten Zonen (an die eine Schnittstelle oder Quelle angebunden ist) mit den zugehörigen Schnittstellen- und Quellinformationen.
--add-source=CIDR [--zone=ZONE]	Weiterleiten des Datenverkehrs von der IP-Adresse oder dem Netzwerk/der Netzmaske an die angegebene Zone. Wenn die Option --zone= nicht angegeben ist, wird die Standardzone verwendet.
--remove-source=CIDR [--zone=ZONE]	Entfernen der Regel, die den gesamten Datenverkehr aus der Zone leitet, die von der IP-Adresse oder dem Netzwerk/Netzmaskennetzwerk stammt. Wenn die Option --zone= nicht angegeben ist, wird die Standardzone verwendet.
--add-interface=INTERFACE [--zone=ZONE]	Weiterleiten des Datenverkehrs von INTERFACE an die angegebene Zone. Wenn die Option --zone= nicht angegeben ist, wird die Standardzone verwendet.
--change-interface=INTERFACE [--zone=ZONE]	Verknüpfen der Schnittstelle mit ZONE , anstatt mit der aktuellen Zone. Wenn die Option --zone= nicht angegeben ist, wird die Standardzone verwendet.
--list-all [--zone=ZONE]	Auflisten aller konfigurierten Schnittstellen, Quellen, Services und Ports für ZONE . Wenn die Option --zone= nicht angegeben ist, wird die Standardzone verwendet.
--list-all-zones	Abrufen sämtlicher Informationen für alle Zonen (Schnittstellen, Quellen, Ports, Dienste)

firewall-cmd-Befehle	Erläuterung
<code>--add-service=SERVICE [--zone=ZONE]</code>	Zulassen des Datenverkehrs für SERVICE . Wenn die Option <code>--zone=</code> nicht angegeben ist, wird die Standardzone verwendet.
<code>--add-port=PORT/PROTOCOL [--zone=ZONE]</code>	Zulassen des Datenverkehrs für Port(s) PORT/PROTOCOL . Wenn die Option <code>--zone=</code> nicht angegeben ist, wird die Standardzone verwendet.
<code>--remove-service=SERVICE [--zone=ZONE]</code>	Entfernen von SERVICE von der Liste der zulässigen Services für die Zone. Wenn die Option <code>--zone=</code> nicht angegeben ist, wird die Standardzone verwendet.
<code>--remove-port=PORT/PROTOCOL [--zone=ZONE]</code>	Löschen von PORT/PROTOCOL von der Liste der zulässigen Ports für die Zone. Wenn die Option <code>--zone=</code> nicht angegeben ist, wird die Standardzone verwendet.
<code>--reload</code>	Verwerfen der Laufzeitkonfiguration und Anwenden der dauerhaften Konfiguration.

Die folgenden Beispielbefehle legen die Standardzone auf **dmz** fest, weisen den gesamten Datenverkehr, der vom Netzwerk **192.168.0.0/24** stammt, der Zone **internal** zu und öffnen die Netzwerkports für den Service **mysql** in der Zone **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
```



Anmerkung

In Situationen, in denen die grundlegende Syntax von **firewalld** nicht ausreicht, können Sie mit der ausdrucksstärkeren Syntax von **rich-rules** komplexe Regeln schreiben. Selbst wenn die rich-rules-Syntax nicht ausreichend ist, können Sie auch *Direct Configuration*-Regeln verwenden, als eine Raw-**nft**-Syntax in Kombination mit **firewalld**-Regeln.

Diese erweiterten Möglichkeiten gehen aber über den Umfang dieses Kapitels hinaus.



Literaturhinweise

Manpages **firewall-cmd(1)**, **firewalld(1)**, **firewalld.zone(5)**, **firewalld.zones(5)** und **nft(8)**

► Angeleitete Übung

Verwalten von Server-Firewalls

In dieser Übung steuern Sie den Zugriff auf Systemservices, indem Sie die Firewall-Regeln des Systems mit **firewalld** anpassen.

Ergebnisse

Sie sollten in der Lage sein, Firewall-Regeln zu konfigurieren, um den Zugriff auf Services zu steuern.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab netsecurity-firewalls start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **servera** im Netzwerk erreichbar ist.

```
[student@workstation ~]$ lab netsecurity-firewalls start
```

- 1. Verwenden Sie auf **workstation** SSH, um sich bei **servera** als Benutzer **student** anzumelden. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Stellen Sie auf dem System **servera** sicher, dass die Pakete *httpd* und *mod_ssl* installiert sind. Diese Pakete stellen den Apache-Webserver bereit, den Sie mit einer Firewall schützen, sowie die erforderlichen Erweiterungen für den Webserver, um Inhalte über SSL bereitzustellen.

```
[student@servera ~]$ sudo yum install httpd mod_ssl  
[sudo] password for student: student  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

- 3. Erstellen Sie als der Benutzer **student** auf **servera** die Datei **/var/www/html/index.html**. Fügen Sie die folgende Textzeile hinzu: **I am servera**.

```
[student@servera ~]$ sudo bash -c \  
"echo 'I am servera.' > /var/www/html/index.html"
```

- 4. Starten und aktivieren Sie den Service **httpd** auf Ihrem System **servera**.

```
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/
lib/systemd/system/httpd.service.
```

- 5. Beenden Sie **servera**.

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

- 6. Versuchen Sie auf **workstation**, unter Verwendung des Klartextports **80/TCP** und des gekapselten SSL-Ports **443/TCP** auf Ihren Webserver auf **servera** zuzugreifen. Beide Versuche sollten fehlschlagen.

- 6.1. Der folgende Befehl muss fehlschlagen:

```
[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. Der folgende Befehl sollte ebenfalls fehlschlagen:

```
[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. Melden Sie sich als der Benutzer **student** bei **servera** an.

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

- 8. Stellen Sie auf **servera** sicher, dass der Service **nftables** maskiert und der Service **firewalld** aktiviert ist und ausgeführt wird.

- 8.1. Stellen Sie fest, ob der Service **nftables** den Status **masked** aufweist.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student: student
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled; vendor
  preset: disabled)
  Active: inactive (dead)
    Docs: man:nft(8)
```

Die Ergebnisse zeigen, dass **nftables** deaktiviert und inaktiv ist, aber nicht maskiert. Führen Sie den folgenden Befehl aus, um den Service zu maskieren.

```
[student@servera ~]$ sudo systemctl mask nftables  
Created symlink /etc/systemd/system/nftables.service → /dev/null.
```

- 8.2. Verifizieren Sie, dass der Service **nftables** den Status **masked** aufweist.

```
[student@servera ~]$ sudo systemctl status nftables  
● nftables.service  
  Loaded: masked (Reason: Unit nftables.service is masked.)  
  Active: inactive (dead)
```

- 8.3. Verifizieren Sie, dass der Status des Service **firewalld** aktiviert ist und ausgeführt wird.

```
[student@servera ~]$ sudo 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 2019-05-22 15:36:02 CDT; 5min ago  
    Docs: man:firewalld(1)  
  Main PID: 703 (firewalld)  
    Tasks: 2 (limit: 11405)  
   Memory: 29.8M  
  CGroup: /system.slice/firewalld.service  
          └─703 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --  
            nopid  
  
May 22 15:36:01 servera.lab.example.com systemd[1]: Starting firewalld - dynamic  
firewall daemon...  
May 22 15:36:02 servera.lab.example.com systemd[1]: Started firewalld - dynamic  
firewall daemon.
```

- 8.4. Beenden Sie **servera**.

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

- 9. Öffnen Sie auf **workstation** Firefox und melden Sie sich bei der Web Console an, die auf **servera** ausgeführt wird, um der Netzwerkzone **public** den Service **httpd** hinzuzufügen.

- 9.1. Öffnen Sie Firefox, und navigieren Sie zu **https://servera.lab.example.com:9090**, um auf die Web Console zuzugreifen. Akzeptieren Sie das von **servera** verwendete selbstsignierte Zertifikat, indem Sie eine Ausnahme hinzufügen.
- 9.2. Aktivieren Sie das Kontrollkästchen neben **Reuse my password for privileged tasks**, um Administratorberechtigungen zu gewährleisten. Melden Sie sich als Benutzer **student** mit dem Passwort **student** an.
- 9.3. Klicken Sie in der linken Navigationsleiste auf **Networking**.

- 9.4. Klicken Sie auf der Hauptseite für **Networking** auf den Link **Firewall**.
 - 9.5. Klicken Sie auf die Schaltfläche **Add Services...**, die sich im rechten oberen Bereich der Seite **Firewall** befindet.
 - 9.6. Blättern Sie auf der Benutzeroberfläche **Add Services** nach unten, oder verwenden Sie **Filter Services**, um das Kontrollkästchen neben dem Service **Secure WWW (HTTPS)** zu suchen und zu aktivieren.
 - 9.7. Klicken Sie auf die Schaltfläche **Add Services**, die sich im rechten unteren Bereich der Benutzeroberfläche **Add Services** befindet.
- 10. Wechseln Sie auf **workstation** zurück zum Terminal und verifizieren Sie Ihre Arbeit, indem Sie versuchen, den Inhalt des Webservers auf **servera** anzuzeigen.

- 10.1. Der folgende Befehl muss fehlschlagen:

```
[student@workstation ~]$ curl http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No route to host
```

- 10.2. Der folgende Befehl muss erfolgreich ausgeführt werden:

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
I am servera.
```



Anmerkung

Wenn Sie Firefox für die Verbindung zum Webserver verwenden, werden Sie zur Verifizierung des Hostzertifikats aufgefordert, damit die Firewall erfolgreich überwunden werden kann.

Beenden

Führen Sie auf **workstation** das Skript **lab netsecurity-firewalls finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab netsecurity-firewalls finish
```

Hiermit ist die angeleitete Übung beendet.

Steuern der SELinux-Portbezeichnung

Ziele

Nach Abschluss dieses Abschnitts sollten Sie verifizieren können, dass die Netzwerkports den richtigen SELinux-Typ aufweisen und die Services eine Bindung zu den Ports herstellen können.

Port-Bezeichnung in SELinux

Hinter SELinux steckt mehr als eine einfache Bezeichnung von Dateien und Prozessen. Der Datenverkehr im Netzwerk wird durch die SELinux-Richtlinie engmaschig erzwungen. Eine der Methoden in SELinux zur Steuerung des Netzwerkverkehrs ist die Bezeichnung von Netzwerk-Ports. So ist beispielsweise in der Richtlinie **targeted** der Port **22/TCP** mit der Bezeichnung **ssh_port_t** verknüpft. Den HTTP-Standardports **80/TCP** und **443/TCP** ist die Bezeichnung **http_port_t** zugeordnet.

Sobald ein Prozess einen Prozess überwachen möchte, überprüft SELinux, ob die mit dem Prozess (der Domain) verknüpfte Bezeichnung eine Bindung mit der Port-Bezeichnung eingehen kann. So kann verhindert werden, dass ein illegitimer Service Ports übernimmt, die ansonsten von legitimen Netzwerkservices genutzt würden.

Verwalten der Portbezeichnung in SELinux

Wenn Sie sich dafür entscheiden, einen Service an einem nicht standardmäßigen Port auszuführen, wird SELinux Datenverkehr mit großer Wahrscheinlichkeit blockieren. In diesem Fall müssen Sie die SELinux-Portbezeichnungen aktualisieren. In einigen Fällen wurde der Port gemäß der Richtlinie **targeted** bereits mit einem brauchbaren Typ bezeichnet. Da beispielsweise Port **8008/TCP** häufig für Webanwendungen verwendet wird, ist dieser Port bereits mit der Bezeichnung **http_port_t**, dem Standard-Port-Typ für den Webserver, versehen.

Auflisten von Portbezeichnungen

Führen Sie den Befehl **semanage port -l** aus, um eine Übersicht von allen aktuellen Portbezeichnungszuweisungen abzurufen. Die Option **-l** listet alle aktuellen Zuweisungen in der folgenden Form auf:

```
port_label_t      tcp|udp      comma-separated, list, of, ports
```

Beispieldaten:

```
[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...
```

Führen Sie den Befehl **grep** aus, um die Suche einzuschränken:

```
[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
```

Eine Portbezeichnung kann zweimal in der Ausgabe vorkommen: einmal bei TCP und einmal bei UDP.

Verwalten von Portbezeichnungen

Führen Sie den Befehl **semanage** aus, um neue Portbezeichnungen zuzuweisen, Portbezeichnungen zu entfernen oder vorhandene zu ändern.



Wichtig

Die meisten Standardservices, die in der Linux-Distribution verfügbar sind, bieten ein SELinux-Richtlinienmodul, das Bezeichnungen für Ports festlegt. Sie können die Beschriftungen an diesen Ports nicht mit **semanage** ändern. Um diese zu ändern, müssen Sie das Richtlinienmodul ersetzen. Das Verfassen und Generieren von Richtlinienmodulen wird in diesem Kurs nicht behandelt.

Mit der folgenden Syntax fügen Sie einer vorhandenen Portbezeichnung (Typ) einen Port hinzu. Durch **-a** wird eine neue Portbezeichnung hinzugefügt. Durch **-t** wird der Typ angegeben. Durch **-p** wird das Protokoll angegeben.

```
[root@host ~]# semanage port -a -t port_label -p tcp|udp PORTNUMBER
```

So lassen Sie beispielsweise zu, dass ein **gopher**-Service Port **71/TCP** überwacht:

```
[root@host~]# semanage port -a -t gopher_port_t -p tcp 71
```

Wenn lokale Änderungen an der Standardrichtlinie angezeigt werden sollen, können Administratoren dem Befehl **semanage** die Option **-C** hinzufügen.

```
[root@host~]# semanage port -l -C
SELinux Port Type          Proto    Port Number
gopher_port_t               tcp      71
```



Anmerkung

Die Richtlinie **targeted** ist im Lieferumfang zahlreicher Porttypen enthalten.

Servicespezifische SELinux-Manpages im Paket *selinux-policy-doc* enthalten eine Dokumentation zu SELinux-Typen, booleschen Werten und Porttypen. Wenn diese Manpages auf Ihrem System noch nicht installiert sind, dann gehen Sie wie folgt vor:

```
[root@host ~]# yum -y install selinux-policy-doc
[root@host ~]# man -k _selinux
```

Entfernen von Portbezeichnungen

Zum Entfernen einer benutzerdefinierten Portbezeichnung wird die gleiche Syntax wie zum Hinzufügen einer Bezeichnung verwendet – mit dem Unterschied, dass beim Hinzufügen die Option **-a** (Add) und beim Löschen die Option **-d** (Delete) erforderlich ist.

Um die Bindung zwischen Port **71/TCP** und **gopher_port_t** aufzuheben, gehen Sie beispielsweise wie folgt vor:

```
[root@host ~]# semanage port -d -t gopher_port_t -p tcp 71
```

Ändern von Portbindungen

Um eine Portbindung zu ändern, vielleicht weil sich die Anforderungen geändert haben, verwenden Sie die Option **-m** (Ändern). Das ist effizienter, als die alte Bindung zu löschen und eine neue zu erstellen.

So können Administratoren beispielsweise mit dem folgenden Befehl Port **71/TCP** von **gopher_port_t** in **http_port_t** ändern:

```
[root@server ~]# semanage port -m -t http_port_t -p tcp 71
```

Zeigen Sie wie zuvor die Änderung mit dem Befehl **semanage** an.

```
[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
```



Literaturhinweise

Manpages **semanage(8)**, **semanage-port(8)** und ***_selinux(8)**

► Angeleitete Übung

Steuern der SELinux-Portbezeichnung

In dieser praktischen Übung konfigurieren Sie Ihr System **servera** so, dass der HTTP-Zugriff an einem Nicht-Standard-Port zugelassen wird.

Ergebnisse:

Sie konfigurieren einen Webserver, der auf **servera** ausgeführt wird und Inhalte über einen Nicht-Standard-Port bereitstellt.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab netsecurity-ports start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Er installiert zudem den **httpd**-Service und konfiguriert die Firewall auf **servera** so, dass HTTP-Verbindungen zugelassen werden.

```
[student@workstation ~]$ lab netsecurity-ports start
```

Ihre Organisation stellt eine neue individuelle Webanwendung bereit. Die Webanwendung wird auf einem Nicht-Standard-Port ausgeführt, in diesem Fall **82/TCP**.

Einer Ihrer Junior-Administratoren hat die Anwendung auf Ihrem **servera** bereits konfiguriert. Auf den Webserver-Inhalt kann jedoch nicht zugegriffen werden.

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum **root**-Benutzer zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Versuchen Sie, das Problem mit dem Webinhalt zu beheben, indem Sie den **httpd**-Service neu starten.
- 3.1. Führen Sie den Befehl **systemctl** aus, um **httpd.service** neu zu starten. Dieser Befehl wird voraussichtlich fehlschlagen.

```
[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.
```

- 3.2. Führen Sie den Befehl **systemctl status -l** aus, um den Status des **httpd**-Services anzuzeigen. Beachten Sie den **permission denied**-Fehler.

```
[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.
```

- 3.3. Führen Sie den Befehl **sealert** aus, um zu überprüfen, ob SELinux **httpd** daran hindert, eine Verbindung zum **Port 82/TCP** herzustellen.

```
[root@servera ~]# 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 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...
```

- 4. Konfigurieren Sie SELinux so, dass **httpd** eine Bindung mit Port **82/TCP** eingehen kann, und starten Sie anschließend den Service **httpd.service** neu.

- 4.1. Suchen Sie mit dem Befehl **semanage** einen angemessenen Port-Typ für Port **82/TCP**.

```
[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
```

http_port_t enthält die HTTP-Standardports **80/TCP** und **443/TCP**. Dies ist der richtige Porttyp für den Webserver.

- 4.2. Führen Sie den Befehl **semanage** aus, um dem Port **82/TCP** den Typ **http_port_t** zuzuweisen.

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
```

- 4.3. Führen Sie den Befehl **systemctl** aus, um den **httpd.service**-Service neu zu starten. Dieser Befehl sollte erfolgreich sein.

```
[root@servera ~]# systemctl restart httpd.service
```

- 5. Prüfen Sie, ob Sie jetzt auf den Webserver auf Port **82/TCP** zugreifen können. Führen Sie den Befehl **curl** aus, um auf den Webservice von **servera** zuzugreifen.

```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
```

- 6. Prüfen Sie in einem anderen Terminal, ob Sie über **workstation** auf den neuen Webservice zugreifen können. Führen Sie den Befehl **curl** aus, um auf den Webservice von **workstation** zuzugreifen.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
curl: (7) Failed to connect to servera.example.com:82; No route to host
```

Dieser Fehler bedeutet, dass Sie über **workstation** immer noch keine Verbindung zum Webservice herstellen können.

- 7. Öffnen Sie auf **servera** Port **82/TCP** auf der Firewall.

- 7.1. Führen Sie den Befehl **firewall-cmd** aus, um den Port **82/TCP** in der permanenten Konfiguration für die Standardzone auf der Firewall auf **servera** zu öffnen.

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp  
success
```

- 7.2. Aktivieren Sie Ihre Firewall-Änderungen auf **servera**.

```
[root@servera ~]# firewall-cmd --reload  
success
```

- 8. Führen Sie den Befehl **curl** aus, um auf den Webservice von **workstation** zuzugreifen.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82  
Hello
```

- 9. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab netsecurity-ports finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab netsecurity-ports finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Verwalten der Netzwerksicherheit

Leistungscheckliste

In dieser praktischen Übung konfigurieren Sie die Firewall- und SELinux-Einstellungen, um den Zugriff auf mehrere Webserver zu ermöglichen, die auf **serverb** ausgeführt werden.

Ergebnisse

Sie sollten die Firewall- und SELinux-Einstellungen auf einem Webserverhost konfigurieren können.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab netsecurity-review start** aus.

Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **serverb** im Netzwerk erreichbar ist.

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

Ihr Unternehmen hat beschlossen, eine neue Webanwendung zu verwenden. Die Anwendung ist an den Ports **80/TCP** und **1001/TCP** empfangsbereit. Der Port **22/TCP** für den **ssh**-Zugriff muss ebenfalls verfügbar sein. Alle vorgenommenen Änderungen müssen nach einem Neustart weiterhin bestehen.

Verwenden Sie bei Aufforderung durch **sudo student** als Passwort.

Wichtig: Die in der Red Hat Online Learning-Umgebung verwendete grafische Oberfläche benötigt Port **5900/TCP**, damit sie auch weiterhin verfügbar ist. Dieser Port trägt auch den Servicenamen **vnc-server**. Wenn Sie sich versehentlich auf Ihrem **serverb** aussperren, können Sie den vorherigen Zustand entweder durch Verwenden von **ssh** auf Ihrem **serverb**-Rechner über den **workstation**-Rechner wiederherstellen, oder indem Sie den **serverb**-Rechner zurücksetzen. Wenn Sie sich für das Zurücksetzen des **serverb**-Rechners entscheiden, müssen Sie die Setup-Skripts für diese praktische Übung nochmals ausführen. Die Konfiguration auf Ihrem Rechner enthält bereits eine benutzerdefinierte Zone namens **ROL**, die diese Ports öffnet.

1. Testen Sie auf **workstation** den Zugriff auf den Standardwebserver unter `http://serverb.lab.example.com` und auf den virtuellen Host unter `http://serverb.lab.example.com:1001`.
2. Melden Sie sich bei **serverb** an, um festzustellen, was den Zugriff auf die Webserver verhindert.
3. Konfigurieren Sie SELinux so, dass der Service **httpd** Port **1001/TCP** überwachen kann.
4. Testen Sie auf **workstation** den Zugriff auf den Standardwebserver unter `http://serverb.lab.example.com` und auf den virtuellen Host unter `http://serverb.lab.example.com:1001`.

5. Melden Sie sich bei **serverb** an, um festzustellen, ob der Firewall die richtigen Ports zugeordnet sind.
6. Fügen Sie der dauerhaften Konfiguration für die Netzwerkzone **public** den Port **1001/TCP** hinzu. Bestätigen Sie Ihre Konfiguration.
7. Bestätigen Sie auf **workstation**, dass der standardmäßige Webserver unter **serverb.lab.example.com** den Text **SERVER B** zurückgibt und dass der virtuelle Host unter **serverb.lab.example.com:1001** den Text **VHOST 1** zurückgibt.

Bewertung

Führen Sie auf **workstation** den Befehl **lab netsecurity-review grade** aus, um den Erfolg dieser praktischen Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab netsecurity-review finish** aus, um diese Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

► Lösung

Verwalten der Netzwerksicherheit

Leistungscheckliste

In dieser praktischen Übung konfigurieren Sie die Firewall- und SELinux-Einstellungen, um den Zugriff auf mehrere Webserver zu ermöglichen, die auf **serverb** ausgeführt werden.

Ergebnisse

Sie sollten die Firewall- und SELinux-Einstellungen auf einem Webserverhost konfigurieren können.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab netsecurity-review start** aus.

Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob der Host **serverb** im Netzwerk erreichbar ist.

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

Ihr Unternehmen hat beschlossen, eine neue Webanwendung zu verwenden. Die Anwendung ist an den Ports **80/TCP** und **1001/TCP** empfangsbereit. Der Port **22/TCP** für den **ssh**-Zugriff muss ebenfalls verfügbar sein. Alle vorgenommenen Änderungen müssen nach einem Neustart weiterhin bestehen.

Verwenden Sie bei Aufforderung durch **sudo student** als Passwort.

Wichtig: Die in der Red Hat Online Learning-Umgebung verwendete grafische Oberfläche benötigt Port **5900/TCP**, damit sie auch weiterhin verfügbar ist. Dieser Port trägt auch den Servicenamen **vnc-server**. Wenn Sie sich versehentlich auf Ihrem **serverb** aussperren, können Sie den vorherigen Zustand entweder durch Verwenden von **ssh** auf Ihrem **serverb**-Rechner über den **workstation**-Rechner wiederherstellen, oder indem Sie den **serverb**-Rechner zurücksetzen. Wenn Sie sich für das Zurücksetzen des **serverb**-Rechners entscheiden, müssen Sie die Setup-Skripts für diese praktische Übung nochmals ausführen. Die Konfiguration auf Ihrem Rechner enthält bereits eine benutzerdefinierte Zone namens **ROL**, die diese Ports öffnet.

1. Testen Sie auf **workstation** den Zugriff auf den Standardwebserver unter <http://serverb.lab.example.com> und auf den virtuellen Host unter <http://serverb.lab.example.com:1001>.
 - 1.1. Testen Sie den Zugriff auf den <http://serverb.lab.example.com>-Webserver. Der Test schlägt derzeit fehl. Letztlich sollte der Webserver **SERVER B** zurückgeben.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
curl: (7) Failed to connect to serverb.lab.example.com port 80: Connection refused
```

- 1.2. Testen Sie den Zugriff auf den virtuellen Host `http://serverb.lab.example.com:1001`. Der Test schlägt derzeit fehl. Letztlich sollte der virtuelle Host **VHOST 1** zurückgeben.

```
[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. Melden Sie sich bei **serverb** an, um festzustellen, was den Zugriff auf die Webserver verhindert.
- 2.1. Öffnen Sie auf **workstation** als Benutzer **student** eine SSH-Sitzung zu **serverb**. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2.2. Stellen Sie fest, ob der Service **httpd** aktiv ist.

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

- 2.3. Aktivieren und starten Sie den Service **httpd**. Der Service **httpd** kann nicht gestartet werden.

```
[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 -xe" for details.
```

- 2.4. Untersuchen Sie die Gründe dafür, weshalb der Service **httpd.service** nicht gestartet werden konnte.

```
[student@serverb ~]$ systemctl status 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 Thu 2019-04-11 19:25:36 CDT; 19s ago
    Docs: man:httpd.service(8)
   Process: 9615 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
   Main PID: 9615 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."

Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Starting The Apache HTTP
Server...
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission denied:
AH00072: make_sock: could not bind to address [::]:1001
```

Kapitel 11 | Verwalten der Netzwerksicherheit

```
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission denied:  
AH000072: make_sock: could not bind to address 0.0.0.0:1001  
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: no listening sockets  
available, shutting down  
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: AH00015: Unable to open logs  
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service: Main process  
exited, code=exited, status=1/FAILURE  
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service: Failed with  
result 'exit-code'.  
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Failed to start The Apache  
HTTP Server.
```

- 2.5. Führen Sie den Befehl **sealert** aus, um zu überprüfen, ob SELinux den Service **httpd** daran hindert, eine Verbindung zum Port **1001/TCP** herzustellen.

```
[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. Konfigurieren Sie SELinux so, dass der Service **httpd** Port **1001/TCP** überwachen kann.

- 3.1. Führen Sie den Befehl **semanage** aus, um den richtigen Porttyp zu finden.

```
[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. Führen Sie den Befehl **semanage** aus, um den Port **1001/TCP** an den Typ **http_port_t** zu binden.

```
[student@serverb ~]$ sudo semanage port -a -t http_port_t -p tcp 1001  
[student@serverb ~]$
```

- 3.3. Bestätigen Sie, dass der Port **1001/TCP** an den Porttyp **http_port_t** gebunden ist.

```
[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. Aktivieren und starten Sie den Service **httpd**.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
```

- 3.5. Verifizieren Sie den aktiven Status des Services **httpd**.

```
[student@serverb ~]$ systemctl is-active httpd; systemctl is-enabled httpd  
active  
enabled
```

- 3.6. Beenden Sie **serverb**.

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

4. Testen Sie auf **workstation** den Zugriff auf den Standardwebserver unter `http://serverb.lab.example.com` und auf den virtuellen Host unter `http://serverb.lab.example.com:1001`.

- 4.1. Testen Sie den Zugriff auf den `http://serverb.lab.example.com`-Webserver. Der Webserver sollte **SERVER B** zurückgeben.

```
[student@workstation ~]$ curl http://serverb.lab.example.com  
SERVER B
```

- 4.2. Testen Sie den Zugriff auf den virtuellen Host `http://serverb.lab.example.com:1001`. Der Test schlägt weiterhin fehl.

```
[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. Melden Sie sich bei **serverb** an, um festzustellen, ob der Firewall die richtigen Ports zugeordnet sind.

- 5.1. Melden Sie sich von **workstation** aus bei **serverb** als Benutzer **student** an.

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

- 5.2. Verifizieren Sie, dass die standardmäßige Firewall-Zone auf **public** gesetzt ist.

```
[student@serverb ~]$ firewall-cmd --get-default-zone  
public
```

Kapitel 11 | Verwalten der Netzwerksicherheit

- 5.3. Wenn im vorherigen Schritt nicht **public** als die Standardzone zurückgegeben wurde, korrigieren Sie dies mit dem folgenden Befehl:

```
[student@serverb ~]$ sudo firewall-cmd --set-default-zone public
```

- 5.4. Bestimmen Sie die offenen Ports, die in der Netzwerkzone **public** aufgelistet sind.

```
[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 dhcpcv6-client http ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

6. Fügen Sie der dauerhaften Konfiguration für die Netzwerkzone **public** den Port **1001/TCP** hinzu. Bestätigen Sie Ihre Konfiguration.

- 6.1. Fügen Sie der Netzwerkzone **public** den Port **1001/TCP** hinzu.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public \
--add-port=1001/tcp
success
```

- 6.2. Laden Sie die Firewall-Konfiguration neu.

```
[student@serverb ~]$ sudo firewall-cmd --reload
success
```

- 6.3. Bestätigen Sie Ihre Konfiguration.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: cockpit dhcpcv6-client http ssh
  ports: 1001/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
```

```
icmp-blocks:  
rich rules:
```

6.4. Beenden Sie **serverb**.

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

7. Bestätigen Sie auf **workstation**, dass der standardmäßige Webserver unter **serverb.lab.example.com** den Text **SERVER B** zurückgibt und dass der virtuelle Host unter **serverb.lab.example.com:1001** den Text **VHOST 1** zurückgibt.

7.1. Testen Sie den Zugriff auf den `http://serverb.lab.example.com`-Webserver.

```
[student@workstation ~]$ curl http://serverb.lab.example.com  
SERVER B
```

7.2. Testen Sie den Zugriff auf den virtuellen Host `http://serverb.lab.example.com:1001`.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001  
VHOST 1
```

Bewertung

Führen Sie auf **workstation** den Befehl **lab netsecurity-review grade** aus, um den Erfolg dieser praktischen Übung zu bestätigen.

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

Beenden

Führen Sie auf **workstation** das Skript **lab netsecurity-review finish** aus, um diese Übung zu beenden.

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

Damit ist die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Mit dem **netfilter**-Subsystem können die Kernel-Module jedes einzelne Paket, das das System durchläuft, prüfen. Alle eingehenden, ausgehenden oder weitergeleiteten Netzwerkpakete werden geprüft.
- Die Verwendung von **firewalld** hat eine vereinfachte Verwaltung, indem der gesamte Netzwerdatenverkehr in Zonen klassifiziert wird. Jede Zone besitzt ihre eigene Liste von Ports und Services. Die Zone **public** ist als Standardzone festgelegt.
- Der Service **firewalld** enthält standardmäßig einige vordefinierte Services. Sie können mit dem Befehl **firewall-cmd --get-services** aufgelistet werden.
- Der Datenverkehr im Netzwerk wird durch die SELinux-Richtlinie engmaschig gesteuert. Netzwerkports sind gekennzeichnet. Beispielsweise ist dem Port **22/TCP** die Bezeichnung **ssh_port_t** zugeordnet. Wenn ein Prozess einen Port überwachen möchte, überprüft SELinux, ob die verknüpfte Bezeichnung eine Bindung mit der Portbezeichnung eingehen kann.
- Der Befehl **semanage** wird verwendet, um Bezeichnungen hinzuzufügen, zu löschen und zu ändern.

Kapitel 12

Installation von Red Hat Enterprise Linux

Ziel

Installieren Sie Red Hat Enterprise Linux auf Servern und virtuellen Rechnern.

Ziele

- Installieren von Red Hat Enterprise Linux auf einem Server.
- Automatisieren des Installationsvorgangs mit Kickstart.
- Installieren eines virtuellen Rechners auf dem Red Hat Enterprise Linux-Server mit Cockpit.

Abschnitte

- Installieren von Red Hat Enterprise Linux (und angeleitete Übung)
- Automatisieren der Installation mit Kickstart (und angeleitete Übung)
- Installieren und Konfigurieren virtueller Rechner (und Test)

Praktische Übung

Installieren von Red Hat Enterprise Linux

Installieren von Red Hat Enterprise Linux

Ziele

In diesem Abschnitt wird beschrieben, wie Red Hat Enterprise Linux auf einem Server installiert wird.

Auswählen von Installationsmedien

Red Hat bietet verschiedene Arten von Installationsmedien, die Sie mit Ihrem aktiven Abonnement von der Customer Portal-Website herunterladen können.

- Eine binäre Image-Datei im ISO 9660-Format, die *Anaconda*, das Red Hat Enterprise Linux-Installationsprogramm und die BaseOS- und AppStream-Paket-Repositorys enthält. Diese Repositorys enthalten die Pakete, die zum Abschließen der Installation ohne zusätzliches Material erforderlich sind.
- Eine kleinere ISO-Start-Image-Datei, in der *Anaconda* enthalten ist, die ein konfiguriertes Netzwerk erfordert, um auf über HTTP, FTP oder NFS zur Verfügung gestellte Paket-Repositorys zuzugreifen.
- Ein QCOW2-Image mit einer vorgefertigten System-Disk, die als virtueller Rechner in Cloud- oder virtuellen Unternehmensumgebungen bereitgestellt werden kann. QCOW2 ist das standardmäßige Image-Format, das von Red Hat mit KVM-basierter Virtualisierung verwendet wird.

Red Hat stellt Installationsmedien für vier unterstützte Prozessorarchitekturen bereit: x86 64-Bit (AMD und Intel), IBM Power Systems (Little Endian), IBM Z und ARM 64-Bit.

Erstellen Sie nach dem Download bootfähige Installationsmedien entsprechend den Anweisungen unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_a_standard_rhel_installation/index#making-media_preparing-for-your-installation.

Erstellen von Images mit Composer

Composer ist ein neues, in RHEL 8 verfügbares Tool. In speziellen Anwendungsfällen ermöglicht *Composer* Administratoren die Erstellung benutzerdefinierter System-Images für die Bereitstellung auf Cloud-Plattformen oder virtuellen Umgebungen.

Composer verwendet die grafische Web Console Cockpit. *Composer* kann auch durch Ausführen des Befehls **composer -cli** an einer Befehlszeile aufgerufen werden.

Manuelle Installation von Red Hat Enterprise Linux

Mit der binären DVD oder der ISO-Startdatei können Administratoren ein neues RHEL-System auf einem Bare-Metal-Server oder auf einem virtuellen Rechner installieren. Das *Anaconda*-Programm unterstützt zwei Installationsmethoden:

- Die manuelle Installation interagiert mit dem Benutzer, um zu fragen, wie *Anaconda* das System installieren und konfigurieren soll.

- Die automatisierte Installation verwendet eine *Kickstart*-Datei, die Anaconda anweist, wie das System installiert werden soll. In einem späteren Abschnitt werden Kickstart-Installationen ausführlicher beschrieben.

Installieren von RHEL mit der grafischen Oberfläche

Wenn Sie das System über die binäre DVD oder über die ISO-Startdatei starten, wird Anaconda als grafische Anwendung gestartet.

Auf dem Bildschirm **Welcome to Red Hat Enterprise Linux 8** wählen Sie die Sprache, in der Sie den Installationsvorgangs durchführen möchten. Dadurch wird auch die Standardsprache des Systems nach der Installation festgelegt. Einzelne Benutzer können nach der Installation die bevorzugte Sprache auswählen.

Anaconda zeigt das Fenster **Installation Summary** an. Hierbei handelt es sich um den zentralen Ort zum Anpassen der Parameter vor Beginn der Installation.

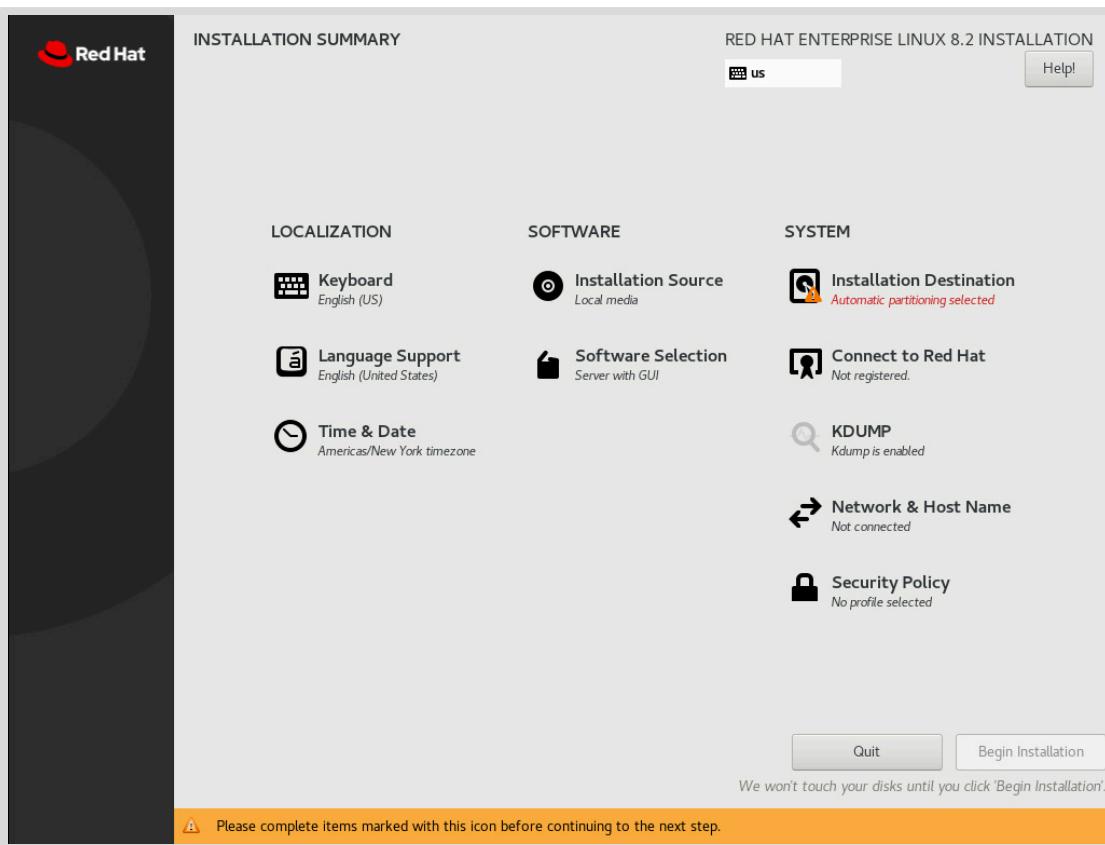


Abbildung 12.1: Installation Summary Window

Konfigurieren Sie in diesem Fenster die Installationsparameter, indem Sie die Symbole in beliebiger Reihenfolge auswählen. Wählen Sie einen Parameter aus, den Sie anzeigen oder bearbeiten möchten. Klicken Sie in einem beliebigen Element auf **Done**, um zu diesem zentralen Bildschirm zurückzukehren.

Anaconda kennzeichnet obligatorische Elemente mit einem Warnsymbol und einer Meldung. Die orangefarbene Statusleiste am unteren Rand des Bildschirms erinnert Sie daran, dass vor dem Beginn der Installation obligatorische Elemente abgeschlossen werden müssen.

Ergänzen Sie nach Bedarf die folgenden Punkte:

- **Keyboard:** Fügen Sie zusätzliche Tastaturlayouts hinzu.
- **Language Support:** Wählen Sie zusätzliche Sprachen für die Installation aus.
- **Time & Date:** Wählen Sie den Ort des Systems aus, indem Sie auf die interaktive Karte klicken oder ihn aus der Dropdown-Liste auswählen. Geben Sie die lokale Zeitzone an, auch wenn Sie *Network Time Protocol (NTP)* verwenden.
- **Installation Source:** Geben Sie den Speicherort des Quellpakets an, den Anaconda für die Installation benötigt. Bei Verwendung der binären DVD bezieht sich das Quelfeld der Installation bereits auf die DVD.
- **Software Selection:** Wählen Sie die zu installierende Basisumgebung und zusätzliche Add-ons aus. In der Umgebung **Minimal Install** werden nur die Pakete installiert, die zum Ausführen von Red Hat Enterprise Linux erforderlich sind.
- **Installation Destination:** Wählen und partitionieren Sie die Disks, auf denen Red Hat Enterprise Linux installiert wird. Diese Aufgabe setzt voraus, dass der Administrator die Partitionstabellen und die Auswahlkriterien von Dateisystemen versteht. Das Standardoptionsfeld für eine automatische Partitionierung weist die ausgewählten Storage-Geräte unter Einbeziehung des gesamten verfügbaren Speicherplatzes zu.
- **Verbindung zu Red Hat:** Registrieren Sie das System mit Ihrem Red Hat-Konto und wählen Sie den *Systemzweck* aus. Mit der Funktion für den Systemzweck kann der Registrierungsprozess das am besten geeignete Abonnement für das System automatisch anhängen. Um das System zu registrieren, müssen Sie zunächst über das Symbol **Network & Host Name** eine Verbindung zum Netzwerk herstellen.
- **KDUMP:** Die Funktion für Crash-Dumps des Kernels *KDUMP* sammelt Informationen über den Status des Systemspeichers, wenn der Kernel abstürzt. Red Hat-Techniker können eine *kdump*-Datei analysieren, um die Ursache eines Absturzes zu ermitteln. Verwenden Sie dieses Anaconda-Element, um *kdump* zu aktivieren oder zu deaktivieren.
- **Network & Host Name:** Im linken Bereich werden die erkannten Netzwerkverbindungen angezeigt. Wählen Sie eine Verbindung aus, um die zugehörigen Details anzuzeigen. Klicken Sie auf **Configure**, um die ausgewählte Netzwerkverbindung zu konfigurieren.
- **Security Policy:** Durch Aktivierung eines Sicherheitsrichtlinienprofils, beispielsweise des Profils *Payment Card Industry Data Security Standard (PCI DSS)*, wendet Anaconda während der Installation Einschränkungen und Empfehlungen an, die vom ausgewählten Profil definiert werden.

Klicken Sie nach Abschluss der Installationskonfiguration und dem Lösen aller Warnungen auf **Begin Installation**. Beim Klicken auf **Quit** wird die Installation abgebrochen, ohne Änderungen am System vorzunehmen.

Nehmen Sie während der Installation des Systems die folgenden Einstellungen vor, wenn die entsprechenden Elemente angezeigt werden:

- **Root Password:** Das Installationsprogramm fordert Sie zur Einrichtung eines **Root**-Passworts auf. Die letzte Phase des Installationsvorgangs beginnt erst, wenn ein **Root**-Passwort festgelegt wurde.
- **User Creation:** Erstellen Sie ein optionales Nicht-Root-Konto. Es wird empfohlen, ein lokales Konto für die allgemeine Verwendung zu besitzen. Sie können Konten auch erstellen, nachdem die Installation abgeschlossen ist.

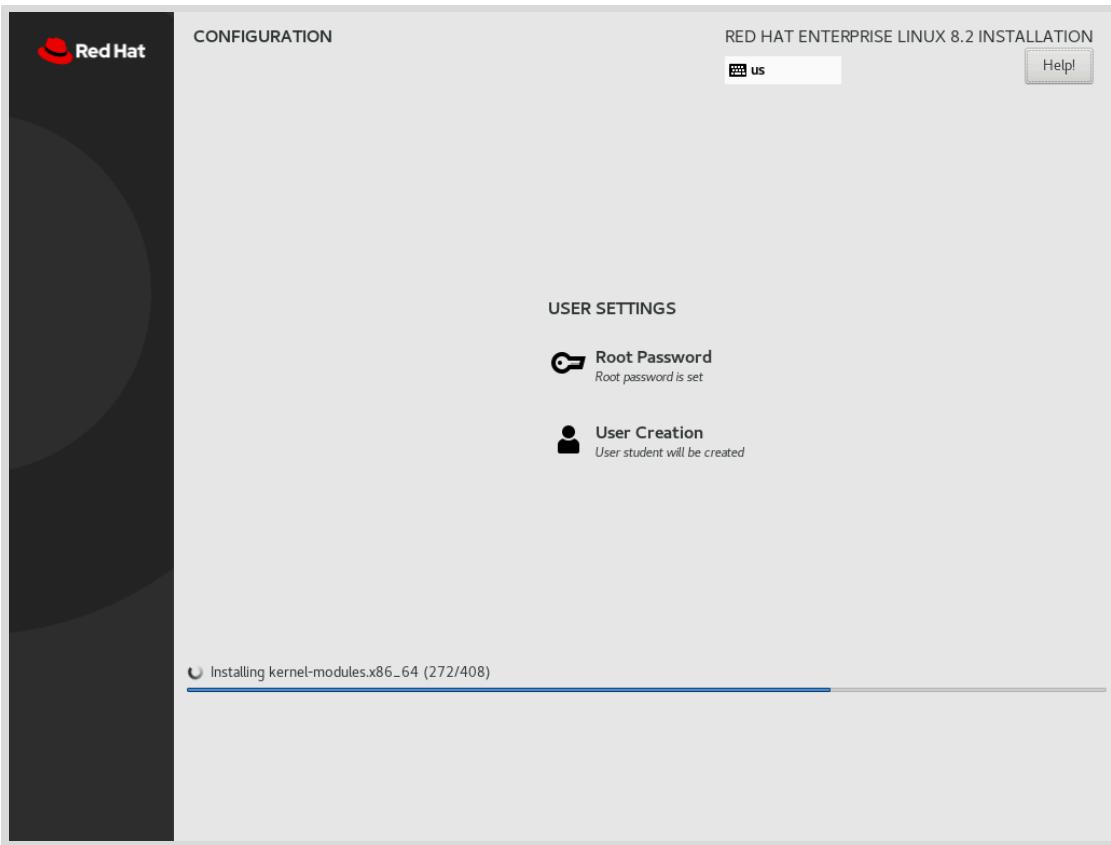


Abbildung 12.2: Festlegen des Root-Passworts und Erstellen eines Benutzers

Klicken Sie nach Abschluss der Installation auf **Reboot**. Anaconda zeigt den Bildschirm **Initial Setup** an, wenn ein grafischer Desktop installiert wurde. Akzeptieren Sie die Lizenzinformationen, und registrieren Sie das System optional beim Abonnement-Manager. Sie können die Systemregistrierung überspringen und später durchführen.

Fehlersuche bei der Installation

Während einer Red Hat Enterprise Linux 8-Installation stellt Anaconda zwei virtuelle Konsolen bereit: Die erste verfügt über fünf Fenster, die vom Software-Terminal-Multiplexer **tmux** bereitgestellt werden. Sie können mit **Strg+Alt+F1** auf diese Konsole zugreifen. Die zweite virtuelle Konsole, die standardmäßig angezeigt wird, zeigt die grafische Benutzeroberfläche von Anaconda. Sie können mit **Strg+Alt+F6** darauf zugreifen.

In der ersten virtuellen Konsole stellt **tmux** eine Shell-Eingabeaufforderung im zweiten Fenster bereit. Sie können damit Befehle eingeben, um das System zu überprüfen und Fehler zu beheben, während die Installation fortgesetzt wird. Die anderen Fenster bieten Diagnosemeldungen, Protokolle und andere Informationen.

In der folgenden Tabelle sind die Tastenkombinationen für den Zugriff auf die virtuellen Konsolen und die **tmux**-Fenster aufgeführt. Für **tmux** werden die Tastenkombinationen in zwei Schritten ausgeführt: Drücken und Loslassen von **Strg+B** und anschließendes Drücken der Zifferntaste des Fensters, auf das Sie zugreifen möchten. Mit **tmux** können Sie auch **Alt+Tab** verwenden, um den aktuellen Fokus zwischen den Fenstern zu drehen.

Tastenfolge	Inhalt
Strg+Alt+F1	Greifen Sie auf den tmux -Terminal-Multiplexer zu.

Tastenfolge	Inhalt
Strg+B 1	Greifen Sie in tmux auf die Hauptinformationsseite für den Installationsvorgang zu.
Strg+B 2	Stellen Sie in tmux eine Root-Shell bereit. Anaconda speichert die Installationsprotokolldateien im Verzeichnis /tmp .
Strg+B 3	Zeigen Sie in tmux die Inhalte der Datei /tmp/anaconda.log an.
Strg+B 4	Zeigen Sie in tmux die Inhalte der Datei /tmp/storage.log an.
Strg+B 5	Zeigen Sie in tmux die Inhalte der Datei /tmp/program.log an.
Strg+Alt+F6	Rufen Sie die grafische Benutzeroberfläche von Anaconda auf.



Anmerkung

Für die Kompatibilität mit früheren Red Hat Enterprise Linux-Versionen stellen die virtuellen Konsolen von **Strg+Alt+F2** bis **Strg+Alt+F5** auch die Root-Shells während der Installation dar.



Literaturhinweise

Weitere Informationen finden Sie im *Performing a standard RHEL installation Guide* unter
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_a_standard_rhel_installation/index

► Angeleitete Übung

Installieren von Red Hat Enterprise Linux

In dieser Übung installieren Sie einen Ihrer Server mit einer minimalen Installation von Red Hat Enterprise Linux neu.

Ergebnisse

Sie sollten Red Hat Enterprise Linux 8 manuell installieren können.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab installing-install start** aus. Dieser Befehl führt ein Startskript aus, um zu bestimmen, ob der Rechner **servera** im Netzwerk erreichbar ist. Darüber hinaus fügt er einen neuen Eintrag im GRUB2-Menü hinzu, um **servera** auf dem Installationsmedium zu starten.

```
[student@workstation ~]$ lab installing-install start
```

Anweisungen

- ▶ 1. Greifen Sie auf die Konsole **servera** zu, und starten Sie das System auf dem Installationsmedium neu.
 - 1.1. Suchen Sie in Ihrer Kursumgebung nach dem Symbol für die **servera**-Konsole. Öffnen Sie die Konsole.
 - 1.2. Geben Sie für den Neustart **Strg+Alt+Entf** über die entsprechende Taste oder den entsprechenden virtuellen oder Menüeintrag in Ihr System ein.
 - 1.3. Wenn das Bootloader-Menü angezeigt wird, wählen Sie **Install Red Hat Enterprise Linux 8** aus.
 - 1.4. Warten Sie auf das Sprachauswahlfenster.
- ▶ 2. Behalten Sie die Sprache standardmäßig bei und klicken Sie auf **Continue**.
- ▶ 3. Verwenden Sie die automatische Partitionierung für die Disk **/dev/vda**.
 - 3.1. Klicken Sie auf **Installation Destination**.
 - 3.2. Klicken Sie auf die erste Disk, **vda**, um sie auszuwählen. Klicken Sie auf **Done**, um die Standardoption der automatischen Partitionierung zu verwenden.
 - 3.3. Klicken Sie im Fenster **Installation Options** auf **Reclaim space**. Da die **/dev/vda**-Disk bereits über Partitionen und Dateisysteme aus der vorherigen Installation verfügt, können Sie mit dieser Auswahl die Disk für die neue Installation löschen. Klicken Sie im Fenster **Reclaim Disk Space** auf **Delete all** und dann auf **Reclaim space**.

- ▶ 4. Legen Sie den Serverhostnamen auf **servera.lab.example.com** fest und verifizieren Sie die Konfiguration der Netzwerkschnittstelle.
 - 4.1. Klicken Sie auf **Network & Host Name**.
 - 4.2. Geben Sie im Feld **Host Name** die Adresse **servera.lab.example.com** ein, und klicken Sie dann auf **Apply**.
 - 4.3. Klicken Sie auf **Configure** und dann auf die Registerkarte **IPv4 Settings**.
 - 4.4. Stellen Sie sicher, dass die Netzwerkparameter korrekt sind. Die IP-Adresse lautet 172.25.250.10, die Netzmaske lautet 24 und das Gateway und der Namensserver sind jeweils auf 172.25.250.254 festgelegt. Klicken Sie auf **Save**.
 - 4.5. Bestätigen Sie, dass die Netzwerkschnittstelle aktiviert ist, indem Sie **ON/OFF** auf **ON** festlegen.
 - 4.6. Klicken Sie auf **Done**.
- ▶ 5. Legen Sie das Feld **Installation Source** auf `http://content.example.com/rhel8.2/x86_64/dvd` fest.
 - 5.1. Klicken Sie auf **Installation Source**.
 - 5.2. Wählen Sie **On the network** aus.
 - 5.3. Geben Sie `content.example.com/rhel8.2/x86_64/dvd` im Feld **http://** ein.
 - 5.4. Klicken Sie auf **Done**.
- ▶ 6. Wählen Sie die Software aus, die für eine Minimalinstallation erforderlich ist.
 - 6.1. Klicken Sie auf **Software Selection**.
 - 6.2. Wählen Sie **Minimal Install** in der Liste **Base Environment** aus.
 - 6.3. Klicken Sie auf **Done**.
- ▶ 7. Konfigurieren Sie den Zweck des Systems.
 - 7.1. Klicken Sie auf **Connect to Red Hat**.
 - 7.2. Wählen Sie **Purpose** aus.
 - 7.3. Wählen Sie die Rolle **Red Hat Enterprise Linux Server** aus.
 - 7.4. Wählen Sie **Self-Support** als ein SLA-Level aus.
 - 7.5. Wählen Sie **Development/Test** für die Verwendung aus.
 - 7.6. Ändern Sie keinen anderen Parameter. Klicken Sie auf **Done**.
- ▶ 8. Klicken Sie auf **Begin Installation**.
- ▶ 9. Legen Sie während des Installationsvorgangs das Passwort für **root** auf **redhat** fest.
 - 9.1. Klicken Sie auf **Root Password**.

- 9.2. Geben Sie **redhat** im Feld **Root Password** ein.
 - 9.3. Geben Sie **redhat** im Feld **Confirm** ein.
 - 9.4. Das Passwort ist unsicher. Daher müssen Sie doppelt auf **Done** klicken.
- 10. Fügen Sie während des Installationsvorgangs den Benutzer **student** hinzu.
- 10.1. Klicken Sie auf **User Creation**.
 - 10.2. Geben Sie **student** im Feld **Full Name** ein.
 - 10.3. Aktivieren Sie **Make this user administrator**, sodass **sudo** von **student** verwendet werden kann, um Befehle als **root** auszuführen.
 - 10.4. Geben Sie **student** im Feld **Password** ein.
 - 10.5. Geben Sie **student** im Feld **Confirm Password** ein.
 - 10.6. Das Passwort ist unsicher. Daher müssen Sie doppelt auf **Done** klicken.
- 11. Klicken Sie nach Abschluss der Installation auf **Reboot**.
- 12. Wenn vom System die Anmeldeaufforderung angezeigt wird, melden Sie sich als **student** mit dem Passwort **student** an.

Beenden

Verwenden Sie die für Ihre Kursumgebung geeignete Methode, um Ihren Rechner **servera** zurückzusetzen.

Hiermit ist die angeleitete Übung beendet.

Automatisieren der Installation mit Kickstart

Ziele

In diesem Abschnitt werden die folgenden Themen behandelt:

- Erläutern der Kickstart-Konzepte und -Architektur.
- Erstellen einer Kickstart-Datei mit der Kickstart Generator-Website.
- Ändern einer vorhandenen Kickstart-Datei mit einem Texteditor und Überprüfen der zugehörigen Syntax mit **ksvalidator**.
- Veröffentlichen einer Kickstart-Datei für das Installationsprogramm.
- Durchführen einer Kickstart-Installation im Netzwerk.

Erstellen eines Kickstart-Profs

Mit dem *Kickstart*-Feature können Sie die Installation von Red Hat Enterprise Linux automatisieren. Mit Kickstart geben Sie alles an, was Anaconda zum Abschluss einer Installation benötigt, einschließlich Festplattenpartitionierung, Konfiguration der Netzwerkschnittstelle, Paketauswahl und anderer Parameter in einer Kickstart-Textdatei. Durch den Verweis auf die Textdatei führt Anaconda die Installation ohne weitere Benutzerinteraktion durch.



Anmerkung

Kickstart in Red Hat Enterprise Linux ähnelt der Jumpstart-Funktion in Oracle Solaris oder der Verwendung einer Antwortdatei für das unbeaufsichtigte Setup für Microsoft Windows.

Die Kickstart-Dateien beginnen mit einer Liste von Befehlen, die festlegen, auf welche Weise der Zielrechner zu installieren ist. Zeilen, die mit einem #-Zeichen beginnen, sind Kommentare, die vom Installationsprogramm ignoriert werden. Zusätzliche Abschnitte beginnen mit einer *Direktive*, deren erstes Zeichen ein % ist, und enden in einer Zeile mit einer **%end**-Direktive.

Der Abschnitt **%packages** gibt die Software an, die auf dem Zielsystem zu installieren ist. Geben Sie einzelne Pakete nach Name an (ohne Versionen). Paketgruppen beginnen mit einem @-Zeichen und werden durch einen Namen oder eine ID angegeben. Umgebungsgruppen (Gruppen von Paketgruppen) beginnen mit @^-Zeichen. Geben Sie Module, Streams und Profile mit der Syntax @*module:stream/profile* an.

Gruppen enthalten obligatorische, standardmäßige und optionale Komponenten. In der Regel werden von Kickstart obligatorische und standardmäßige Komponenten installiert. Um ein Paket oder eine Paketgruppe aus der Installation auszuschließen, stellen Sie ihm bzw. ihr ein --Zeichen voran. Ausgeschlossene Pakete oder Paketgruppen können jedoch weiterhin installiert werden, wenn sie obligatorische Abhängigkeiten von anderen angeforderten Paketen sind.

Eine Kickstart-Konfiguration verwendet normalerweise zwei zusätzliche Abschnitte: **%pre** und **%post**, die Shell-Skriptbefehle enthalten, die das System weiter konfigurieren. Das **%pre**-Skript wird ausgeführt, bevor die Disk-Partitionierung erfolgt. Normalerweise wird dieser Abschnitt nur

verwendet, wenn Aktionen erforderlich sind, um ein Gerät vor der Disk-Partitionierung zu erkennen oder zu initialisieren. Das **%post**-Skript wird ausgeführt, nachdem die Installation andernfalls abgeschlossen ist.

Sie müssen die primären Kickstart-Befehle vor den Abschnitten **%pre**, **%post** und **%packages** angeben. Andernfalls können Sie diese Abschnitte jedoch in beliebiger Reihenfolge in der Datei platzieren.

Kickstart-Dateibefehle

Installationsbefehle

Definieren Sie die Installationsquelle und wie Sie die Installation durchführen. Zu jeder folgt ein Beispiel.

- **url**: Gibt die URL an, die auf das Installationsmedium verweist.

```
url --url="http://classroom.example.com/content/rhel8.2/x86_64/dvd/"
```

- **repo**: Gibt an, wo zusätzliche Pakete für die Installation gefunden werden. Diese Option muss auf ein gültiges **yum**-Repository verweisen.

```
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.2/x86_64/dvd/AppStream/
```

- **text**: Erzwingt eine Installation von Textmodus.
- **vnc**: Lässt Remote-Anzeigen der grafischen Installation per VNC zu.

```
vnc --password=redhat
```

Partitionierungsbefehle

Definieren Sie das zu verwendende Gerät und das Partitionsschema.

- **clearpart**: Entfernt Partitionen aus dem System, bevor neue Partitionen erstellt werden. Standardmäßig werden keine Partitionen entfernt.

```
clearpart --all --drives=sda,sdb --initlabel
```

- **part**: Gibt Größe, Format und Name einer Partition an.

```
part /home --fstype=ext4 --label=homes --size=4096 --maxsize=8192 --grow
```

- **autopart**: Erstellt automatisch eine Root-Partition, eine Swap-Partition und eine geeignete Startpartition für die Architektur. Auf ausreichend großen Laufwerken entsteht dadurch auch eine **/home**-Partition.
- **ignoredisk**: Steuert den Zugriff von Anaconda auf an das System angehängte Disks.

```
ignoredisk --drives=sdc
```

- **bootloader**: Legt fest, wo der Bootloader zu installieren ist.

```
bootloader --location=mbr --boot-drive=sda
```

- **volgroup, logvol**: Erstellt LVM-Volume-Gruppen und logische Volumes.

```
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**: Initialisiert Disks, deren Formatierung nicht erkannt wird.

Netzwerbefehle

Definieren Sie die vom Host verwendeten Netzwerkfeatures.

- **network**: Konfiguriert Netzwerkinformationen für das Zielsystem. Aktiviert Netzwerkgeräte in der Installationsumgebung.

```
network --device=eth0 --bootproto=dhcp
```

- **firewall**: Definiert die Firewall-Konfiguration für das Zielsystem.

```
firewall --enabled --service=ssh,http
```

Standort- und Sicherheitsbefehle

Konfigurieren Sie Einstellungen für Sicherheit, Sprache und Regionen.

- **lang**: Stellt die während der Installation zu verwendende Sprache sowie die Standardsprache des installierten Systems ein. Erforderlich.

```
lang en_US.UTF-8
```

- **keyboard**: Legt den Tastaturtyp des Systems fest. Erforderlich.

```
keyboard --vckeymap=us --xlayouts=''
```

- **timezone**: Definiert die Zeitzone, NTP-Server und ob die Hardware-Uhr UTC verwendet.

```
timezone --utc --ntpservers=time.example.com Europe/Amsterdam
```

- **authselect**: Legt Authentifizierungsoptionen fest. Von **authselect** erkannte Optionen sind für diesen Befehl gültig. Siehe authselect(8).

- **rootpw**: Definiert das initiale **root**-Kennwort.

```
rootpw --plaintext redhat
or
rootpw --iscrypted $6$KUnFfrTz08jv.PiH$Y1Bb0tXBkWzoMuRfb0.SpbQ...XDR1UuchoMG1
```

- **selinux**: Legt den SELinux-Modus für das installierte System fest.

```
selinux --enforcing
```

- **services**: Ändert den Standardsatz von Services, die unter dem standardmäßigen Ziel **systemd** ausgeführt werden.

```
services --disabled=network,iptables,ip6tables --enabled=NetworkManager,firewalld
```

- **group, user**: Erstellt eine lokale Gruppe oder einen lokalen Benutzer auf dem System.

```
group --name=admins --gid=10001
user --name=jdoe --gecos="John Doe" --groups=admins --password=changeme --
plaintext
```

Verschiedene Befehle

Konfigurieren Sie verschiedene Elemente, die sich auf die Protokollierung während der Installation und den Zustand der Hostversorgung nach Abschluss der Installation beziehen.

- **logging**: Dieser Befehl definiert, auf welche Weise Anaconda während der Installation protokolliert.

```
logging --host=loghost.example.com --level=info
```

- **firstboot**: Wenn aktiviert, wird der Setup-Agent beim ersten Systemstart gestartet. Das *initial-setup*-Paket muss installiert sein.

```
firstboot --disabled
```

- **reboot, poweroff, halt**: Geben Sie die letzte Aktion an, die nach Abschluss der Installation ausgeführt werden soll.



Anmerkung

Das Dienstprogramm **ksverdiff** aus dem *pykickstart*-Paket ist nützlich, um Änderungen in der Kickstart-Dateisyntax zwischen zwei Versionen von Red Hat Enterprise Linux oder Fedora zu identifizieren.

Beispielsweise identifiziert **ksverdiff -f RHEL7 -t RHEL8** Änderungen in der Syntax zwischen RHEL 7 und RHEL 8. Die verfügbaren Versionen werden am Anfang der Datei **/usr/lib/python3.6/site-packages/pykickstart/version.py** aufgelistet.

Beispiel für eine Kickstart-Datei

Der erste Teil der Datei besteht aus den Installationsbefehlen, wie dem Partitionieren der Festplatte und der Installationsquelle.

```
#version=RHEL8
ignoredisk --only-use=vda
# System bootloader configuration
```

Kapitel 12 | Installation von Red Hat Enterprise Linux

```
bootloader --append="console=ttyS0 console=ttyS0,115200n8 no_timer_check
    net.ifnames=0 crashkernel=auto" --location=mbr --timeout=1 --boot-drive=vda
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Use text mode install
text
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.2/
x86_64/dvd/AppStream/
# Use network installation
url --url="http://classroom.example.com/content/rhel8.2/x86_64/dvd/"
# Keyboard layouts
# old format: keyboard us
# new format:
keyboard --vckeymap=us --xlayouts=''
# System language
lang en_US.UTF-8
# Root password
rootpw --plaintext redhat
# System authorization information
auth --enablesystem --passalgo=sha512
# SELinux configuration
selinux --enforcing
firstboot --disable
# Do not configure the X Window System
skipx
# System services
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chrony"
# System timezone
timezone America/New_York --isUtc
# Disk partitioning information
part / --fstype="xfs" --ondisk=vda --size=10000
```

Der zweite Teil enthält den Abschnitt **%packages**, der genau angibt, welche Pakete und Paketgruppen installiert werden sollten und welche Pakete nicht installiert werden sollten.

```
%packages
@core
chrony
cloud-init
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
-plymouth

%end
```

Der letzte Teil enthält alle **%pre**- und **%post**-Installationsskripts.

```
%post --erroronfail

# For cloud images, 'eth0' _is_ the predictable device name, since
# we don't want to be tied to specific virtual (!) hardware
rm -f /etc/udev/rules.d/70*
ln -s /dev/null /etc/udev/rules.d/80-net-name-slot.rules

# simple eth0 config, again not hard-coded to the build hardware
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
EOF

%end
```



Anmerkung

In einer Kickstart-Datei führen fehlende erforderliche Werte dazu, dass das Installationsprogramm interaktiv eine Antwort anfordert oder die Installation abgebrochen wird.

Schritte für die Kickstart-Installation

Führen Sie die folgenden Schritte durch, um die Installation von Red Hat Enterprise Linux erfolgreich zu automatisieren:

1. Erstellen Sie eine Kickstart-Datei.
2. Veröffentlichen Sie die Kickstart-Datei für das Installationsprogramm.
3. Starten Sie Anaconda, und verweisen Sie auf die Kickstart-Datei.

Erstellen einer Kickstart-Datei

Verwenden Sie eine der folgenden Methoden, um eine Kickstart-Datei zu erstellen:

- Verwenden Sie die Kickstart Generator-Website.
- Verwenden Sie einen Texteditor.

Die Kickstart Generator-Website unter <https://access.redhat.com/labs/kickstartconfig/> zeigt Dialogfelder für Benutzereingaben an und erstellt eine Kickstart-Direktiven-Textdatei mit den Auswahlmöglichkeiten des Benutzers. Jedes Dialogfeld entspricht den konfigurierbaren Elementen im Anaconda-Installationsprogramm.

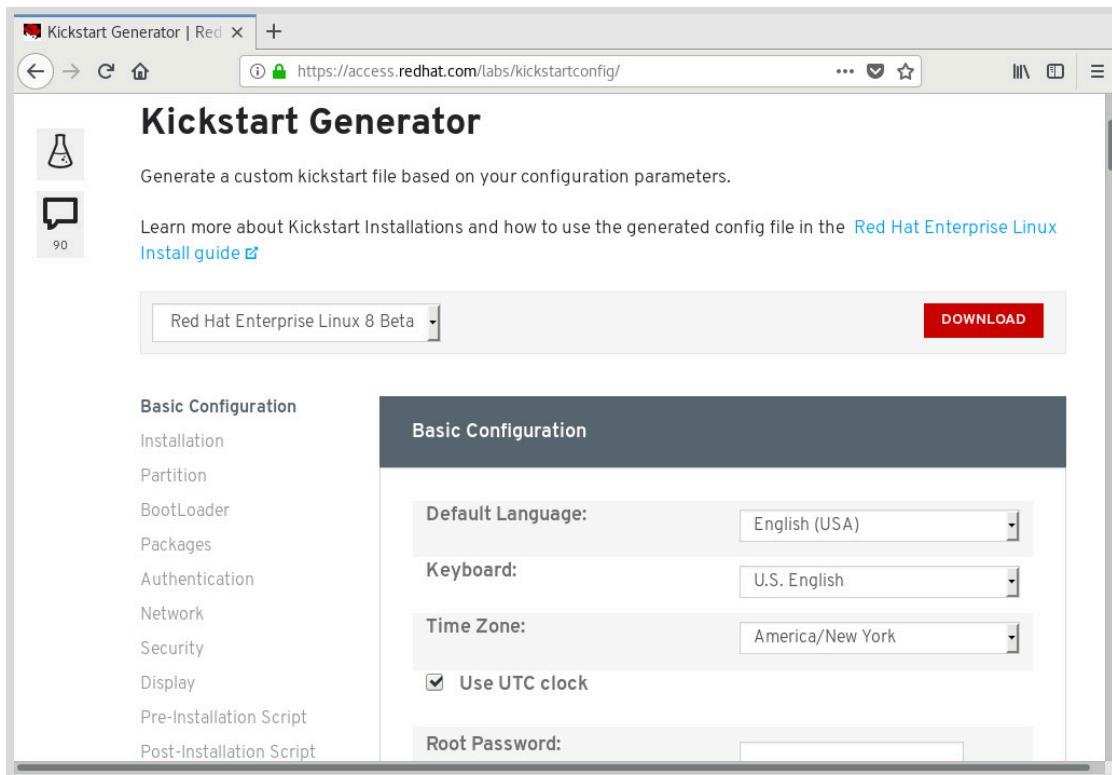


Abbildung 12.3: Grundlegende Konfiguration mit Kickstart Generator

**Anmerkung**

Zum Zeitpunkt der Erstellung dieses Dokuments war auf der Kickstart Generator-Website Red Hat Enterprise Linux 8 nicht als Menüoption vorhanden. Eine gültige Auswahl war Red Hat Enterprise Linux 8 Beta.

Das Erstellen einer Kickstart-Datei von Grund auf ist in der Regel zu komplex. Die Bearbeitung einer vorhandenen Kickstart-Datei ist jedoch üblich und nützlich. Jede Installation erstellt eine **/root/anaconda-ks.cfg**-Datei, welche die bei der Installation verwendeten Kickstart-Anweisungen enthält. Diese Datei stellt einen guten Ausgangspunkt für das manuelle Erstellen einer Kickstart-Datei dar.

Das Dienstprogramm **ksvalidator** prüft eine Kickstart-Datei auf Syntaxfehler. Es stellt sicher, dass Schlüsselwörter und Optionen korrekt verwendet werden. URL-Pfade, einzelne Pakete und Gruppen sowie die Bestandteile von **%post** oder **%pre** werden hingegen nicht überprüft. Wenn z. B. die Direktive **firewall --disabled** falsch buchstabiert ist, kann **ksvalidator** einen der folgenden Fehler ausgeben:

```
[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
```

Das `pykickstart`-Paket stellt **ksvalidator** bereit.

Veröffentlichen der Kickstart-Datei in Anaconda

Stellen Sie die Kickstart-Datei für das Installationsprogramm zur Verfügung, indem Sie sie an einem der folgenden Speicherorte ablegen:

- Ein Netzwerkserver, der während der Installation über FTP, HTTP oder NFS verfügbar ist.
- Eine verfügbare USB-Disk oder CD-ROM.
- Eine lokale Festplatte des zu installierenden Systems.

Um die automatisierte Installation zu starten, muss das Installationsprogramm auf die Kickstart-Datei zugreifen. Die gebräuchlichste Automatisierungsmethode verwendet einen Netzwerkserver wie einen FTP-, Web- oder NFS-Server. Netzwerkserver nutzen die Kickstart-Dateiwartung, da Änderungen einmal durchgeführt werden können und anschließend sofort für mehrere künftige Installationen verwendet werden können.

Die Bereitstellung von Kickstart-Dateien auf USB oder CD-ROM ist ebenfalls praktisch. Hierzu wird die Kickstart-Datei in die Startmedien eingebettet, die zum Starten der Installation verwendet werden. Wenn die Kickstart-Datei geändert wird, müssen Sie jedoch neue Installationsmedien erstellen.

Durch die Bereitstellung der Kickstart-Datei auf einer lokalen Disk können Sie ein System schnell neu erstellen.

Starten von Anaconda und verweisen auf die Kickstart-Datei

Wenn die Kickstart-Methode ausgewählt ist, wird dem Installationsprogramm mitgeteilt, wo sich die Kickstart-Datei befindet, indem der Parameter **inst.ks=LOCATION** an den Installationskernell weitergegeben wird. Einige Beispiele:

- `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`

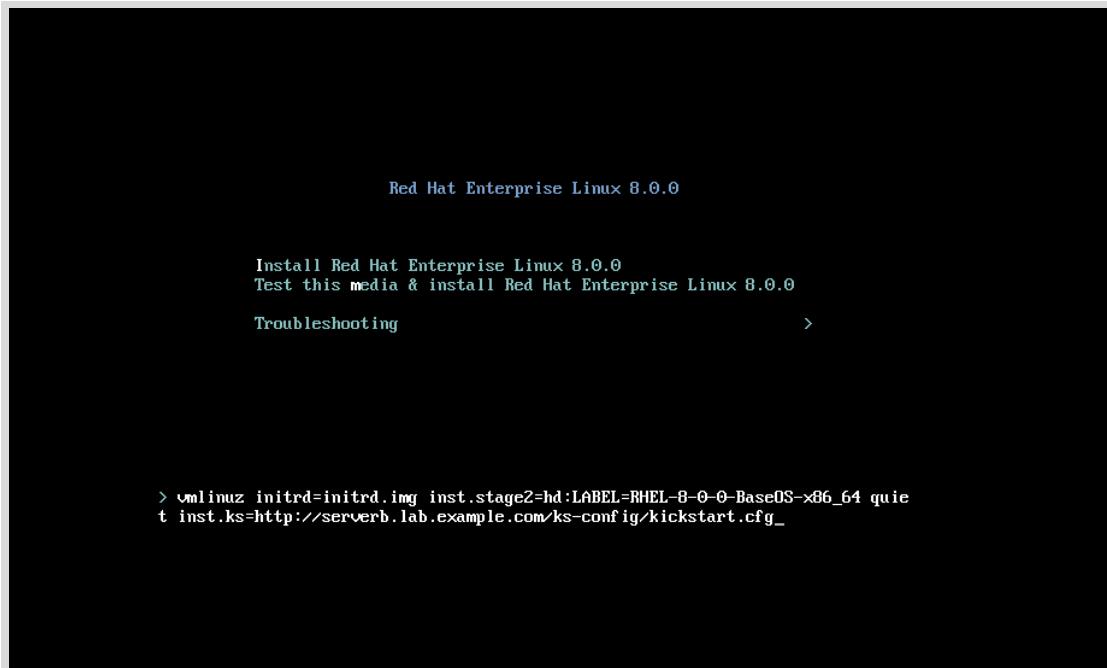


Abbildung 12.4: Angeben des Kickstart-Speicherorts während der Installation

Beim Installieren von virtuellen Rechnern mittels **Virtual Machine Manager** oder **virt-manager** kann die Kickstart-URL in einem Feld unter den **URL Options** angegeben werden.

Beim Installieren von physischen Systemen starten Sie unter Verwendung der Installationsmedien und drücken Sie die **Tab**-Taste, um den Startvorgang zu unterbrechen. Fügen Sie dem Installationskernel einen **inst.ks=LOCATION**-Parameter hinzu.



Literaturhinweise

Kapitel *Kickstart installation basics in Performing an advanced RHEL installation* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-installation-basics_installing-rhel-as-an-experienced-user

Abschnitt *Kickstart commands for installation program configuration and flow control* unter *Appendix B. Kickstart commands and options reference in Performing an advanced RHEL installation* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installation-basics_installing-rhel-as-an-experienced-user#kickstart-commands-for-installation-program-configuration-and-flow-control_kickstart-commands-and-options-reference

Kapitel *Boot options in Performing an advanced RHEL installation* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/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

► Angeleitete Übung

Automatisieren der Installation mit Kickstart

In dieser Übung erstellen Sie eine Kickstart-Datei und überprüfen die Syntax.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen einer Kickstart-Datei.
- Verwenden von **ksvalidator**, um die Syntax der Kickstart-Datei zu überprüfen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab installing-kickstart start** aus.

Dieser Befehl führt ein Startskript aus, um zu bestimmen, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem wird überprüft, ob Apache auf **servera** installiert und konfiguriert ist.

```
[student@workstation ~]$ lab installing-kickstart start
```

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Kopieren Sie **/root/anaconda-ks.cfg** auf **servera** in eine Datei namens **/home/student/kickstart.cfg**, damit sie von **student** bearbeitet werden kann. Kopieren Sie die Datei **/root/anaconda-ks.cfg** mit dem Befehl **sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg** nach **/home/student/kickstart.cfg**. Wenn **sudo** Sie zur Eingabe des Passworts für den Benutzer **student** auffordert, geben Sie **student** als Passwort ein.

```
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

- 3. Nehmen Sie die folgenden Änderungen an **/home/student/kickstart.cfg** vor.
- 3.1. Kommentieren Sie die „reboot“-Anweisung aus:

```
#reboot
```

- 3.2. Kommentieren Sie den Befehl **repo** für das BaseOS-Repository aus. Ändern Sie den Befehl **repo** für AppStream so, dass er auf das AppStream-Repository des Kursraums verweist.

```
#repo --name="koji-override-0" --baseurl=http://download-node-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.2.0-updates-20200423.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.2/x86_64/dvd/AppStream/
```

- 3.3. Ändern Sie den Befehl **url**, um die im Kursraum verwendeten Quellmedien für die Installation per HTTP anzugeben:

```
url --url="http://classroom.example.com/content/rhel8.2/x86_64/dvd/"
```

- 3.4. Kommentieren Sie den Befehl **network** aus:

```
#network --bootproto=dhcp --device=link --activate
```

- 3.5. Legen Sie das Root-Passwort auf **redhat** fest. Ändern Sie die Zeile, die mit **rootpw** beginnt, wie folgt:

```
rootpw --plaintext redhat
```

- 3.6. Löschen Sie die Zeile, die den Befehl **auth** enthält, und fügen Sie die Zeile **authselect select sssd** zum Festlegen des Service **sssd** als Identitäts- und Authentifizierungsquelle hinzu.

```
authselect select sssd
```

In Red Hat Enterprise Linux 8 ersetzt der Befehl **authselect** den Befehl **authconfig**.

- 3.7. Vereinfachen Sie den Befehl **services** so, dass er genau wie folgt aussieht:

```
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chronyd"
```

- 3.8. Kommentieren Sie die Befehle **part** aus. Fügen Sie den Befehl **autopart** hinzu:

```
# Disk partitioning information
#part biosboot --fstype="biosboot" --size=1
#part /boot/efi --fstype="efi" --size=100 --fsoptions="..."
#part / --fstype="xfs
autopart
```

- 3.9. Löschen Sie den gesamten Inhalt zwischen **%post** und **%end**. Fügen Sie folgende Zeile hinzu: **echo "Kickstarted on \$(date)" >> /etc/issue**

Der Abschnitt **%post** sollte wie folgt aussehen.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

3.10. Vereinfachen Sie die Paketspezifikation, sodass sie wie folgt aussieht:

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
httpd
-plymouth
%end
```

Wenn Sie mit der Bearbeitung der Datei fertig sind, speichern Sie die Datei und beenden den Editor.

- 4. Führen Sie den Befehl **ksvalidator** aus, um die Kickstart-Datei auf Syntaxfehler zu prüfen.

```
[student@servera ~]$ ksvalidator kickstart.cfg
```

- 5. Kopieren Sie **kickstart.cfg** in das Verzeichnis **/var/www/html/ks-config**.

```
[student@servera ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

- 6. Beenden Sie **servera**.

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

Beenden

Führen Sie auf **workstation** das Skript **lab installing-kickstart finish** aus, um diese Übung abzuschließen.

```
[student@workstation ~]$ lab installing-kickstart finish
```

Hiermit ist die angeleitete Übung beendet.

Installieren und Konfigurieren virtueller Rechner

Ziele

In diesem Abschnitt wird beschrieben, wie ein virtueller Rechner auf Ihrem Red Hat Enterprise Linux-Server mit Cockpit installiert wird.

Einführung in die KVM-Virtualisierung

Die Virtualisierung ermöglicht das Aufteilen eines einzigen physischen Systems in mehrere *virtuelle Rechner (VM)*, auf denen jeweils ein eigenes Betriebssystem ausgeführt werden kann.

Red Hat Enterprise Linux 8 unterstützt *KVM* (*einen Kernel-basierten virtuellen Rechner*), eine vollständige Virtualisierungslösung, die im Linux-Standardkernel integriert ist. KVM kann mehrere Windows- und Linux-Gastbetriebssysteme ausführen.

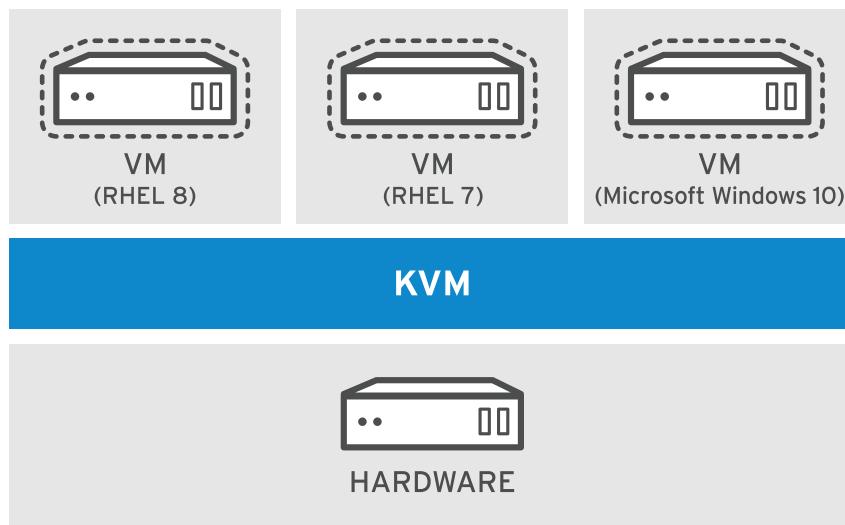


Abbildung 12.5: KVM-Virtualisierung

Verwalten Sie KVM in Red Hat Enterprise Linux mit dem Befehl **virsh** oder mit dem Tool für virtuelle Rechner von Cockpit.

KVM bietet die VM-Technologie für alle Red Hat-Produkte, von physischen Standalone-Instanzen von Red Hat Enterprise Linux bis hin zu Red Hat OpenStack Platform:

- Auf physischen Hardwaresystemen wird Red Hat Enterprise Linux ausgeführt, um die KVM-Virtualisierung bereitzustellen. Red Hat Enterprise Linux ist in der Regel ein *Thick Host*, also ein System, das VMs unterstützt und gleichzeitig andere lokale und Netzwerkservices, Anwendungen und Verwaltungsfunktionen bereitstellt.
- *Red Hat Virtualization (RHV)* bietet eine zentralisierte Weboberfläche, über die Administratoren die gesamte virtuelle Infrastruktur verwalten können. Es umfasst erweiterte Features wie KVM-Migration, Redundanz und Hochverfügbarkeit. Ein *Red Hat Virtualization Hypervisor* ist eine optimierte Version von Red Hat Enterprise Linux für den alleinigen Zweck der Bereitstellung und Unterstützung von VMs.

- Red Hat OpenStack Platform (RHOSP) bietet die Grundlage zum Erstellen, Bereitstellen und Skalieren einer Public oder Private Cloud.

Red Hat unterstützt virtuelle Rechner, die auf diesen Betriebssystemen ausgeführt werden:

- Red Hat Enterprise Linux 6 und höher
- Microsoft Windows 10 und höher
- Microsoft Windows Server 2016 und höher

Konfigurieren eines physischen Red Hat Enterprise Linux-Systems als einen Virtualisierungshost

Red Hat Enterprise Linux kann von Administratoren als Virtualisierungshost konfiguriert werden und ist dann geeignet für Entwicklungs-, Test- und Schulungszwecke oder das Arbeiten in mehreren Betriebssystemen gleichzeitig.

Installieren der Virtualisierungstools

Installieren Sie das Yum-Modul `virt`, um ein System darauf vorzubereiten, ein Virtualisierungshost zu werden.

```
[root@host ~]# yum module list virt
Name           Stream      Profiles      Summary
virt           rhel [d][e]  common [d]   Virtualization module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@host ~]# yum module install virt
...output omitted...
```

Verifizieren der Systemanforderungen

KVM benötigt entweder einen Intel-Prozessor mit Intel VT-x- und Intel 64-Erweiterungen für x86-basierte Systeme oder einen AMD-Prozessor mit den Erweiterungen AMD-V und AMD64. Führen Sie den Befehl `virt-host-validate` aus, um Ihre Hardware zu verifizieren und die Systemanforderungen zu überprüfen.

```
[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 'memory' controller support: PASS
QEMU: Checking for cgroup 'memory' controller mount-point: PASS
QEMU: Checking for cgroup 'cpu' controller support    : PASS
QEMU: Checking for cgroup 'cpu' controller mount-point: PASS
QEMU: Checking for cgroup 'cpuacct' controller support: PASS
QEMU: Checking for cgroup 'cpuacct' controller mount-point: PASS
QEMU: Checking for cgroup 'cpuset' controller support: PASS
QEMU: Checking for cgroup 'cpuset' controller mount-point: PASS
QEMU: Checking for cgroup 'devices' controller support: PASS
QEMU: Checking for cgroup 'devices' controller mount-point: PASS
```

```
QEMU: Checking for cgroup 'blkio' controller support      : PASS
QEMU: Checking for cgroup 'blkio' controller mount-point : PASS
QEMU: Checking for device assignment IOMMU support       : PASS
```

Das System muss alle Validierungselemente bestehen, um ein KVM-Host sein zu können.

Verwalten virtueller Rechner mit Cockpit

Das Yum-Modul *virt* stellt den Befehl **virsh** bereit, um Ihre virtuellen Rechner zu verwalten. Das Cockpit-Tool bietet eine Webkonsole für die KVM-Verwaltung und die Erstellung virtueller Rechner.

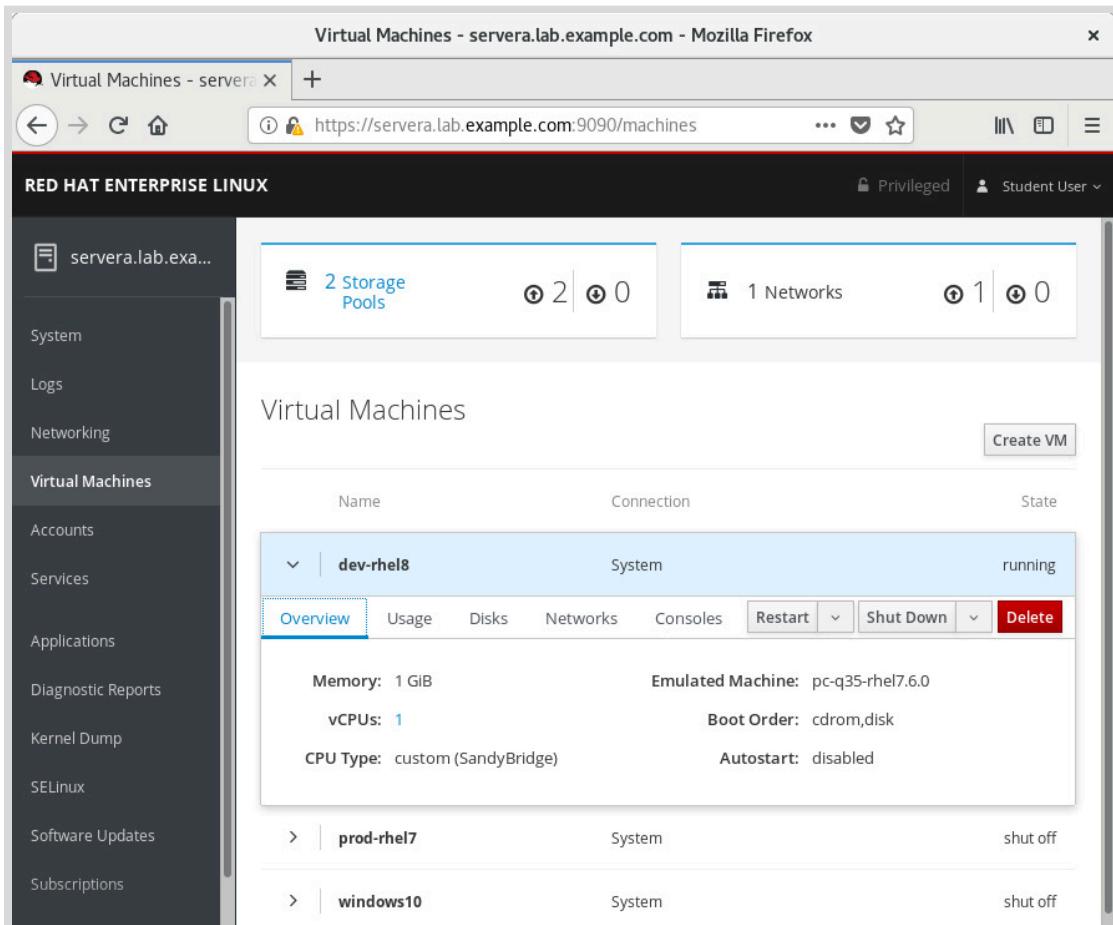


Abbildung 12.6: Verwaltung virtueller Rechner in Cockpit

Installieren Sie das Paket *cockpit-machines*, um Cockpit das Menü **Virtual Machines** hinzuzufügen.

```
[root@host ~]# yum install cockpit-machines
```

Wenn Cockpit noch nicht ausgeführt wird, starten Sie und aktivieren Sie es.

```
[root@host ~]# systemctl enable --now cockpit.socket
```

Greifen Sie auf der Cockpit-Weboberfläche auf das Menü **Virtual Machines** zu, um einen neuen virtuellen Rechner mit Cockpit zu erstellen. Klicken Sie dort auf **Create VM**, und geben Sie im Fenster **Create New Virtual Machine** die VM-Konfiguration ein.

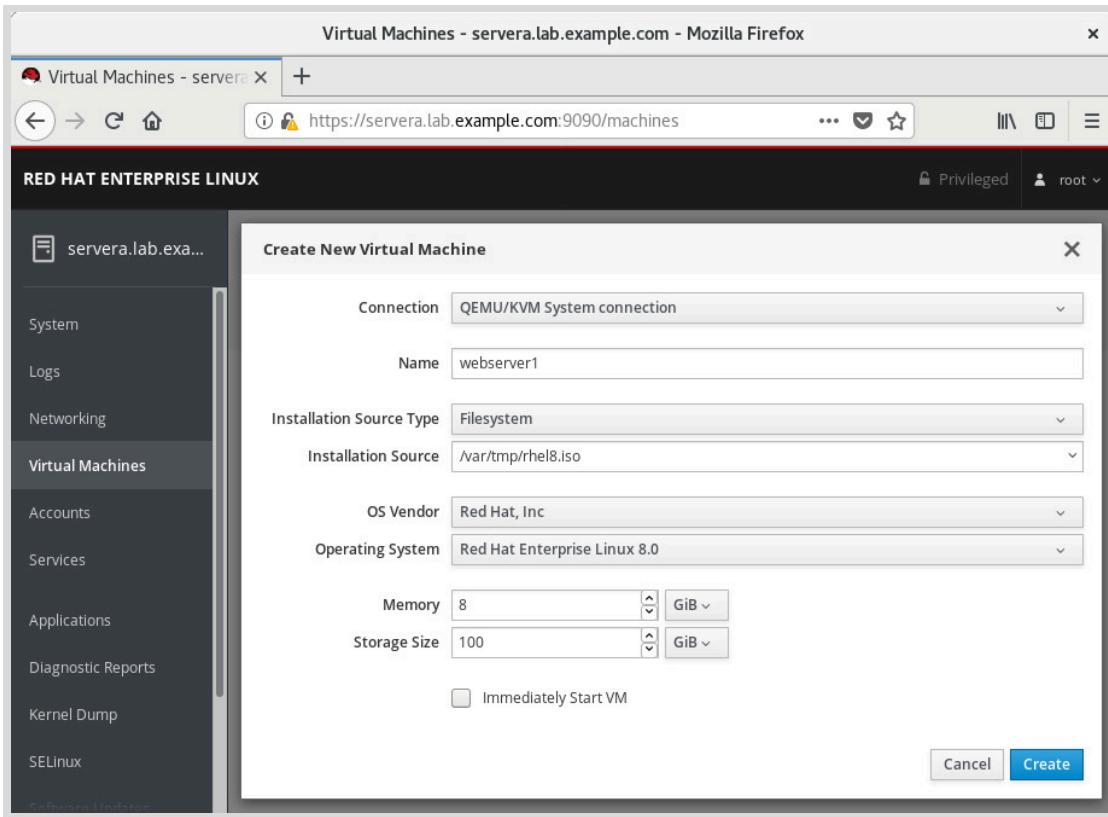


Abbildung 12.7: Erstellen eines virtuellen Rechners in Cockpit

- **Name** legt einen *Domain*-Namen für die Konfiguration des virtuellen Rechners fest. Dieser Name steht in keinem Zusammenhang mit dem Netzwerkhostnamen, den Sie dem System in der installierten VM geben.
- **Installation Source Type** ist die Methode zum Abrufen der Installations-ISO-Datei. Zur Auswahl stehen das lokale Dateisystem oder eine HTTPS-, FTP- oder NFS-URL.
- **Installation Source** gibt den Pfad zur Installationsquelle an.
- **OS Vendor** und **Operating System** geben das Betriebssystem des virtuellen Rechners an. Die Virtualisierungsebene stellt eine Hardware-Emulation dar, die mit dem ausgewählten Betriebssystem kompatibel ist.
- **Memory** ist die Menge an RAM, die der neuen VM zur Verfügung gestellt werden soll.
- **Storage Size** ist die Disk-Größe für die neue VM. Ordnen Sie der VM nach der Installation weitere Disks zu.
- **Immediately Start VM** gibt an, ob die VM unmittelbar nach dem Klicken auf **Create** gestartet werden soll.

Klicken Sie auf **Create**, um die VM zu erstellen, und **Install**, um die Installation des Betriebssystems zu starten. Cockpit zeigt die VM-Konsole an, von der aus Sie das System installieren können.

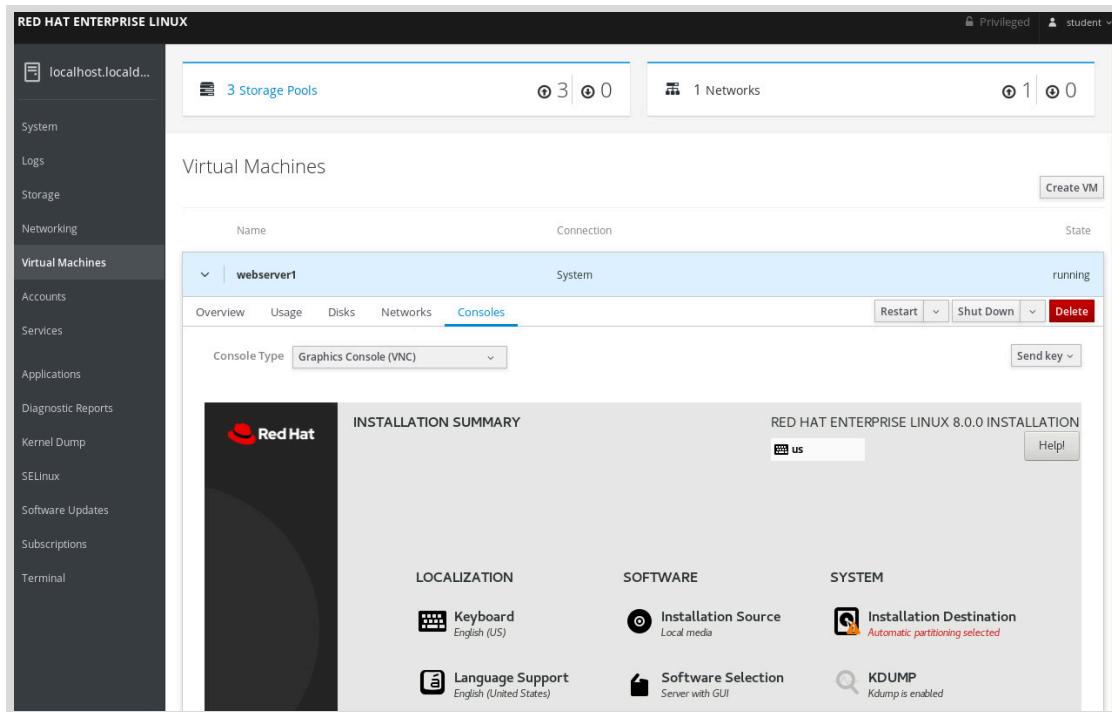


Abbildung 12.8: Installieren des Betriebssystems des virtuellen Rechners



Literaturhinweise

Weitere Informationen finden Sie im *Configuring and managing virtualization Guide* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_virtualization/index

Was ist Virtualisierung?

<https://www.redhat.com/en/topics/virtualization/what-is-virtualization>

► Quiz

Installieren und Konfigurieren virtueller Rechner

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welche drei Gastbetriebssysteme unterstützt Red Hat als virtuelle KVM-Rechner?
(Wählen Sie drei Antworten aus.)
- a. Fedora 28 und höher
 - b. Red Hat Enterprise Linux 6 und höher
 - c. CoreOS Container Linux 2023 und höher
 - d. Microsoft Windows 7 SP1
 - e. Microsoft Windows 10 und höher
 - f. Microsoft Windows Server 2016 und höher
- 2. Welche zwei Komponenten sind erforderlich, um Ihr System als Virtualisierungshost zu konfigurieren und virtuelle Rechner mit Web Console zu verwalten? (Wählen Sie zwei Antworten aus.)
- a. Das Yum-Modul *virt*
 - b. Die Paketgruppe *openstack*
 - c. Das Paket *cockpit-machines*
 - d. Die Paketgruppe *Virtualization Platform*
 - e. Das Yum-Modul *kvm*
 - f. Das Paket *cockpit-virtualization*
- 3. Welcher Befehl verifiziert, dass Ihr System Virtualisierung unterstützt?
- a. grep kvm /proc/cpuinfo
 - b. virsh validate
 - c. virt-host-validate
 - d. rhv-validate
 - e. cockpit-validate
- 4. Welche zwei Tools können Sie zum Starten und Stoppen Ihrer virtuellen Rechner auf einem Red Hat Enterprise Linux-System verwenden? (Wählen Sie zwei Antworten aus.)
- a. vmctl
 - b. libvirtd
 - c. virsh
 - d. OpenStack
 - e. Webkonsole

► Lösung

Installieren und Konfigurieren virtueller Rechner

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

► 1. Welche drei Gastbetriebssysteme unterstützt Red Hat als virtuelle KVM-Rechner?

(Wählen Sie drei Antworten aus.)

- a. Fedora 28 und höher
- b. Red Hat Enterprise Linux 6 und höher
- c. CoreOS Container Linux 2023 und höher
- d. Microsoft Windows 7 SP1
- e. Microsoft Windows 10 und höher
- f. Microsoft Windows Server 2016 und höher

► 2. Welche zwei Komponenten sind erforderlich, um Ihr System als Virtualisierungshost zu konfigurieren und virtuelle Rechner mit Web Console zu verwalten? (Wählen Sie zwei Antworten aus.)

- a. Das Yum-Modul *virt*
- b. Die Paketgruppe *openstack*
- c. Das Paket *cockpit-machines*
- d. Die Paketgruppe *Virtualization Platform*
- e. Das Yum-Modul *kvm*
- f. Das Paket *cockpit-virtualization*

► 3. Welcher Befehl verifiziert, dass Ihr System Virtualisierung unterstützt?

- a. grep kvm /proc/cpuinfo
- b. virsh validate
- c. virt-host-validate
- d. rhv-validate
- e. cockpit-validate

► 4. Welche zwei Tools können Sie zum Starten und Stoppen Ihrer virtuellen Rechner auf einem Red Hat Enterprise Linux-System verwenden? (Wählen Sie zwei Antworten aus.)

- a. vmctl
- b. libvirtd
- c. virsh
- d. OpenStack
- e. Webkonsole

► Praktische Übung

Installation von Red Hat Enterprise Linux

Leistungscheckliste

In dieser Übung erstellen Sie eine Kickstart-Datei und führen eine Kickstart-Installation auf **serverb** durch.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen einer Kickstart-Datei.
- Bereitstellen der Kickstart-Datei für das Installationsprogramm.
- Ausführen einer Kickstart-Installation.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab installing-review start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob die Rechner **servera** und **serverb** im Netzwerk erreichbar sind, und konfiguriert Apache auf **serverb**. Er konfiguriert auch das Bootmenü auf **serverb** für die Übung, um eine Kickstart-Installation durchzuführen.

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

Bereiten Sie entsprechend der Angabe eine Kickstart-Datei auf **serverb** vor, und stellen Sie sie unter <http://serverb.lab.example.com/ks-config/kickstart.cfg> zur Verfügung. Führen Sie mithilfe der von Ihnen vorbereiteten Kickstart-Datei eine Kickstart-Installation auf **servera** durch.

1. Kopieren Sie **/root/anaconda-ks.cfg** nach **/home/student/kickstart.cfg** auf **serverb**, damit der Benutzer **student** Bearbeitungen daran vornehmen kann.
2. Nehmen Sie die folgenden Änderungen an **/home/student/kickstart.cfg** vor.
 - Kommentieren Sie den Befehl **reboot** aus.
 - Kommentieren Sie den Befehl **repo** für das BaseOS-Repository aus. Ändern Sie den Befehl **repo**, damit das AppStream-Repository auf http://classroom.example.com/content/rhel8.2/x86_64/dvd/AppStream/ verweist. Der Repository-Name sollte auf Appstream gesetzt werden.
 - Ändern Sie den Befehl **url** so, dass http://classroom.example.com/content/rhel8.2/x86_64/dvd/ als Installationsquelle verwendet wird.
 - Kommentieren Sie den Befehl **network** aus.
 - Ändern Sie den Befehl **rootpw** für die Verwendung von **Klartext** und legen Sie das Root-Passwort auf **redhat** fest.

- Löschen Sie die Zeile, die den Befehl **auth** enthält, und fügen Sie die Zeile **authselect select sssd** zum Festlegen des Service **sssd** als Identitäts- und Authentifizierungsquelle hinzu.
 - Vereinfachen Sie den Befehl **services**, sodass nur die Services **kdump** und **rhsmdcertd** deaktiviert sind. Lassen Sie nur **sshd**, **rngd** und **chrony** aktiviert.
 - Fügen Sie den Befehl **autopart** hinzu. Die Befehle **part** sollten bereits auskommentiert sein.
 - Vereinfachen Sie den Abschnitt „%post“ so, dass nur ein Skript ausgeführt wird, um den Text **Kickstarted on DATE** an das Ende der Datei **/etc/issue** anzufügen. **DATE** ist eine variable Information und sollte vom Skript mit dem Befehl **date** ohne weitere Optionen generiert werden.
 - Vereinfachen Sie den Abschnitt **%package** wie folgt: Beziehen Sie die Pakete **@core**, **chrony**, **dracut-config-generic**, **dracut-norescue**, **firewalld**, **grub2**, **kernel**, **rsync**, **tar** und **httpd** ein. Stellen Sie sicher, dass das Paket **plymouth** nicht installiert wird.
3. Überprüfen Sie die Syntax von **kickstart.cfg**.
 4. Stellen Sie die Datei **/home/student/kickstart.cfg** unter **http://serverb.lab.example.com/ks-config/kickstart.cfg** zur Verfügung.
 5. Kehren Sie zum System **workstation** zurück, um Ihre Arbeit zu überprüfen.

Bewertung

Führen Sie auf **workstation** das Skript **lab installing-review grade** aus, um diese Übung zu bewerten. Booten Sie **servera** neu, um eine Kickstart-Installation durchzuführen.

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

Korrigieren Sie alle Fehler in **kickstart.cfg**, die vom **serverb**-Webserver freigegeben wird, indem Sie entweder **/var/www/html/ks-config/kickstart.cfg** direkt ändern oder **~/kickstart.cfg** ändern und sie nach **/var/www/html/ks-config/** kopieren.

Booten Sie **servera** neu, um eine Kickstart-Installation durchzuführen. Wählen Sie im GRUB-Menü **Kickstart Red Hat Enterprise Linux 8** aus, und drücken Sie die **Eingabetaste**.

Beenden

Führen Sie auf **workstation** das Skript **lab installing-review finish** aus, um diese Übung abzuschließen. Dieses Skript entfernt den während der Übung auf **serverb** konfigurierten Webserver.

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

Setzen Sie das System **servera** auf den Standardzustand zurück.

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Installation von Red Hat Enterprise Linux

Leistungscheckliste

In dieser Übung erstellen Sie eine Kickstart-Datei und führen eine Kickstart-Installation auf **serverb** durch.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen einer Kickstart-Datei.
- Bereitstellen der Kickstart-Datei für das Installationsprogramm.
- Ausführen einer Kickstart-Installation.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie auf **workstation** den Befehl **lab installing-review start** aus. Dieser Befehl führt ein Startskript aus, um zu ermitteln, ob die Rechner **servera** und **serverb** im Netzwerk erreichbar sind, und konfiguriert Apache auf **serverb**. Er konfiguriert auch das Bootmenü auf **serverb** für die Übung, um eine Kickstart-Installation durchzuführen.

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

Bereiten Sie entsprechend der Angabe eine Kickstart-Datei auf **serverb** vor, und stellen Sie sie unter <http://serverb.lab.example.com/ks-config/kickstart.cfg> zur Verfügung. Führen Sie mithilfe der von Ihnen vorbereiteten Kickstart-Datei eine Kickstart-Installation auf **servera** durch.

1. Kopieren Sie **/root/anaconda-ks.cfg** nach **/home/student/kickstart.cfg** auf **serverb**, damit der Benutzer **student** Bearbeitungen daran vornehmen kann.
 - 1.1. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als **student** an.

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

- 1.2. Kopieren Sie **/root/anaconda-ks.cfg** auf **serverb** in eine Datei namens **/home/student/kickstart.cfg**, damit sie von **student** bearbeitet werden kann. Kopieren Sie die Datei **/root/anaconda-ks.cfg** mit dem Befehl **sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg** nach **/home/student/kickstart.cfg**. Wenn **sudo** Sie zur Eingabe des Passworts für den Benutzer **student** auffordert, geben Sie **student** als Passwort ein.

```
[student@serverb ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg
[sudo] password for student: student
```

2. Nehmen Sie die folgenden Änderungen an **/home/student/kickstart.cfg** vor.
- Kommentieren Sie den Befehl **reboot** aus.
 - Kommentieren Sie den Befehl **repo** für das BaseOS-Repository aus. Ändern Sie den Befehl **repo**, damit das AppStream-Repository auf http://classroom.example.com/content/rhel8.2/x86_64/dvd/AppStream/ verweist. Der Repository-Name sollte auf Appstream gesetzt werden.
 - Ändern Sie den Befehl **url** so, dass http://classroom.example.com/content/rhel8.2/x86_64/dvd/ als Installationsquelle verwendet wird.
 - Kommentieren Sie den Befehl **network** aus.
 - Ändern Sie den Befehl **rootpw** für die Verwendung von **Klartext** und legen Sie das Root-Passwort auf **redhat** fest.
 - Löschen Sie die Zeile, die den Befehl **auth** enthält, und fügen Sie die Zeile **authselect select sssd** zum Festlegen des Service **sssd** als Identitäts- und Authentifizierungsquelle hinzu.
 - Vereinfachen Sie den Befehl **services**, sodass nur die Services **kdump** und **rhsmdcertd** deaktiviert sind. Lassen Sie nur **sshd**, **rngd** und **chrony** aktiviert.
 - Fügen Sie den Befehl **autopart** hinzu. Die Befehle **part** sollten bereits auskommentiert sein.
 - Vereinfachen Sie den Abschnitt „%post“ so, dass nur ein Skript ausgeführt wird, um den Text **Kickstarted on DATE** an das Ende der Datei **/etc/issue** anzufügen. **DATE** ist eine variable Information und sollte vom Skript mit dem Befehl **date** ohne weitere Optionen generiert werden.
 - Vereinfachen Sie den Abschnitt **%package** wie folgt: Beziehen Sie die Pakete @core, **chrony**, **dracut-config-generic**, **dracut-norescue**, **firewalld**, **grub2**, **kernel**, **rsync**, **tar** und **httpd** ein. Stellen Sie sicher, dass das Paket **plymouth** nicht installiert wird.

- 2.1. Kommentieren Sie die „reboot“-Anweisung aus:

```
#reboot
```

- 2.2. Der Befehl **repo** wird zweimal in **kickstart.cfg** gefunden. Kommentieren Sie den Befehl **repo** für das BaseOS-Repository aus. Ändern Sie den Befehl **repo**, damit das AppStream-Repository auf das AppStream-Repository des Kursraums verweist.

```
#repo --name="koji-override-0" --baseurl=http://download-
node-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.2.0-
updates-20200423.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.2/
x86_64/dvd/AppStream/
```

- 2.3. Ändern Sie den Befehl **url**, um die im Kursraum verwendeten Quellmedien für die Installation per HTTP anzugeben:

```
url --url="http://classroom.example.com/content/rhel8.2/x86_64/dvd/"
```

- 2.4. Kommentieren Sie den Befehl **network** aus:

```
#network --bootproto=dhcp --device=link --activate
```

- 2.5. Legen Sie das Root-Passwort auf **redhat** fest. Ändern Sie die Zeile, die mit **rootpw** beginnt, wie folgt:

```
rootpw --plaintext redhat
```

- 2.6. Löschen Sie die Zeile, die den Befehl **auth** enthält, und fügen Sie die Zeile **authselect select sssd** zum Festlegen des Service **sssd** als Identitäts- und Authentifizierungsquelle hinzu.

```
authselect select sssd
```

- 2.7. Vereinfachen Sie den Befehl **services** so, dass er genau wie folgt aussieht:

```
services --disabled="kdump, rhsmcertd" --enabled="sshd, rngd, chronyd"
```

- 2.8. Kommentieren Sie die Befehle **part** aus. Fügen Sie den Befehl **autopart** hinzu:

```
# Disk partitioning information
#part biosboot --fstype="biosboot" --size=1
#part /boot/efi --fstype="efi" --size=100 --fsoptions="..."
#part / --fstype="xfs" --size=10137 --label=root
autopart
```

- 2.9. Löschen Sie den gesamten Inhalt zwischen **%post** und **%end**. Fügen Sie folgende Zeile hinzu: **echo "Kickstarted on \$(date)" >> /etc/issue**

Der Abschnitt **%post** sollte wie folgt aussehen.

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

- 2.10. Vereinfachen Sie die Paketspezifikation, sodass sie wie folgt aussieht:

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
```

```
httpd  
-plymouth  
%end
```

3. Überprüfen Sie die Syntax von **kickstart.cfg**.

- 3.1. Führen Sie den Befehl **ksvalidator** aus, um die Kickstart-Datei auf Syntaxfehler zu prüfen.

```
[student@serverb ~]$ ksvalidator kickstart.cfg
```

4. Stellen Sie die Datei **/home/student/kickstart.cfg** unter **http://serverb.lab.example.com/ks-config/kickstart.cfg** zur Verfügung.

- 4.1. Kopieren Sie **kickstart.cfg** in das Verzeichnis **/var/www/html/ks-config/**.

```
[student@serverb ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

5. Kehren Sie zum System **workstation** zurück, um Ihre Arbeit zu überprüfen.

- 5.1. Beenden Sie **serverb**.

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

Bewertung

Führen Sie auf **workstation** das Skript **lab installing-review grade** aus, um diese Übung zu bewerten. Booten Sie **servera** neu, um eine Kickstart-Installation durchzuführen.

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

Korrigieren Sie alle Fehler in **kickstart.cfg**, die vom **serverb**-Webserver freigegeben wird, indem Sie entweder **/var/www/html/ks-config/kickstart.cfg** direkt ändern oder **~/kickstart.cfg** ändern und sie nach **/var/www/html/ks-config/** kopieren.

Booten Sie **servera** neu, um eine Kickstart-Installation durchzuführen. Wählen Sie im GRUB-Menü **Kickstart Red Hat Enterprise Linux 8** aus, und drücken Sie die **Eingabetaste**.

Beenden

Führen Sie auf **workstation** das Skript **lab installing-review finish** aus, um diese Übung abzuschließen. Dieses Skript entfernt den während der Übung auf **serverb** konfigurierten Webserver.

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

Setzen Sie das System **servera** auf den Standardzustand zurück.

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Die binäre RHEL 8-DVD enthält Anaconda und alle Repositorys, die für die Installation erforderlich sind.
- Die RHEL 8-ISO-Startdatei enthält das Anaconda-Installationsprogramm, das während der Installation über das Netzwerk auf Repositorys zugreift.
- Das Kickstart-System führt unbeaufsichtigte Installationen durch.
- Kickstart-Dateien können mithilfe der Kickstart Generator-Website oder durch Kopieren und Bearbeiten von **/root/anaconda-ks.cfg** erstellt werden.
- Das Yum-Modul *virt* stellt die Pakete bereit, damit ein RHEL-System zu einem Virtualisierungshost wird.
- Das Paket *cockpit-machines* fügt Cockpit das Menü **Virtual Machines** hinzu.

Kapitel 13

Ausführen von Containern

Ziel

Abrufen, Ausführen und Verwalten einfacher kompakter Services als Container auf einem einzelnen Red Hat Enterprise Linux-Server

Zielsetzungen

- Erklären des Containerkonzepts und dessen Verwendung zum Verwalten und Bereitstellen von Anwendungen mit unterstützenden Softwarebibliotheken und -abhängigkeiten
- Installieren von Container-Managementtools und Ausführen eines einfachen Containers ohne Root
- Suchen, Abrufen, Überprüfen und Verwalten der von einer Remote-Container-Registry abgerufenen und auf Ihrem Server gespeicherten Container-Images
- Ausführen von Containern mit erweiterten Optionen, Auflisten der im System ausgeführten Container sowie Starten, Anhalten und Beenden von Containern
- Bereitstellen von persistentem Storage für Containerdaten durch das Mounten eines Verzeichnisses vom Containerhost in einem ausgeführten Container
- Starten, Anhalten und Überprüfen des Status eines Containers als systemd-Service

Abschnitte

- Einführung in Container (und Test)
- Ausführen eines einfachen Containers (und angeleitete Übung)
- Suchen und Verwalten von Container-Images (und angeleitete Übung)
- Durchführen des erweiterten Container-Managements (und angeleitete Übung)
- Anhängen von persistentem Storage an einen Container (und angeleitete Übung)
- Verwalten von Containern als Services (und angeleitete Übung)

Praktische Übung

Ausführen von Containern

Einführung in Container

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten Sie erklären können, was ein *Container* ist und wie er verwendet wird, um Anwendungen mit unterstützenden Softwarebibliotheken und -abhängigkeiten zu verwalten und bereitzustellen.

Einführung in die Containertechnologie

Softwareanwendungen hängen in der Regel von anderen Bibliotheken, Konfigurationsdateien oder Services ab, die von ihrer Laufzeitumgebung bereitgestellt werden. In der Regel ist die Laufzeitumgebung für eine Softwareanwendung in einem Betriebssystem installiert, das auf einem physischen Host oder virtuellem Rechner ausgeführt wird. Alle Anwendungsabhängigkeiten werden zusammen mit dem Betriebssystem auf dem Host installiert.

In Red Hat Enterprise Linux werden Verpackungssysteme wie RPM zum Verwalten von Anwendungsabhängigkeiten verwendet. Wenn Sie das Paket **httpd** installieren, wird durch das RPM-System sichergestellt, dass auch die korrekten Bibliotheken und anderen Abhängigkeiten für dieses Paket installiert sind.

Der Hauptnachteil von herkömmlich implementierten Softwareanwendungen besteht darin, dass die Abhängigkeiten der Anwendung mit der Laufzeitumgebung verflochten sind. Für eine Anwendung sind möglicherweise Versionen der unterstützenden Software erforderlich, die älter oder neuer sind als die im Betriebssystem bereitgestellte Software. Ebenso benötigen zwei Anwendungen auf demselben System möglicherweise unterschiedliche Versionen derselben Software, die nicht miteinander kompatibel sind.

Zum Lösen dieses Konflikts kann etwa die Anwendung als Container verpackt und bereitgestellt werden. Ein Container ist ein Satz aus einem oder mehreren Prozessen, die vom Rest des Systems isoliert sind.

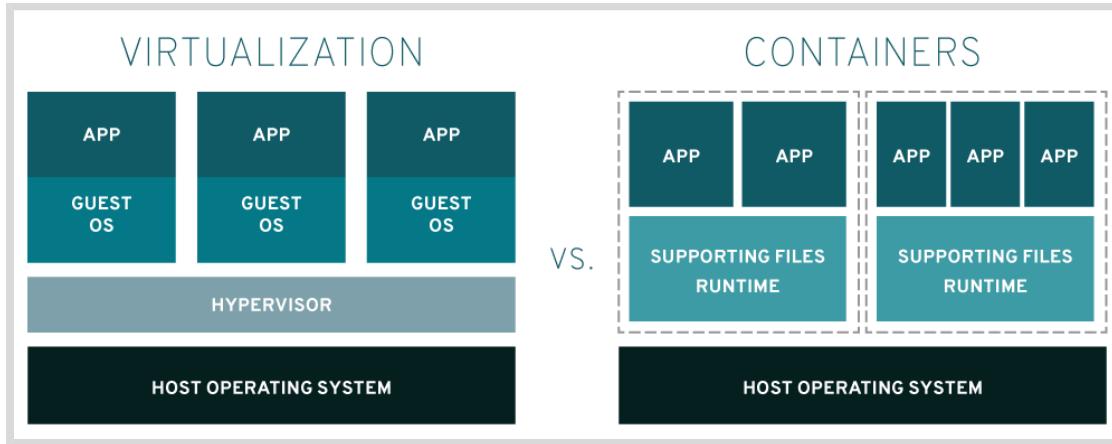
Stellen Sie sich einen physischen Versandcontainer vor. Ein Versandcontainer ist eine Standardmethode zum Verpacken und Versenden von Waren. Er wird etikettiert, beladen, entladen und als Einzelbehältnis zwischen zwei Standorten transportiert. Die Inhalte des Containers sind vom Inhalt anderer Container isoliert, sodass Sie sich nicht gegenseitig beeinflussen.

Mithilfe von Softwarecontainern können Anwendungspakete erstellt werden, um die Bereitstellung und Verwaltung zu vereinfachen.

Container und virtuelle Rechner im Vergleich

Container bieten viele der Vorteile virtueller Rechner wie Sicherheit, Storage und Netzwerkskopulation.

Beide Technologien isolieren Ihre Anwendungsbibliotheken und Laufzeitressourcen vom Hostbetriebssystem oder Hypervisor und umgekehrt.



Container und virtuelle Rechner unterscheiden sich dahingehend, wie Sie mit der Hardware und dem zugrunde liegenden Betriebssystem interagieren.

Virtualisierung:

- Ermöglicht die gleichzeitige Ausführung mehrerer Betriebssysteme auf einer einzelnen Hardwareplattform.
- Verwendet einen Hypervisor, um Hardware in mehrere virtuelle Hardwaresysteme einzuteilen, sodass mehrere Betriebssysteme nebeneinander ausgeführt werden können.
- Erfordert eine vollständige Betriebssystemumgebung, um die Anwendung zu unterstützen.

Vergleichen Sie dies mit Containern, die:

- Direkt auf dem Betriebssystem ausgeführt werden und die Hardware- und Betriebssystemressourcen über alle Container im System hinweg gemeinsam verwenden. So können Anwendungen kompakt bleiben und parallel ausgeführt werden.
- denselben Betriebssystemkernel gemeinsam verwenden, die Container-Anwendungsprozesse vom Rest des Systems isolieren und mit diesem Kernel kompatible Software verwenden.
- Viel weniger Hardweareressourcen voraussetzen als virtuelle Rechner, wodurch sie zudem schneller gestartet und angehalten werden können und sich die Storage-Anforderungen verringern.



Anmerkung

Einige Anwendungen sind möglicherweise nicht geeignet, als Container ausgeführt zu werden. Beispielsweise benötigen Anwendungen, die auf Hardwareinformationen auf niedriger Ebene zugreifen, einen direkteren Hardwarezugriff als Container im Allgemeinen bereitstellen.

Untersuchen der Implementierung von Containern

Red Hat Enterprise Linux implementiert Container mit Kerntechnologien wie:

- Kontrollgruppen (*cgroups*) für das Ressourcenmanagement
- Namespaces für die Prozessisolierung
- SELinux und Seccomp (Secure Computing Mode) zum Erzwingen von Sicherheitsgrenzen



Anmerkung

Eine ausführlichere Erläuterung der Containerarchitektur und -sicherheit finden Sie im Whitepaper „Ten layers of container security“ (Zehn Schichten der Containersicherheit) [[https://www.redhat.com/en/resources/container-security-openshift-cloud-devops-whitepaper](https://www.redhat.com/en/resources/container-securityOpenshift-Cloud-DevOps-Whitepaper)].

Planen für Container

Mit Containern lassen sich gehostete Anwendungen effizient wiederverwenden und portieren. Sie können problemlos von einer Umgebung in eine andere verschoben werden, beispielsweise aus der Entwicklung bis zur Produktion. Sie können mehrere Versionen eines Containers speichern und nach Bedarf schnell auf jede einzelne Version zugreifen.

Container sind in der Regel temporär oder *kurzlebig*. Sie können die durch die Ausführung eines Containers generierten Daten dauerhaft im persistenten Storage speichern. Die Container werden jedoch in der Regel bei Bedarf ausgeführt und werden anschließend angehalten und entfernt. Ein neuer Containerprozess wird gestartet, sobald der betreffende Container das nächste Mal benötigt wird.

Ausführen von Containern über Container-Images

Container werden über *Container-Images* ausgeführt. Container-Images dienen als Blueprints zum Erstellen von Containern.

In Container-Images werden Anwendungspakete mit allen zugehörigen Abhängigkeiten erstellt, beispielsweise:

- Systembibliotheken
- Laufzeiten für Programmiersprachen
- Bibliotheken für Programmiersprachen
- Konfigurationseinstellungen
- Statische Datendateien

Container-Images sind unveränderlich oder *unveränderliche* Dateien, in denen der gesamte Code und die Abhängigkeiten enthalten sind, die zum Ausführen eines Containers erforderlich sind.

Container-Images werden entsprechend den Spezifikationen erstellt, etwa entsprechend der Image-Formatspezifikation der Open Container Initiative (OCI). Diese Spezifikationen definieren das Format für Container-Images sowie die Metadaten über die Container-Hostbetriebssysteme und Hardwarearchitekturen, die das Image unterstützt.

Entwerfen containerbasierter Architekturen

Sie könnten eine komplexe, aus mehreren Services bestehende Softwareanwendung in einem einzelnen Container installieren. Beispielsweise verfügen Sie möglicherweise über einen Webserver, der eine Datenbank und ein Messaging-System verwenden muss. Die Verwendung eines Containers für mehrere Services ist jedoch schwer zu verwalten.

Bei einem besseren Design werden die einzelnen Komponenten, der Webserver, die Datenbank und das Messaging-System in separaten Containern ausgeführt. So wirken sich für einzelne Anwendungskomponenten vorgenommen Aktualisierungen und Wartungsaufgaben weder auf andere Komponenten noch auf den Anwendungs-Stack aus.

Verwalten von Containern mit Podman

Zum Kennenlernen von Containern empfiehlt es sich, mit einzelnen Containern auf einem einzelnen Server zu arbeiten, der als Container-Host fungiert. Red Hat Enterprise Linux bietet eine Reihe von Container-Tools, die dazu verwendet werden können, darunter:

- **podman** zum direkten Verwalten von Containern und Container-Images.
- **skopeo** zum Überprüfen, Kopieren, Löschen und Signieren von Images.
- **buildah** zum Erstellen von neuen Container-Images.

Diese Tools sind OCI-konform (Open Container Initiative). Sie können verwendet werden, um die von OCI-konformen Container-Engines wie Docker erstellten Linux-Container zu verwalten. Diese Tools sind speziell für die Ausführung von Containern unter Red Hat Enterprise Linux auf einem Container-Host mit einem Knoten konzipiert.

In diesem Kapitel verwenden Sie die Befehle **podman** und **skopeo**, um Container und vorhandene Container-Images auszuführen und zu verwalten.



Anmerkung

Die Verwendung von **buildah** zum Erstellen eigener Container-Images geht über den Umfang dieses Kurses hinaus, wird jedoch im Red Hat-Trainingskurs *Red Hat OpenShift I: Containers & Kubernetes* (DO180) behandelt.

Ausführen von Containern ohne Root

Auf dem Container-Host können Sie Container als root-Benutzer oder als normaler, unprivilegierter Benutzer ausführen. Von unprivilegierten Benutzern ausgeführte Container werden als *Container ohne Root* bezeichnet.

Container ohne Root sind sicherer, verfügen jedoch über einige Einschränkungen. So sind Container ohne Root beispielsweise nicht in der Lage, ihre Netzwerkservices über die privilegierten Ports (unter Port 1024) des Containers-Hosts zu veröffentlichen.

Bei Bedarf können Sie Container direkt als **root** ausführen, was jedoch die Sicherheit des Systems etwas schwächt, wenn ein Angreifer infolge eines Fehlers den Container kompromittieren kann.

Angemessenes Verwalten von Containern

Neue Anwendungen implementieren in zunehmendem Maße funktionale Komponenten mit Containern. Diese Container bieten Services, die von anderen Teilen der Anwendung verwendet werden. In einem Unternehmen kann die Verwaltung einer wachsenden Anzahl von Containern schnell zu einer überwältigenden Aufgabe werden.

Damit Container in der Produktionsumgebung bereitgestellt werden können, muss diese an einige der folgenden Herausforderungen angepasst werden können:

- Die Plattform muss die Verfügbarkeit von Containern gewährleisten, die den Kunden wesentliche Services bieten.
- Die Umgebung muss auf Anwendungsauslastungsspitzen reagieren, indem die Anzahl der ausgeführten Container erhöht oder verringert und der Datenverkehr ausgeglichen wird.
- Die Plattform sollte den Ausfall eines Containers oder eines Hosts ermitteln und entsprechend reagieren.

- Entwickler benötigen möglicherweise einen automatisierten Workflow, um Kunden neue Anwendungsversionen transparent und sicher bereitzustellen.

Kubernetes ist ein Orchestrierungsservice, der die Bereitstellung, Verwaltung und Skalierung containerbasierter Anwendungen in einem Cluster von Container-Hosts erleichtert. Er unterstützt die Verwaltung von DNS-Aktualisierungen beim Starten neuer Container. Er unterstützt das Umleiten von Datenverkehr an Ihre Container mit einem Load Balancer, womit Sie die Anzahl der Container, die einen Service manuell oder automatisch bereitstellen, hoch- und herunterskalieren können. Zudem unterstützt er benutzerdefinierte Health Checks, um Ihre Container zu überwachen und sie neu zu starten, wenn sie ausfallen.

Red Hat bietet eine Distribution von Kubernetes namens *Red Hat OpenShift*. OpenShift umfasst mehrere modulare Komponenten und Services, die auf einer Kubernetes-Infrastruktur basieren. Es bietet zusätzliche Features. Dazu zählen u. a. das webbasierte Remote-Management, Multi-Tenancy, Überwachung und Auditing, Lebenszyklusverwaltung für Anwendungen sowie Self-Service-Instanzen für Entwickler.

Red Hat OpenShift wird in diesem Kurs nicht behandelt. Weitere Informationen dazu finden Sie jedoch unter <https://www.openshift.com>.



Anmerkung

In Unternehmen werden einzelne Container nicht generell über die Befehlszeile ausgeführt. Stattdessen sollten Container in der Produktion mit einer auf Kubernetes basierenden Plattform wie Red Hat OpenShift ausgeführt werden.

Sie müssen jedoch möglicherweise Befehle verwenden, um mit Containern und Images manuell oder in einem kleinen Maßstab zu arbeiten. Dazu können Sie auf einem Red Hat Enterprise Linux 8-System eine Reihe von Containertools installieren.

In diesem Kapitel liegt der Fokus auf diesem Anwendungsfall, damit Sie die grundlegenden Konzepte hinter Containern, ihre Funktionsweise und ihre Nützlichkeit besser verstehen.



Literaturhinweise

Man Pages **cGroups(7)**, **Namespaces(7)**, **Seccomp(2)**.

Image-Spezifikation der Open Container Initiative (OCI)

<https://github.com/opencontainers/image-spec/blob/master/spec.md>

Weitere Informationen finden Sie im Kapitel *Starting with containers* im Handbuch *Red Hat Enterprise Linux 8 Building, Running, and Managing Containers* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/building_running_and_managing_containers/index#starting-with-containers_building-running-and-managing-containers

► Quiz

Einführung in Container

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welches zum Ausführen von Containern verwendete Tool wird von Red Hat Enterprise Linux bereitgestellt?
- a. buildah
 - b. container
 - c. podman
 - d. skopeo
- 2. Welche zwei der folgenden Aussagen beschreiben die Containertechnologie? (Wählen Sie zwei Antworten aus.)
- a. Container erstellen vollständige Betriebssystempakete wie virtuelle Rechner.
 - b. Container führen einen Satz aus einem oder mehreren Prozessen aus, die vom Rest des Systems isoliert sind.
 - c. Jeder Container enthält einen eigenen Kernel.
 - d. Container bietet eine Standardmöglichkeit zum Erstellen von Anwendungspaketen, um die Bereitstellung und das Management zu vereinfachen.
- 3. Welche zwei der folgenden Aussagen über Container-Images sind richtig? (Wählen Sie zwei Antworten aus.)
- a. In Container-Images wird ein Anwendungspaket mit allen zugehörigen Laufzeitabhängigkeiten erstellt.
 - b. Container-Image, die Docker verwenden, können Podman nicht verwenden.
 - c. Container-Images können nur auf einem Container-Host ausgeführt werden, der mit der exakt gleichen Version der Software im Image installiert ist.
 - d. Container-Images dienen als Blueprints zum Erstellen von Containern.
- 4. Welche drei Kerntechnologien werden für die Implementierung von Containern in Red Hat Enterprise Linux-Container verwendet? (Wählen Sie drei Antworten aus.)
- a. Hypervisor-Code für das Hosten von VMs
 - b. Kontrollgruppen (cGroups) für das Ressourcenmanagement
 - c. Namespaces für die Prozessisolierung
 - d. Vollständiges Betriebssystem für Kompatibilität mit dem Host des Containers
 - e. SELinux und Seccomp für Sicherheitszwecke

► Lösung

Einführung in Container

Wählen Sie die richtigen Antworten auf die folgenden Fragen aus:

- 1. Welches zum Ausführen von Containern verwendete Tool wird von Red Hat Enterprise Linux bereitgestellt?
- a. buildah
 - b. container
 - c. podman
 - d. skopeo
- 2. Welche zwei der folgenden Aussagen beschreiben die Containertechnologie? (Wählen Sie zwei Antworten aus.)
- a. Container erstellen vollständige Betriebssystempakete wie virtuelle Rechner.
 - b. Container führen einen Satz aus einem oder mehreren Prozessen aus, die vom Rest des Systems isoliert sind.
 - c. Jeder Container enthält einen eigenen Kernel.
 - d. Container bietet eine Standardmöglichkeit zum Erstellen von Anwendungspaketen, um die Bereitstellung und das Management zu vereinfachen.
- 3. Welche zwei der folgenden Aussagen über Container-Images sind richtig? (Wählen Sie zwei Antworten aus.)
- a. In Container-Images wird ein Anwendungspaket mit allen zugehörigen Laufzeitabhängigkeiten erstellt.
 - b. Container-Image, die Docker verwenden, können Podman nicht verwenden.
 - c. Container-Images können nur auf einem Container-Host ausgeführt werden, der mit der exakt gleichen Version der Software im Image installiert ist.
 - d. Container-Images dienen als Blueprints zum Erstellen von Containern.
- 4. Welche drei Kerntechnologien werden für die Implementierung von Containern in Red Hat Enterprise Linux-Container verwendet? (Wählen Sie drei Antworten aus.)
- a. Hypervisor-Code für das Hosten von VMs
 - b. Kontrollgruppen (cGroups) für das Ressourcenmanagement
 - c. Namespaces für die Prozessisolierung
 - d. Vollständiges Betriebssystem für Kompatibilität mit dem Host des Containers
 - e. SELinux und Seccomp für Sicherheitszwecke

Ausführen eines einfachen Containers

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten Sie Container-Management-Tools installieren und einen einfachen Container ohne Root ausführen können.

Installieren von Container-Management-Tools

Damit Sie Container auf Ihrem System ausführen und verwalten können, müssen Sie zunächst die erforderlichen Befehlszeilen-Tools installieren. Installieren Sie das Modul `container-tools` über den Befehl `yum`.

```
[root@host ~]# yum module install container-tools
```

Das Modul `container-tools` enthält Softwarepakete, die mehrere Tools installieren. Die in diesem Kapitel verwendeten Tools sind **podman** und **skopeo**.



Anmerkung

Standardmäßig installiert das System die *fast stream*-Tools, **container-tools:rhel8**, die alle drei Monate basierend auf der neuesten, stabilen Upstream-Version der Container-Tools aktualisiert werden.

Alternative *Stable-Streams*, die eine bestimmte Version der Tools sperren, erhalten keine Funktionsaktualisierungen. Red Hat plant einmal im Jahr die Veröffentlichung neuer Stable-Streams, die zwei Jahre lang unterstützt werden.

Auswählen von Container-Images und -Registries

Eine Container-Registry ist ein Repository zum Speichern und Abrufen von Container-Images. Container-Images werden von einem Entwickler in eine Container-Registry hochgeladen oder übertragen. Sie können diese Container-Images aus der Registry auf ein lokales System herunterladen oder abrufen, damit Sie sie zum Ausführen von Containern verwenden können.

Sie können eine öffentliche Registry verwenden, die Images von Drittanbietern enthält, oder Sie können eine private Registry verwenden, die von Ihrer Organisation kontrolliert wird. Die Quelle Ihrer Container-Images ist von Bedeutung. Wie bei jedem anderen Softwarepaket müssen Sie wissen, ob Sie dem Code im Container-Image vertrauen können. Unterschiedliche Registries weisen dahingehend unterschiedliche Richtlinien auf, ob und wie die ihnen gesendeten Container-Images bereitgestellt, bewertet und getestet werden.

Red Hat verteilt zertifizierte Container-Images über zwei Haupt-Container-Registries, auf die Sie mit ihren Red Hat Anmeldedaten zugreifen können.

- **registry.redhat.io** für Container, die auf offiziellen Red Hat-Produkten basieren.
- **registry.connect.redhat.com** für Container, die auf Drittanbieterprodukten basieren.

Red Hat lässt die ältere Registry **registry.access.redhat.com** allmählich ablaufen.

Der Red Hat Container Catalog (<https://access.redhat.com/containers>) bietet eine webbasierte Oberfläche, mit der Sie diese Registrys nach zertifizierten Inhalten durchsuchen können.



Anmerkung

In diesem Kursraum wird eine auf Red Hat Quay basierende private Registry ausgeführt, um Container-Images bereitzustellen. Weitere Informationen zu dieser Software finden Sie unter <https://access.redhat.com/products/red-hat-quay>.

Container-Benennungskonventionen

Container-Images werden anhand der folgenden Syntax für vollqualifizierte Image-Namen benannt:

registry_name/user_name/image_name:tag

- Bei **registry_name** handelt es sich um den Namen der das Image speichernden Registry. Hierbei handelt es sich in der Regel um den vollständig qualifizierten Domain-Namen der Registry.
- Der **user_name** steht für den Benutzer oder für die Organisation, wozu das Image gehört.
- Der **image_name** muss im Benutzer-Namespace eindeutig sein.
- Das **Tag** gibt die Image-Version an. Wenn der Image-Name kein Image-Tag enthält, dann wird **latest** angenommen.

Ausführen von Containern

Zum Ausführen eines Containers auf Ihrem lokalen System müssen Sie zunächst ein Container-Image abrufen. Verwenden Sie Podman, um ein Image aus einer Registry abzurufen. Beim Abrufen von Images sollte immer der vollständig qualifizierte Image-Name verwendet werden. Mit dem Befehl **podman pull** wird das von Ihnen angegebene Image aus der Registry abgerufen und lokal gespeichert:

```
[user@host ~]$ podman pull registry.access.redhat.com/ubi8/ubi:latest
Trying to pull registry.access.redhat.com/ubi8/ubi:latest...Getting image source
signatures
Copying blob 77c58f19bd6e: 70.54 MiB / 70.54 MiB [=====] 10s
Copying blob 47db82df7f3f: 1.68 KiB / 1.68 KiB [=====] 10s
Copying config a1f8c9699786: 4.26 KiB / 4.26 KiB [=====] 0s
Writing manifest to image destination
Storing signatures
a1f8c969978652a6d1b2dfb265ae0c6c346da69000160cd3ecd5f619e26fa9f3
```

Nach dem Abruf speichert Podman die Images lokal. Sie können diese mit dem Befehl **podman images** auflisten:

```
[user@host ~]$ podman images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
registry.access.redhat.com/ubi8/ubi    latest   a1f8c9699786   5 weeks ago  211 MB
```

In der obigen Ausgabe ist zu sehen, dass das Image-Tag **latest** lautet und dass die Image-ID **a1f8c9699786** ist.

Verwenden Sie den Befehl **podman run**, um einen Container aus diesem Image auszuführen. Beim Ausführen des Befehls **podman run** wird ein neuer Container aus einem Container-Image erstellt und gestartet. Verwenden Sie bei Bedarf Optionen vom Typ **-it**, um mit dem Container zu interagieren. Optionen vom Typ **-it** teilen dem Container ein Terminal zu und ermöglichen es Ihnen, Tastatureingaben zu senden.

```
[user@host ~]$ podman run -it registry.access.redhat.com/ubi8/ubi:latest  
[root@8b032455db1a ~]#
```

Wichtig

Wenn Sie einen Container mit dem vollständig qualifizierten Image-Namen ausführen, das Image jedoch noch nicht lokal gespeichert ist, wird über den Befehl **podman run** zunächst das Image aus der Registry abgerufen und dann ausgeführt.



Anmerkung

Viele Podman-Flags haben auch eine alternative Langform. Einige davon werden im Folgenden erklärt.

- **-t** entspricht **--tty**, d. h., für den Container muss **pseudo-tty** (Pseudo-Terminal) zugeordnet sein.
- **-i** entspricht **--interactive**. Wenn diese Option verwendet wird, akzeptiert der Container die Standardeingabe.
- Bei **-d** oder der zugehörigen Langform **--detach** wird der Container im Hintergrund (getrennt) ausgeführt. Wenn diese Option verwendet wird, führt Podman den Container im Hintergrund aus und zeigt die generierte Container-ID an.

Eine vollständige Liste mit den Flags finden Sie auf der Man Page **podman-run(1)**.

Beim Verweisen auf den Container erkennt Podman den Container-Namen oder die generierte Container-ID. Verwenden Sie die Option **--name**, um den Container-Namen beim Ausführen des Containers mit Podman festzulegen. Container-Namen müssen eindeutig sein. Wenn im Befehl **podman run** kein Container-Name enthalten ist, generiert Podman einen eindeutigen Zufallsnamen.

Im folgenden Beispiel wird dem Container ein Name zugewiesen. Anschließend wird *innerhalb* des Containers ein Bash-Terminal ausdrücklich gestartet und darin ein Befehl ausgeführt:



Anmerkung

Beachten Sie, dass das Tag **latest** angenommen wird, wenn kein Tag explizit angegeben ist.

Der Befehl im nächsten Beispiel wird in einer einzelnen Zeile eingegeben.

```
[user@host ~]$ podman run -it --name=rhel8 registry.access.redhat.com/ubi8/ubi /bin/bash
[root@c20631116955 ~]# cat /etc/os-release
NAME="Red Hat Enterprise Linux"
VERSION="8.2 (Ootpa)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="8.2"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Red Hat Enterprise Linux 8.2 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8.2:GA"
HOME_URL="https://www.redhat.com/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_BUGZILLA_PRODUCT_VERSION=8.2
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.2"
[root@c20631116955 ~]# exit
exit
[user@host ~]$
```

Sie können auch einen Schnellbefehl in einem Container ausführen, ohne mit ihm zu interagieren, und dann den Container entfernen, sobald der Befehl abgeschlossen ist. Verwenden Sie dazu **podman run --rm**, gefolgt vom Container-Image und einem Befehl.

```
[user@host ~]$ podman run --rm registry.access.redhat.com/ubi8/ubi cat /etc/os-release
NAME="Red Hat Enterprise Linux"
VERSION="8.2 (Ootpa)"
ID="rhel"
ID_LIKE="fedora"
VERSION_ID="8.2"
PLATFORM_ID="platform:el8"
PRETTY_NAME="Red Hat Enterprise Linux 8.2 (Ootpa)"
ANSI_COLOR="0;31"
CPE_NAME="cpe:/o:redhat:enterprise_linux:8.2:GA"
HOME_URL="https://www.redhat.com/"
BUG_REPORT_URL="https://bugzilla.redhat.com/"

REDHAT_BUGZILLA_PRODUCT="Red Hat Enterprise Linux 8"
REDHAT_BUGZILLA_PRODUCT_VERSION=8.2
REDHAT_SUPPORT_PRODUCT="Red Hat Enterprise Linux"
REDHAT_SUPPORT_PRODUCT_VERSION="8.2"
[user@host ~]$
```

Analysieren der Container-Isolation

Container bieten eine Laufzeitisolation von Ressourcen. Container verwenden Linux-Namespace, um separate, isolierte Umgebungen für Ressourcen bereitzustellen, z. B. Prozesse, Netzwerkkommunikation und Volumes. Prozesse, die in einem Container ausgeführt werden, sind von allen anderen Prozessen auf dem Hostrechner isoliert.

Zeigen Sie die Prozesse an, die im Container ausgeführt werden:

```
[user@host ~]$ podman run -it registry.access.redhat.com/ubi7/ubi /bin/bash
[root@ef2550ed815d /]# ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  4.5  0.1 11840  2904 pts/0    Ss   22:10  0:00 /bin/bash
root           15  0.0  0.1 51768  3388 pts/0    R+   22:10  0:00 ps aux
```

Beachten Sie, dass sich der Benutzername und die ID innerhalb des Containers vom Benutzernamen und der ID auf dem Hostrechner unterscheiden:

```
[root@ef2550ed815d /]# id
uid=0(root) gid=0(root) groups=0(root)
[root@ef2550ed815d /]# exit
exit
[user@host ~]$ id
uid=1000(user) gid=1000(user) groups=1000(user),10(wheel)
```



Literaturhinweise

Man Pages **podman-pull(1)**, **podman-images(1)** und **podman-run(1)**.

Weitere Informationen finden Sie im Kapitel *Starting with containers* im Handbuch *Red Hat Enterprise Linux 8 Building, Running, and Managing Containers* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/building_running_and_managing_containers/index#starting-with-containers_building-running-and-managing-containers

► Angeleitete Übung

Ausführen eines einfachen Containers

In dieser Übung installieren Sie Container-Tools und testen sie, indem Sie einen einfachen Container ohne Root ausführen.

Ergebnisse

Sie sollten in der Lage sein, Container-Management-Tools zu installieren und sie zum Ausführen eines Containers zu verwenden.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-basic start** aus. Dieser Befehl führt ein Startskript aus, um zu bestimmen, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem wird die Container-Registry überprüft und konfiguriert und sichergestellt, dass das für diese Übung verwendete Container-Image dort gespeichert wird.

```
[student@workstation ~]$ lab containers-basic start
```

Anweisungen

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Installieren Sie das Yum-Modul *container-tools* mit dem Befehl **yum**.

```
[student@servera ~]$ sudo yum module install container-tools
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 3. Melden Sie sich über den Befehl **podman login** bei der Container-Registry an.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 4. Führen Sie den Befehl **podman pull** aus, um ein Container-Image aus der Registry mit dem vollqualifizierten Namen abzurufen.

```
[student@servera ~]$ podman pull registry.lab.example.com/rhel8/httpd-24:latest
Trying to pull registry.lab.example.com/rhel8/httpd-24:latest...
Getting image source signatures
Copying blob 77c58f19bd6e done
Copying blob 47db82df7f3f done
Copying blob 9d20433efa0c done
Copying blob 71391dc11a78 done
Copying config 7e93f25a94 done
Writing manifest to image destination
Storing signatures
7e93f25a946892c9c175b74a0915c96469e3b4845a6da9f214fd3ec19c3d7070
```

- 5. Führen Sie einen Container im Image aus, verbinden Sie ihn mit dem Terminal, weisen Sie ihm einen Namen zu und starten Sie dann eine interaktive Bash-Shell mit dem Befehl **podman run**. Da kein Tag angegeben ist, wird das Tag **latest** angenommen:

Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run --name myweb -it registry.lab.example.com/rhel8/
httpd-24 /bin/bash
bash-4.4$
```

- 6. Listen Sie aktive Prozesse innerhalb des Containers auf. Es werden nur die Prozesse angezeigt, die im Container ausgeführt werden. Es werden keine weiteren Prozesse angezeigt, die auf dem Server ausgeführt werden.

```
bash-4.4$ ps aux
USER        PID %CPU %MEM      VSZ      RSS TTY      STAT START   TIME COMMAND
default       1  6.6  0.1  12020   3120 pts/0      Ss  21:52   0:00 /bin/bash
default       6  0.0  0.1  44596   3296 pts/0      R+  21:52   0:00 ps aux
```

- 7. Zeigen Sie den aktuellen Benutzernamen und die ID im Container an.

```
bash-4.4$ id
uid=1001(default) gid=0(root) groups=0(root)
```

- 8. Beenden Sie die Container-Shell.

```
bash-4.4$ exit
exit
```

- 9. Führen Sie den Befehl **httpd -v** in einem Container aus. Verwenden Sie dazu das Container-Image **rhel8/httpd-24** und löschen Sie den Container, wenn der Befehl beendet wird:

Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run --rm registry.lab.example.com/rhel8/httpd-24 httpd
-v
Server version: Apache/2.4.37 (Red Hat Enterprise Linux)
Server built: Dec 2 2019 14:15:24
```

- 10. Beenden Sie **servera**.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-basic finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab containers-basic finish
```

Hiermit ist die angeleitete Übung beendet.

Suchen und Verwalten von Container-Images

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten Sie die von einer Remote-Container-Registry abgerufenen und auf Ihrem Server gespeicherten Container-Images suchen, abrufen, überprüfen und verwalten können.

Konfigurieren von Container-Registries

Podman verwendet die Datei **registries.conf** auf Ihrem Hostsystem, um Informationen über die Container-Registries zu erhalten, die verwendet werden können.

```
[user@host ~]$ cat /etc/containers/registries.conf
# This is a system-wide configuration file used to
# keep track of registries for various container backends.
# It adheres to TOML format and does not support recursive
# lists of registries.

# The default location for this configuration file is /etc/containers/
registries.conf.

# The only valid categories are: 'registries.search', 'registries.insecure',
# and 'registries.block'.

[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 = []

# If you need to block pull access from a registry, uncomment the section below
# and add the registries fully-qualified name.
#
[registries.block]
registries = []
```



Wichtig

Bei einem regulären Benutzer (ohne Root) von Podman ist diese Datei im Verzeichnis **\$HOME/.config/containers** gespeichert. Die Konfigurationseinstellungen in dieser Datei überschreiben die systemweiten Einstellungen in der Datei **/etc/containers/registries.conf**.

Die Liste der Registries, die Podman durchsuchen kann, wird im Abschnitt **[registries.search]** dieser Datei konfiguriert. Wenn Sie kein vollständig qualifiziertes Image in der Befehlszeile angeben, durchsucht Podman diesen Abschnitt in der angegebenen Reihenfolge, um zu bestimmen, wie ein vollständiger Image-Pfad zu bilden ist.

Durch Ausführen des Befehls **podman info** werden Konfigurationsinformationen für Podman angezeigt, darunter seine konfigurierten Registries.

```
[user@host ~]$ podman info
...output omitted...
insecure registries:
  registries: []
registries:
  registries:
    - registry.redhat.io
    - quay.io
    - docker.io
...output omitted...
```

Registry-Sicherheit

Unsichere Registries werden im Abschnitt **[registries.insecure]** der Datei **registries.conf** aufgelistet. Wenn eine Registry als unsicher aufgeführt ist, werden Verbindungen zu dieser Registry nicht mit TLS-Verschlüsselung geschützt. Wenn eine Registry durchsuchbar und unsicher ist, kann Sie in **[registries.search]** und **[registries.insecure]** aufgelistet sein.

Container-Registries können auch so konfiguriert werden, dass eine Authentifizierung erforderlich ist. Wie bereits besprochen, verwenden Sie den Befehl **podman login**, um sich bei einer Container-Registry anzumelden, die eine Authentifizierung erfordert.

Suchen von Container-Images

Verwenden Sie den Befehl **podman search**, um Container-Registries nach einem bestimmten Container-Image zu durchsuchen. Im folgenden Beispiel wird gezeigt, wie die Container-Registry **registry.redhat.io** auf alle Images durchsucht wird, in denen der Name **rhel8** enthalten ist:

```
[user@host ~]$ podman search registry.redhat.io/rhel8
INDEX      NAME          DESCRIPTION                  STARS  OFFICIAL   AUTOMATED
redhat.io  registry.redhat.io/openj9/openj9-8-rhel8  OpenJ9  1.8  OpenShift S2I
  image for Java Appl...  0
redhat.io  registry.redhat.io/openjdk/openjdk-8-rhel8  OpenJDK 1.8  Image for
  Java Applications base...  0
redhat.io  registry.redhat.io/openj9/openj9-11-rhel8  OpenJ9  11  OpenShift S2I
  image for Java Appl...  0
redhat.io  registry.redhat.io/openjdk/openjdk-11-rhel8  OpenJDK S2I  image for
  Java Applications on U...  0
redhat.io  registry.redhat.io/rhel8/memcached           Free and open source,
  high-performance, dist...  0
redhat.io  registry.redhat.io/rhel8/llvm-toolset        The LLVM back-end
  compiler and core librarie...  0
```

Kapitel 13 | Ausführen von Containern

```
redhat.io  registry.redhat.io/rhel8/rust-toolset      Rust and Cargo, which is
          a build system and ...  0
redhat.io  registry.redhat.io/rhel8/go-toolset        Golang compiler which
          will replace the curre...  0
...output omitted...
```

Führen Sie denselben Befehl mit der Option **--no-trunc** aus, um eine längere Image-Beschreibung anzuzeigen:

```
[user@host ~]$ podman search --no-trunc registry.access.redhat.com/rhel8
INDEX      NAME           DESCRIPTION             STARS   OFFICIAL   AUTOMATED
...output omitted...
redhat.io  registry.redhat.io/rhel8/nodejs-10       Node.js 10 available
          as container is a base platform for building and running various Node.js 10
          applications and frameworks. Node.js is a platform built on Chrome's JavaScript
          runtime for easily building fast, scalable network applications. Node.js uses
          an event-driven, non-blocking I/O model that makes it lightweight and efficient,
          perfect for data-intensive real-time applications that run across distributed
          devices.          0

redhat.io  registry.redhat.io/rhel8/python-36         Python 3.6 available
          as container is a base platform for building and running various Python 3.6
          applications and frameworks. Python is an easy to learn, powerful programming
          language. It has efficient high-level data structures and a simple but effective
          approach to object-oriented programming.          0

redhat.io  registry.redhat.io/rhel8/perl-526         Perl 5.26 available
          as container is a base platform for building and running various Perl 5.26
          applications and frameworks. Perl is a high-level programming language with roots
          in C, sed, awk and shell scripting. Perl is good at handling processes and files,
          and is especially good at handling text.          0
...output omitted...
```

Die folgende Tabelle enthält einige nützliche Optionen für den Befehl **podman search**:

Nützliche Podman-Suchoptionen

Option	Beschreibung
--limit <number>	Begrenzt die Anzahl der aufgelisteten Images pro Registry.

Option	Beschreibung
<code>--filter <filter=value></code>	<p>Filtert die Ausgabe basierend auf den angegebenen Bedingungen. Unterstützte Filter:</p> <ul style="list-style-type: none"> • stars=<number>: Nur Images mit mindestens dieser Anzahl an Sternen anzeigen. • is-automated=<true false>: Nur automatisch erstellte Images anzeigen. • is-official=<true false>: Nur Images anzeigen, die als offiziell gekennzeichnet sind.
<code>--tls-verify <true false></code>	<p>Aktiviert oder deaktiviert die HTTPS-Zertifikatvalidierung für alle verwendeten Registries. Default=true</p>

Verwenden von Red Hat Container Catalog

Red Hat verwaltet Repositorys mit zertifizierten Container-Images. Unter <https://access.redhat.com/containers> können Sie danach suchen.

Mit der Verwendung dieses Repositorys steigen Sicherheit und Zuverlässigkeit gegen Sicherheitslücken, die durch nicht getestete Images entstehen können. Der standardmäßige Befehl **podman** ist mit den Repositorys kompatibel, auf die im Red Hat Container Catalog verwiesen wird.

Überprüfen von Container-Images

Sie können Informationen zu einem Image anzeigen, bevor Sie es auf Ihr System herunterladen. Durch Ausführen des Befehls **skopeo inspect** kann ein Remote-Container-Image in einer Registry überprüft und Informationen dazu angezeigt werden.

Im folgenden Beispiel wird ein Container-Image überprüft, wobei Image-Informationen zurückgegeben werden, ohne dass das Image per Pull-Vorgang auf das lokale System übertragen wird:



Anmerkung

Durch Ausführen des Befehls **skopeo inspect** können verschiedene Image-Formate aus unterschiedlichen Quellen wie Remote-Registries oder lokale Verzeichnisse überprüft werden. Über den Transportmechanismus **docker://** wird **skopeo** angewiesen, eine Container-Image-Registry abzufragen.

```
[user@host ~]$ skopeo inspect docker://registry.redhat.io/rhel8/python-36
...output omitted...
      "name": "ubi8/python-36",
      "release": "107",
      "summary": "Platform for building and running Python 3.6
applications",
...output omitted...
```

Sie können auch lokal gespeicherte Image-Informationen überprüfen. Führen Sie dazu den Befehl **podman inspect** aus. Dieser Befehl bietet möglicherweise mehr Informationen als der Befehl **skopeo inspect**.

Listen Sie lokal gespeicherte Images auf:

```
[user@host ~]$ podman images
REPOSITORY                      TAG      IMAGE ID      CREATED        SIZE
quay.io/generic/rhel7            latest   1d3b6b7d01e4  3 weeks ago   688 MB
registry.redhat.io/rhel8/python-36    latest   e55cd9a2e0ca  6 weeks ago   811 MB
registry.redhat.io/ubi8/ubi       latest   a1f8c9699786  6 weeks ago   211 MB
```

Überprüfen Sie ein lokal gespeichertes Image und geben Sie Informationen zurück:

```
[user@host ~]$ podman inspect registry.redhat.io/rhel8/python-36
...output omitted...
      "Config": {
        "User": "1001",
        "ExposedPorts": {
          "8080/tcp": {}
        }
      ...output omitted...
      "name": "ubi8/python-36",
      "release": "107",
      "summary": "Platform for building and running Python 3.6
applications",
...output omitted...
```

Entfernen von lokalen Container-Images

Container-Images sind unveränderlich, ändern sich also nicht. Entsprechend werden alte Images nicht aktualisiert. Wenn Sie Software in einem Container aktualisieren möchten, muss ein neues Image verwendet werden, welches das alte ersetzt.

Wird ein aktualisiertes Image zur Verfügung gestellt, ändert der Publisher das Tag **latest**, sodass es mit dem neuen Image verknüpft wird. Sie können weiterhin auf ein älteres Image zugreifen, indem Sie auf sein spezifisches Versions-Tag verweisen, und Sie können Container darüber ausführen. Sie können auch das ältere Image entfernen, das neueste Image abrufen und nur das neueste (aktualisierte) Image verwenden, um Container auszuführen.

Beispielsweise profitieren die von Red Hat bereitgestellten Images von der langen Erfahrung von Red Hat in Bezug auf die Verwaltung von Sicherheitsschwachstellen und Defekten in Red Hat Enterprise Linux und anderen Produkten. Das Red Hat-Sicherheitsteam stärkt und kontrolliert diese qualitativ hochwertigen Images. Sie werden neu erstellt, sobald neue Schwachstellen entdeckt werden, und durchlaufen einen Qualitätssicherungsprozess.

Führen Sie den Befehl **podman rmi** aus, um ein lokal gespeichertes Image zu entfernen.

Listen Sie lokal gespeicherte Images auf:

```
[user@host ~]$ podman images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
quay.io/generic/rhel7    latest   1d3b6b7d01e4  3 weeks ago  688 MB
registry.redhat.io/rhel8/python-36    latest   e55cd9a2e0ca  6 weeks ago  811 MB
registry.redhat.io/ubi8/ubi    latest   a1f8c9699786  6 weeks ago  211 MB
```

Entfernen Sie das Image **registry.redhat.io/rhel8/python-36:latest**.

```
[user@host ~]$ podman rmi registry.redhat.io/rhel8/python-36:latest
e55cd9a2e0ca5f0f4e0249404d1abe3a69d4c6ffa5103d0512dd4263374063ad
[user@host ~]$
```

Listen Sie lokal gespeicherte Images auf, und verifizieren Sie, dass es entfernt wurde:

```
[user@host ~]$ podman images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
quay.io/generic/rhel7    latest   1d3b6b7d01e4  3 weeks ago  688 MB
registry.redhat.io/ubi8/ubi    latest   a1f8c9699786  6 weeks ago  211 MB
```



Literaturhinweise

Man Pages **podman-search(1)**, **podman-inspect(1)** und **skopeo(1)**.

Weitere Informationen finden Sie im Kapitel *Working with Container Images* im Handbuch *Red Hat Enterprise Linux 8 Building, Running, and Managing Containers* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/building_running_and_managing_containers/index#working-with-container-images_building-running-and-managing-containers

► Angeleitete Übung

Suchen und Verwalten von Container-Images

In dieser Übung verwenden Sie **podman**, um Container-Images auf Ihrem Server abzurufen, zu verwalten und zu löschen.

Ergebnisse

Sie sollten die von einer Remote-Container-Registry abgerufenen und auf Ihrem Server gespeicherten Container-Images suchen, abrufen, überprüfen und entfernen können.

Bevor Sie Beginnen

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-managing start** aus. Dieser Befehl führt ein Startskript aus, um zu bestimmen, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem wird die Container-Registry überprüft und konfiguriert und sichergestellt, dass das für diese Übung verwendete Container-Image dort gespeichert wird.

```
[student@workstation ~]$ lab containers-managing start
```

Anweisungen

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Zeigen Sie die Konfigurationsdatei der Container-Registry an, und zeigen Sie konfigurierte Registrys an.

```
[student@servera ~]$ cat /home/student/.config/containers/registries.conf
unqualified-search-registries = ['registry.lab.example.com']

[[registry]]
location = "registry.lab.example.com"
insecure = true
blocked = false
```

- 3. Suchen Sie mit dem Befehl **podman search** in der Registry nach Images mit einem Namen, der mit „ubi“ beginnt.

```
[student@servera ~]$ podman search registry.lab.example.com/ubi
INDEX      NAME          DESCRIPTION  STARS  OFFICIAL
example.com  registry.lab.example.com/ubi7/ubi        0
example.com  registry.lab.example.com/ubi8/ubi        0
```

- 4. Melden Sie sich über den Befehl **podman login** bei der Container-Registry an.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 5. Verwenden Sie den Befehl **skopeo inspect**, um Informationen zu einem Image in der Registry anzuzeigen, bevor Sie es herunterladen.

Der folgende **skopeo inspect**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ skopeo inspect docker://registry.lab.example.com/rhel8/
httpd-24
...output omitted...
{
    "Config": {
        "User": "1001",
        "ExposedPorts": {
            "8080/tcp": {},
            "8443/tcp": {}
        },
        "Env": [
            "PATH=/opt/app-root/src/bin:/opt/app-root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "container=oci",
            "SUMMARY=Platform for running Apache httpd 2.4 or building httpd-based application",
            "DESCRIPTION=Apache httpd 2.4 available as container, is a powerful, efficient, and extensible web server. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Virtual hosting allows one Apache installation to serve many different Web sites.",
            "STI_SCRIPTS_URL=image:///usr/libexec/s2i",
            "STI_SCRIPTS_PATH=/usr/libexec/s2i",
            "APP_ROOT=/opt/app-root",
            "HOME=/opt/app-root/src",
            "PLATFORM=el8",
            "HTTPD_VERSION=2.4",
            "HTTPD_CONTAINER_SCRIPTS_PATH=/usr/share/container-scripts/httpd/",
            "HTTPD_APP_ROOT=/opt/app-root",
            "HTTPD_CONFIGURATION_PATH=/opt/app-root/etc/httpd.d",
            "HTTPD_MAIN_CONF_PATH=/etc/httpd/conf",
            "HTTPD_MAIN_CONF_MODULES_D_PATH=/etc/httpd/conf.modules.d",
            "HTTPD_MAIN_CONF_D_PATH=/etc/httpd/conf.d",
        ]
    }
}
```

```
"HTTPD_TLS_CERT_PATH=/etc/httpd/tls",
"HTTPD_VAR_RUN=/var/run/httpd",
"HTTPD_DATA_PATH=/var/www",
"HTTPD_DATA_ORIG_PATH=/var/www",
"HTTPD_LOG_PATH=/var/log/httpd"
],
"Entrypoint": [
    "container-entrypoint"
],
"Cmd": [
    "/usr/bin/run-httpd"
],
"WorkingDir": "/opt/app-root/src",
...output omitted...
```

- 6. Führen Sie den Befehl **podman pull** aus, um ein Container-Image aus der Registry abzurufen.

```
[student@servera ~]$ podman pull registry.lab.example.com/rhel8/httpd-24
Trying to pull registry.lab.example.com/rhel8/httpd-24...
Getting image source signatures
Copying blob 77c58f19bd6e done
Copying blob 47db82df7f3f done
Copying blob 9d20433efafa0c done
Copying blob 71391dc11a78 done
Copying config 7e93f25a94 done
Writing manifest to image destination
Storing signatures
7e93f25a946892c9c175b74a0915c96469e3b4845a6da9f214fd3ec19c3d7070
```

- 7. Verwenden Sie den Befehl **podman images**, um lokal gespeicherte Images anzuzeigen.

```
[student@servera ~]$ podman images
REPOSITORY                      TAG      IMAGE ID      CREATED        SIZE
registry.lab.example.com/rhel8/httpd-24  latest   7e93f25a9468  4 weeks ago  430 MB
```

- 8. Verwenden Sie den Befehl **podman inspect**, um Informationen zu einem lokal gespeicherten Image anzuzeigen.

```
[student@servera ~]$ podman inspect registry.lab.example.com/rhel8/httpd-24
...output omitted...
{
    "Config": {
        "User": "1001",
        "ExposedPorts": {
            "8080/tcp": {},
            "8443/tcp": {}
        },
        "Env": [
            "PATH=/opt/app-root/src/bin:/opt/app-root/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",
            "container=oci",
            "OCI_RUNTIME=/opt/app-root/bin/docker-run"
        ],
        "Labels": {
            "com.redhat.component": "HTTPD"
        }
    }
}
```

```
"SUMMARY=Platform for running Apache httpd 2.4 or building httpd-based application",
 "DESCRIPTION=Apache httpd 2.4 available as container, is a powerful, efficient, and extensible web server. Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Virtual hosting allows one Apache installation to serve many different Web sites.",
 "STI_SCRIPTS_URL=image:///usr/libexec/s2i",
 "STI_SCRIPTS_PATH=/usr/libexec/s2i",
 "APP_ROOT=/opt/app-root",
 "HOME=/opt/app-root/src",
 "PLATFORM=e18",
 "HTTPD_VERSION=2.4",
 "HTTPD_CONTAINER_SCRIPTS_PATH=/usr/share/container-scripts/
httpd/",
 "HTTPD_APP_ROOT=/opt/app-root",
 "HTTPD_CONFIGURATION_PATH=/opt/app-root/etc/httpd.d",
 "HTTPD_MAIN_CONF_PATH=/etc/httpd/conf",
 "HTTPD_MAIN_CONF_MODULES_D_PATH=/etc/httpd/conf.modules.d",
 "HTTPD_MAIN_CONF_D_PATH=/etc/httpd/conf.d",
 "HTTPD_TLS_CERT_PATH=/etc/httpd/tls",
 "HTTPD_VAR_RUN=/var/run/httpd",
 "HTTPD_DATA_PATH=/var/www",
 "HTTPD_DATA_ORIG_PATH=/var/www",
 "HTTPD_LOG_PATH=/var/log/httpd"
],
"Entrypoint": [
    "container-entrypoint"
],
"Cmd": [
    "/usr/bin/run-httpd"
],
"WorkingDir": "/opt/app-root/src",
...output omitted...
```

- 9. Führen Sie den Befehl **podman rmi** aus, um ein lokal gespeichertes Image zu entfernen.

```
[student@servera ~]$ podman rmi registry.lab.example.com/rhel8/httpd-24
Untagged: registry.lab.example.com/rhel8/httpd-24:latest
Deleted: 7e93...7070
```

- 10. Führen Sie den Befehl **podman images** aus, um zu verifizieren, dass das lokal gespeicherte Image entfernt wird.

```
[student@servera ~]$ podman images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
```

► 11. Beenden Sie **servera**.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-managing finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab containers-managing finish
```

Hiermit ist die angeleitete Übung beendet.

Durchführen des erweiterten Container-Managements

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten Sie Container mit erweiterten Optionen ausführen, die auf dem System ausgeführten Container auflisten und Container starten, anhalten und beenden können.

Verwalten von Containern mit Podman

Sie können Podman verwenden, um Container mit erweiterten Konfigurationsoptionen auszuführen und laufende oder angehaltene Container zu verwalten. In diesem Abschnitt erfahren Sie, wie Sie Podman verwenden, um Container während ihres gesamten Lebenszyklus zu verwalten.

Konfigurieren von Containern

In einer anderen Übung haben Sie den Befehl **podman run** verwendet, um Container aus Container-Images zu starten. Wenn Sie einen Container ausführen, wird ein Prozess im neuen Container gestartet. Der Prozess kann eine Anwendung sein, z. B. ein Web- oder Datenbankserver. Diese Anwendung muss möglicherweise mit anderen Systemen über das Netzwerk kommunizieren und muss daher möglicherweise konfiguriert werden.

Um Netzwerkzugriff auf den Container bereitzustellen, müssen Clients eine Verbindung zu Ports auf dem Container-Host herstellen, die den Netzwerkdatenverkehr an Ports im Container übergeben. Um den Container zu konfigurieren, können Sie dem Container oft einige Umgebungsvariablen mit benutzerdefinierten Einstellungen übergeben, anstatt das Container-Image zu ändern.

Zuordnen von Container-Hostports zum Container

Wenn Sie einen Netzwerkport auf dem Container-Host einem Port im Container zuordnen, wird der an den Host-Netzwerkport gesendete Netzwerkdatenverkehr vom Container empfangen.

Sie können beispielsweise Port 8000 auf dem Container-Host Port 8080 auf dem Container zuordnen. Der Container kann einen **httpd**-Prozess ausführen, der Port 8080 überwacht. Daher wird der an den Container-Hostport 8000 gesendete Datenverkehr vom im Container ausgeführten Webserver empfangen.

Richten Sie eine Portzuordnung mit **podman run** ein. Verwenden Sie dazu die Option **-p**. Dies erfordert zwei durch Doppelpunkte getrennte Portnummern, den Port auf dem Container-Host, gefolgt vom Port im Container.

Im folgenden Beispiel wird die Option **-d** verwendet, um den Container im getrennten Modus (als Daemon) auszuführen. Wenn Sie die Option **-d** verwenden, gibt **podman** nur die Container-ID auf dem Bildschirm zurück. Mit der Option **-p 8000:8080** wird Port 8000 auf dem Container-Host Port 8080 im Container zugeordnet. Das Container-Image **registry.redhat.io/rhel8/httpd-24** führt einen Apache HTTP Server aus, der Verbindungen auf Port 8080 überwacht.

```
[user@host ~]$ podman run -d -p 8000:8080 registry.redhat.io/rhel8/httpd-24  
4a24ee199b909cc7900f2cd73c07e6fce9bd3f53b14e6757e91368c561a8edf4  
[user@host ~]$
```

Sie können den Befehl **podman port** mit einer Container-ID oder einem Namen verwenden, um die zugehörigen Portzuordnungen aufzulisten, oder mit der Option **-a**, um alle verwendeten Portzuordnungen aufzulisten. Im folgenden Beispiel werden alle auf dem Container-Host definierten Portzuordnungen aufgeführt. Die Ausgabe zeigt, dass Port 8000 auf dem Container-Host Port 8080/tcp auf dem Container zugeordnet ist, der über die mit **4a24ee199b90** beginnende ID verfügt.

```
[user@host ~]$ podman port -a  
4a24ee199b90      8080/tcp  -> 0.0.0.0:8000
```

Außerdem müssen Sie sicherstellen, dass die Firewall auf dem Container-Host externen Clients die Verbindung mit dem zugeordneten Port ermöglicht. Im vorherigen Beispiel müssen Sie möglicherweise auch Port 8000/tcp zu Ihren aktuellen Firewall-Regeln auf dem Container-Host hinzufügen:

```
[root@host ~]# firewall-cmd --add-port=8000/tcp  
success
```



Wichtig

Ein Container ohne Root kann einen Port auf dem Container-Host unter Port 1024 (einem „privilegierten Port“) nicht öffnen. Entsprechend funktioniert **-p 80:8080** normalerweise nicht bei einem Container, der durch einen Benutzer ohne **Root** ausgeführt wird. Dies ist eine Einschränkung für Benutzer auf einem Linux-System, die keine **Root**-Berechtigung besitzen. Um einen Port auf dem Container-Host unter 1024 einem Container-Port zuzuordnen, müssen Sie **podman** als **root** ausführen oder andere Anpassungen am System vornehmen.

Sie können einen Port über 1024 auf dem Container-Host einem privilegierten Port im Container zuordnen, selbst wenn Sie einen Container ohne Root ausführen. Die Zuordnung **-p 8080:80** funktioniert, wenn der Container einen Service bereitstellt, der Port 80 überwacht.

Übergeben von Umgebungsvariablen zur Konfiguration eines Containers

Die Konfiguration eines Containers kann komplex sein, da Sie zum Konfigurieren in der Regel das Container-Image nicht ändern möchten. Sie können jedoch Umgebungsvariablen an den Container übergeben, und der Container kann die Werte dieser Umgebungsvariablen verwenden, um seine Anwendung zu konfigurieren.

Wenn Sie Informationen zu den verfügbaren Variablen und deren Einsatzzweck erhalten möchten, führen Sie den Befehl **podman inspect** aus, um das Container-Image zu überprüfen. Im Folgenden sehen Sie beispielsweise ein Container-Image aus einer der Red Hat-Registries:

```
[user@host ~]$ podman inspect registry.redhat.io/rhel8/mariadb-103:1-102  
[  
{
```

```
...output omitted...
  "Labels": {
    ...output omitted...
      "name": "rhel8/mariadb-103",
      "release": "102",
      "summary": "MariaDB 10.3 SQL database server",
      "url": "https://access.redhat.com/containers/#/registry.access.redhat.com/rhel8/mariadb-103/images/1-102",
      "usage": "podman run -d -e MYSQL_USER=user -e MYSQL_PASSWORD=pass -e MYSQL_DATABASE=db -p 3306:3306 rhel8/mariadb-103",
      "vcs-ref": "ab3c3f15b6180b967a312c93e82743e842a4ac7c",
      "vcs-type": "git",
      "vendor": "Red Hat, Inc.",
      "version": "1"
    },
  ...output omitted...
```

Die Bezeichnung **url** verweist auf eine Webseite im Red Hat Container Catalog, in der Umgebungsvariablen und weitere Informationen zur Verwendung des Container-Images dokumentiert sind. Die Bezeichnung **usage** ist ein Beispiel eines typischen **podman**-Befehls zum Ausführen des Images.

Die über die Bezeichnung **url** für dieses Image bereitgestellte Seite zeigt, dass der Container Port 3306 für den Datenbankserver verwendet und dass die folgenden Umgebungsvariablen für die Konfiguration des Datenbankservices verfügbar sind:

MYSQL_USER

Benutzername für das zu erstellende MySQL-Konto

MYSQL_PASSWORD

Passwort für das Benutzerkonto

MYSQL_DATABASE

Datenbankname

MYSQL_ROOT_PASSWORD

Passwort für den root-Benutzer (optional)

Führen Sie den Befehl **podman run** mit der Option **-e** aus, um Umgebungsvariablen an einen Prozess im Container weiterzugeben. Im folgenden Beispiel werden Konfigurationseinstellungen über Umgebungs- und Portoptionen auf den Container angewendet.

```
[user@host ~]$ podman run -d --name container_name -e MYSQL_USER=user_name
-e MYSQL_PASSWORD=user_password -e MYSQL_DATABASE=database_name
-e MYSQL_ROOT_PASSWORD=mysql_root_password -p 3306:3306
registry.redhat.io/rhel8/mariadb-103:1-102
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
```

Mit der Option **--name** wird dem Container ein Name Ihrer Wahl zugewiesen, sodass ein bestimmter Container leicht zu identifizieren ist. Wenn Sie Ihrem Container keinen Namen zuweisen, weist **Podman** einen zufällig ausgewählten Namen zu.

Verwalten von Containern

Das Erstellen und Starten eines Containers ist nur der erste Schritt im Lebenszyklus des Containers. Zu diesem Lebenszyklus gehören auch das Anhalten, Neustarten oder Entfernen des

Kapitel 13 | Ausführen von Containern

Containers. Benutzer können auch den Container-Status und die Metadaten auf Fehlerbehebung, Aktualisierung oder Berichterstellung prüfen.

Mit dem Befehl **podman ps** werden die ausgeführten Container aufgelistet:

[user@host ~]\$ podman ps			COMMAND
CONTAINER ID	IMAGE		
89dd9b6354ba①	registry.redhat.io/rhel8/mariadb-103:1-102②		run-mysqld③
CREATED	STATUS	PORTS	NAMES
10 minutes ago④	Up 10 seconds⑤	0.0.0.0:3306->3306/tcp⑥	my-database⑦

- ① Jedem Container wird bei der Erstellung eine eindeutige hexadezimale Container-ID zugewiesen. Die Container-ID steht in keinem Zusammenhang mit der Image-ID.
- ② Container-Image, das zum Starten des Containers verwendet wurde.
- ③ Der Befehl, der beim Starten des Containers ausgeführt wurde.
- ④ Datum und Zeitpunkt, an dem der Container gestartet wurde.
- ⑤ Gesamtbetriebszeit des Containers (wenn dieser noch ausgeführt wird) oder Zeitdauer nach dem Beenden.
- ⑥ Ports, die über den Container oder eine Portweiterleitung bereitgestellt wurden, sofern konfiguriert.
- ⑦ Der Container-Name.

Von Podman werden angehaltene Container standardmäßig nicht sofort verworfen. Podman behält die lokalen Dateisysteme und andere Statuswerte bei, um sie bei der Post-mortem-Analyse zu nutzen, sofern Sie den Container nicht neu starten. Wenn Sie einen Container über den Befehl **podman run** mit der Option **--rm** starten, wird der Container automatisch entfernt, wenn er beendet wird.

Mit dem Befehl **podman ps -a** werden alle, also auch die angehaltenen Container aufgelistet:

[user@host ~]\$ podman ps -a			COMMAND
CONTAINER ID	IMAGE		
30b743973e98	registry.redhat.io/rhel8/httpd-24:1-105		/bin/bash
CREATED	STATUS	PORTS	NAMES
17 minutes ago	Exited (0) 18 minutes ago	80/tcp	my-httdp



Anmerkung

Beim Erstellen eines Containers wird **podman** abgebrochen, sofern der Container-Name bereits verwendet wird, selbst wenn sich der Container in einem *angehaltenen* Status befindet. Dieser Schutz verhindert doppelte Container-Namen.

Mit dem Befehl **podman stop** wird ein ausgeführter Container kontrolliert beendet. Der Befehl **stop** sendet ein SIGTERM-Signal, um einen ausgeführten Container zu beenden. Wenn der Container nach einem Kullanzzeitraum (standardmäßig 10 Sekunden) nicht aufhört, sendet Podman ein SIGKILL-Signal.

```
[user@host ~]$ podman stop my-httdp-container
77d4b7b8ed1fd57449163bcb0b78d205e70d2314273263ab941c0c371ad56412
```



Wichtig

Wenn ein Container-Image von einem angehaltenen Container verwendet wird, kann das Image nur dann mit **podman rmi** oder **podman image rm** gelöscht werden, wenn Sie die Option **-f** angeben, mit der alle Container zuerst mit dem Image entfernt werden.

Mit dem Befehl **podman rm** wird ein Container aus dem Host entfernt. Der Container muss angehalten werden, es sei denn, Sie geben die Option **-f** an, mit der auch ausgeführte Container entfernt werden. Mit dem Befehl **podman rm -a** werden alle angehaltenen Container aus dem Host entfernt. Die Container-IDs aller Container, die entfernt werden, werden ausgegeben.

```
[user@host ~]$ podman rm my-database  
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
```

Mit dem Befehl **podman restart** wird ein angehaltener Container neu gestartet. Mit dem Befehl wird ein neuer Container mit derselben Container-ID erstellt, wobei Status und Dateisystem des angehaltenen Containers wiederverwendet werden.

```
[user@host ~]$ podman restart my-httdp-container  
77d4b7b8ed1fd57449163bcb0b78d205e70d2314273263ab941c0c371ad56412
```

Mit dem Befehl **podman kill** werden UNIX-Signale an den Hauptprozess im Container gesendet. Dies sind dieselben Signale, die vom Befehl **kill** verwendet werden.

Dies kann hilfreich sein, wenn der Hauptprozess im Container Aktionen beim Empfang bestimmter Signale ausführen kann oder wenn Fehler behoben werden sollen. Wenn kein Signal angegeben ist, sendet **podman kill** das Signal SIGKILL, wodurch der Hauptprozess und der Container beendet werden.

```
[user@host ~]$ podman kill my-httdp-container  
77d4b7b8ed1fd57449163bcb0b78d205e70d2314273263ab941c0c371ad56412
```

Sie geben das Signal mit der Option **-s** an:

```
[user@host ~]$ podman kill -s SIGKILL my-httdp-container  
77d4b7b8ed1fd57449163bcb0b78d205e70d2314273263ab941c0c371ad56412
```

Jedes UNIX-Signal kann an den Hauptprozess gesendet werden. Der Befehl **podman kill** akzeptiert entweder den Namen oder die Nummer des Signals.



Anmerkung

Der Befehl **podman stop** versucht, den Befehl **stop** für das Container-Image auszuführen. Schlägt der Befehl fehl, werden jedoch die Signale **SIGTERM** und **SIGKILL** an den Container gesendet.

Ausführen von Befehlen in einem Container

Wenn ein Container gestartet wird, führt er den Einstiegspunktbefehl des Container-Images aus. Es kann jedoch erforderlich sein, zur Verwaltung des ausgeführten Containers andere

Kapitel 13 | Ausführen von Containern

Befehle auszuführen. Beispielsweise möchten Sie möglicherweise eine interaktive Shell an einen ausgeführten Container anhängen, um ihn zu überprüfen oder zu debuggen.

Mit dem Befehl **podman exec** wird ein weiterer Prozess in einem bereits ausgeführten Container gestartet:

```
[user@host ~]$ podman exec 7ed6e671a600 cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)
[user@host ~]$
```

Im vorherigen Beispiel wird zum Ausführen des Befehls die Container-ID verwendet. Es ist häufig einfacher, stattdessen den Container-Namen zu verwenden. Wenn Sie eine interaktive Shell anhängen möchten, müssen Sie die Optionen **-i** und **-t** angeben, um eine interaktive Sitzung zu öffnen und ein Pseudo-Terminal für die Shell zuzuteilen.

```
[user@host ~]$ podman exec -it my_webserver /bin/bash
bash-4.4$ hostname
7ed6e671a600
bash-4.4$ exit
[user@host ~]$
```

Podman merkt sich den letzten Container, der in einem Befehl verwendet wird. Mithilfe der Option **-1** können Sie die frühere Container-ID oder den Namen im letzten Podman-Befehl ersetzen.

```
[user@host ~]$ podman exec -1 cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)
[user@host ~]$
```



Literaturhinweise

Man Pages **podman-run(1)**, **podman-exec(1)**, **podman-ps(1)**, **podman-stop(1)**, **podman-restart(1)**, **podman-kill(1)**, **podman-rm(1)**, **podman-rmi(1)**, **podman-images(1)** und **podman-port(1)**

Weitere Informationen finden Sie im Kapitel *Working With Containers* im Handbuch *Red Hat Enterprise Linux 8 Building, Running, and Managing Containers* unter https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/building_running_and_managing_containers/index#working-with-containers_building-running-and-managing-containers

► Angeleitete Übung

Durchführen des erweiterten Container-Managements

In dieser Übung verwenden Sie Podman, um auf Ihrem Server ausgeführte Container zu verwalten.

Ergebnisse

Sie sollten in der Lage sein, Container zu erstellen und zu verwalten.

Bevor Sie Beginnen

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-advanced start** aus. Dieser Befehl führt ein Startskript aus, um zu bestimmen, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem werden der MariaDB-Client auf **servera** installiert, die Container-Registry überprüft und konfiguriert und sichergestellt, dass die für diese Übung verwendete Container-Images dort gespeichert werden.

```
[student@workstation ~]$ lab containers-advanced start
```

Anweisungen

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Melden Sie sich über den Befehl **podman login** bei der Container-Registry an.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 3. Erstellen Sie durch Ausführen des Befehls **podman run** einen getrennten MariaDB-Datenbank-Container, der auf der in den folgenden Teilschritten angegebenen Spezifikation basiert. Bestätigen Sie, dass der Container ausgeführt wird und die korrekten Ports veröffentlicht.

**Anmerkung**

Wenn Sie einen Container über den Befehl **podman run** starten und noch nicht das angegebene Container-Image aus **registry.lab.example.com** abgerufen haben, wird das angeforderte Image über den Befehl automatisch aus der Registry abgerufen.

- 3.1. Erstellen Sie einen getrennten Container namens **mydb**, der das Container-Image **registry.lab.example.com/rhel8/mariadb-103:1-102** verwendet. Der Befehl muss Port 3306 im Container auf derselben Portnummer auf dem Host veröffentlichen. Sie müssen auch die folgenden Variablenwerte deklarieren, um den Container mit dem Datenbankbenutzer **user1** (Passwort **redhat**) zu konfigurieren, das Passwort **root** auf **redhat** festzulegen und um die Datenbank **items** zu erstellen.

Variable	Wert
MYSQL_USER	user1
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	items
MYSQL_ROOT_PASSWORD	redhat

Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden. Der Einfachheit halber können Sie den folgenden Befehl aus der Datei **/tmp/containers-advanced/create-mydb.txt** kopieren und einfügen.

```
[student@servera ~]$ podman run -d --name mydb -e MYSQL_USER=user1 -e
  MYSQL_PASSWORD=redhat -e MYSQL_DATABASE=items -e MYSQL_ROOT_PASSWORD=redhat -p
  3306:3306 registry.lab.example.com/rhel8/mariadb-103:1-102
Trying to pull registry.lab.example.com/rhel8/mariadb-103:1-102...
Getting image source signatures
Copying blob 71391dc11a78 done
Copying blob 77c58f19bd6e done
Copying blob 67b9f0b530d9 done
Copying blob 47db82df7f3f done
Copying config 11a47e0fbe done
Writing manifest to image destination
Storing signatures
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
[student@servera ~]$
```

- 3.2. Mit dem Befehl **podman ps** können Sie bestätigen, dass der Container ausgeführt wird und die korrekten Ports veröffentlicht.

```
[student@servera ~]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND
abcb42ef2ff1  registry.lab.example.com/rhel8/mariadb-103:1-102  run-mysqld
CREATED       STATUS                                     PORTS
bout a minute ago Up About a minute ago  0.0.0.0:3306->3306/tcp  mydb
NAMES
```

- 4. Stellen Sie mithilfe des Befehls **mysql** in Ihrem Container **mydb** eine Verbindung mit der MariaDB-Datenbank her. Bestätigen Sie, dass die Datenbank **items** vorhanden ist. Beenden Sie dann MariaDB, und halten Sie den Container **mydb** an.
- 4.1. Stellen Sie eine Verbindung mit MariaDB her. Verwenden Sie dazu den Benutzer **user1** und das Passwort **redhat**. Geben Sie Port 3306 und die IP-Adresse von **localhost** an. Hierbei handelt es sich um Ihren Container-Host (**127.0.0.1**).

```
[student@servera ~]$ mysql -u user1 -p --port=3306 --host=127.0.0.1
Enter password: redhat
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 10.3.17-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

- 4.2. Bestätigen Sie, dass die Datenbank **items** vorhanden ist. Beenden Sie dann MariaDB.

```
MariaDB [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| items          |
| test           |
+-----+
3 rows in set (0.001 sec)

MariaDB [(none)]> exit
Bye
[student@servera ~]$
```

- 4.3. Halten Sie den Container **mydb** an. Die Container-ID unterscheidet sich von der folgenden:

```
[student@servera ~]$ podman stop mydb
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
```

- 5. Erstellen Sie einen Container, auf dem Apache HTTP Server ausgeführt wird, wodurch auch eine interaktive Bash-Shell im Container gestartet wird. Führen Sie einen Befehl über die Shell des Containers aus. Beenden Sie dann den Container und verifizieren Sie, dass der Container nicht mehr ausgeführt wird.
- 5.1. Erstellen Sie einen Container mit dem Namen **myweb**, auf dem Apache HTTP Server 2.4 ausgeführt wird, wodurch auch eine interaktive Shell im Container gestartet wird. Verwenden Sie das Container-Image **registry.lab.example.com/rhel8/httpd-24:1-105**. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run --name myweb -it registry.lab.example.com/rhel8/
httpd-24:1-105 /bin/bash
...output omitted...
bash-4.4$
```

- 5.2. Führen Sie an der interaktiven Shell-Eingabeaufforderung im Container den Befehl **cat /etc/redhat-release** aus, um den Inhalt der Datei **/etc/redhat-release** im Container anzuzeigen. Beenden Sie den Container.

```
bash-4.4$ cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)
bash-4.4$ exit
exit
[student@servera ~]$
```

- 5.3. Verifizieren Sie, dass der Container **myweb** nicht mehr ausgeführt wird.

CONTAINER ID	IMAGE	COMMAND
6d95bd8559de	registry.lab.example.com/rhel8/httpd-24:1-105	/bin/bash
abcb42ef2ff1	registry.lab.example.com/rhel8/mariadb-103:1-102	run-mysqld

CREATED	STATUS	PORTS	NAMES
About a minute ago	Exited (0) 25 seconds ago		myweb
9 minutes ago	Exited (0) 3 minutes ago	0.0.0.0:3306->3306/tcp	mydb

- 6. Erstellen Sie einen getrennten HTTPD-Webserver-Container mit dem Namen **mysecondweb**. Stellen Sie mithilfe des Namens eine Verbindung zum Container her, und zeigen Sie dann den Namen und die Version des Kernels an. Stellen Sie ein zweites Mal eine Verbindung zum Container her. Verwenden Sie jedoch die Option **(-1)**, um die ID des Containers aus dem vorherigen Befehl abzurufen und die durchschnittliche Systemauslastung anzuzeigen. Lassen Sie den Container weiterhin ausgeführt.

- 6.1. Erstellen Sie einen getrennten Container mit dem Namen **mysecondweb**. Ihre Ausgabe weicht möglicherweise von der Ausgabe dieses Übungsbeispiels ab.

```
[student@servera ~]$ podman run --name mysecondweb -d registry.lab.example.com/
rhel8/httpd-24:1-105
9e8f14e74fd4d82d95a765b8aaaeb1e93b9fe63c54c2cc805509017315460028
```

- 6.2. Stellen Sie eine Verbindung zum Container **mysecondweb** her, um den Linux-Namen und die Kernel-Version mit dem Befehl **podman exec** anzuzeigen.

```
[student@servera ~]$ podman exec mysecondweb uname -sr
Linux 4.18.0-193.el8.x86_64
```

- 6.3. Führen Sie den Befehl **podman exec** erneut aus. Verwenden Sie dieses Mal die Option **(-1)**, um die Container-ID aus dem vorherigen Befehl zu verwenden, um die durchschnittliche Systemauslastung anzuzeigen.

```
[student@servera ~]$ podman exec -l uptime
00:14:53 up 2:15, 0 users, load average: 0.08, 0.02, 0.01
```

6.4. Lassen Sie den Container **mysecondweb** weiterhin ausgeführt.

- 7. Erstellen Sie einen Container mit dem Namen **myquickweb**, der den Inhalt der Datei **/etc/redhat-release** auffüllt und dann automatisch beendet und den Container löscht. Bestätigen Sie, dass der Container gelöscht wurde.
- 7.1. Erstellen Sie den Container.
Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run --name myquickweb --rm registry.lab.example.com/rhel8/httpd-24:1-105 cat /etc/redhat-release
Red Hat Enterprise Linux release 8.2 (Ootpa)
```

- 7.2. Verwenden Sie den Befehl **podman ps -a**, um zu bestätigen, dass der Container **myquickweb** gelöscht wurde. Der Container **myquickweb** sollte in der Ausgabe des Befehls **podman ps -a** nicht aufgeführt werden.

```
[student@servera ~]$ podman ps -a | grep myquickweb
[student@servera ~]$
```

- 8. Führen Sie mit dem Befehl **podman** Massenvorgänge für vorhandene Container aus:

- Listen Sie alle Container auf, die ausgeführt werden oder angehalten wurden.
- Stellen Sie sicher, dass alle vorhandenen Container angehalten werden.
- Entfernen Sie alle Container.
- Verifizieren Sie, dass alle Container entfernt werden.

- 8.1. Listen Sie alle Container auf, die ausgeführt werden oder angehalten wurden. Ihre Ausgabe kann abweichen.

```
[student@servera ~]$ podman ps -a
CONTAINER ID  IMAGE                                     COMMAND
9e8f14e74fd4  registry.lab.example.com/rhel8/httpd-24:1-105   /usr/bin/run-http
6d95bd8559de  registry.lab.example.com/rhel8/httpd-24:1-105   /bin/bash
abcb42ef2ff1  registry.lab.example.com/rhel8/mariadb-103:1-102  run-mysqld

CREATED      STATUS          PORTS          NAMES
5 minutes ago Up 5 minutes ago                   mysecondweb
18 minutes ago Exited (0) 17 minutes ago       myweb
26 minutes ago Exited (0) 19 minutes ago       0.0.0.0:3306->3306/tcp mydb
```

- 8.2. Halten Sie alle Container an.

```
[student@servera ~]$ podman stop -a
6d95bd8559de81486b0876663e72260a8108d83aef5c5d660cb8f133f439c025
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
9e8f14e74fd4d82d95a765b8aaeb1e93b9fe63c54c2cc805509017315460028
```

8.3. Entfernen Sie alle Container.

```
[student@servera ~]$ podman rm -a  
6d95bd8559de81486b0876663e72260a8108d83aef5c5d660cb8f133f439c025  
9e8f14e74fd4d82d95a765b8aaaeb1e93b9fe63c54c2cc805509017315460028  
abcb42ef2ff1b85a50e3cd9bc15877ef823979c8166d0076ce5ebc5ea19c0815
```

8.4. Verifizieren Sie, dass alle Container entfernt wurden.

```
[student@servera ~]$ podman ps -a  
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES  
[student@servera ~]$
```

► 9. Beenden Sie **servera**.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-advanced finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab containers-advanced finish
```

Hiermit ist die angeleitete Übung beendet.

Zuordnen von persistentem Storage zu einem Container

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten Sie in der Lage sein, durch das Mounten eines Verzeichnisses vom Container-Host innerhalb eines ausgeführten Containers persistenten Storage von Container-Daten bereitzustellen.

Vorbereiten von persistenten Speicherorten

Storage im Container ist *kurzlebig*, d. h., der Inhalt geht nach dem Entfernen des Containers verloren.

Wenn die vom Container verwendeten Daten beim Neustart des Containers beibehalten werden müssen, ist der kurzlebige Storage nicht ausreichend. Beispielsweise ist Ihr Container möglicherweise ein Datenbankserver, und Sie müssen die Datenbank selbst beibehalten, wenn der Container neu gestartet wird. Um Container-Anwendungen mit dieser Anforderung zu unterstützen, müssen Sie den Container mit *persistentem Storage* bereitstellen.

Bereitstellen von persistentem Storage über den Container-Host

Eine einfache Möglichkeit, einen Container mit persistentem Storage bereitzustellen, besteht darin, ein Verzeichnis auf dem Container-Host zum Speichern der Daten zu verwenden. Podman kann ein Hostverzeichnis in einem laufenden Container mounten. Die Container-Anwendung sieht diese Hostverzeichnisse als Teil des Container-Storages an, so wie ein Remote-Netzwerkvolume von Anwendungen als Teil des Hostdateisystems angesehen wird. Wenn Sie den Container entfernen, werden die Inhalte des Verzeichnisses des Container-Hosts vom System nicht zurückgefordert. Ein neuer Container kann ihn mounten, um auf die Daten zuzugreifen.

Beispielsweise kann ein Datenbank-Container ein Hostverzeichnis verwenden, um Datenbankdateien zu speichern. Wenn dieser Datenbank-Container ausfällt, können Sie einen neuen Container unter Verwendung desselben Hostverzeichnisses erstellen. Die Datenbankdaten werden dabei für Client-Anwendungen beibehalten. Es spielt keine Rolle für den Datenbank-Container, wo Sie dieses Hostverzeichnis speichern. Es kann sich überall befinden, etwa in einer lokalen Festplattenpartition oder in einem Remote-Netzwerkdateisystem.

Vorbereiten des Hostverzeichnisses

Wenn Sie ein Host Verzeichnis vorbereiten, müssen Sie es so konfigurieren, dass die Prozesse im Container darauf zugreifen können. Die Verzeichniskonfiguration umfasst Folgendes:

- Konfigurieren des Eigentümerschaft und der Berechtigungen des Verzeichnisses
- Festlegen des entsprechenden SELinux-Kontexts

Das Benutzerkonto, das die Anwendung im Container verwendet, muss Zugriff auf das Hostverzeichnis haben. Stellen Sie sicher, dass die richtigen Berechtigungen für das Hostverzeichnis festgelegt werden, damit die Anwendung darauf zugreifen kann.

Kapitel 13 | Ausführen von Containern

Zudem müssen Sie das Hostverzeichnis mit dem entsprechenden SELinux-Kontexttyp **container_file_t** konfigurieren. Podman verwendet den SELinux-Kontexttyp **container_file_t**, um zu steuern, auf welche Dateien des Hostsystems der Container zugreifen darf. Wenn auf der Container-Schicht ein Sicherheitsfehler vorliegt, verhindert dieser zusätzliche Schutz, dass die im Container ausgeführte Anwendung auf außerhalb des freigegebenen Verzeichnisses befindliche Hostdateien zugreift. Dieser Schutz ist besonders wichtig für Anwendungen, die als **root**-Benutzer in einem root-Container ausgeführt werden.

Ohne diesen zusätzlichen Schutz von SELinux könnten diese Anwendungen **root**-Zugriff auf alle Dateien auf dem Hostsystem haben und in der Lage sein, den Host und die anderen Container zu kompromittieren. Podman kann den SELinux-Kontext des Hostverzeichnisses für Sie festlegen, wenn Sie den Container starten.

Bereitstellen von Volumes

Nach der Erstellung und Konfiguration des Hostverzeichnisses muss dieses Verzeichnis in einem Container bereitgestellt werden. Wenn Sie ein Hostverzeichnis auf einem Container mounten möchten, fügen Sie dem Befehl **podman run** die Option **--volume** (oder **-v**) hinzu. Geben Sie dabei den Hostverzeichnispfad und den Container-Storage-Pfad getrennt durch einen Doppelpunkt an:

```
--volume host_dir:container_dir:z
```

Mit der Option **Z** wendet Podman automatisch den SELinux-Kontexttyp **container_file_t** auf das Hostverzeichnis an.

Wenn Sie beispielsweise das Hostverzeichnis **/home/user/dbfiles** für MariaDB-Datenbankdateien wie **/var/lib/mysql** im Container verwenden möchten, sollten Sie den folgenden Befehl ausführen. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[user@host ~]$ podman run -d --name mydb -v /home/user/dbfiles:/var/lib/mysql:z  
-e MYSQL_USER=user -e MYSQL_PASSWORD=redhat -e MYSQL_DATABASE=inventory  
registry.redhat.io/rhel8/mariadb-103:1-102
```



Literaturhinweise

Man Page **podman-run(1)**

Dealing with user namespaces and SELinux on rootless containers (Umgang mit Benutzer-Namespace und SELinux in Containern ohne Root)

<https://www.redhat.com/sysadmin/user-namespaces-selinux-rootless-containers>

► Angeleitete Übung

Zuordnen von persistentem Storage zu einem Container

In dieser Übung erstellen Sie einen Container, der auf Webinhalte im persistenten Storage zugreift, der vom Container-Host bereitgestellt wird.

Ergebnisse

Sie sollten in der Lage sein, einen Container mit persistentem Storage bereitzustellen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-storage start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem werden die Container-Tools auf **servera** installiert.

```
[student@workstation ~]$ lab containers-storage start
```

Anweisungen

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Erstellen Sie das Verzeichnis **/home/student/webcontent/html/** und dann die Testseite **index.html**. Sie werden dieses Verzeichnis als persistenten Storage verwenden, wenn Sie einen Webserver-Container bereitstellen.
- 2.1. Erstellen Sie das Verzeichnis **~/webcontent/html/**.

```
[student@servera ~]$ mkdir -p ~/webcontent/html/  
[student@servera ~]$
```

- 2.2. Erstellen Sie die Datei **index.html**. Fügen Sie dann einige Inhalte hinzu.

```
[student@servera ~]$ echo "Hello World" > ~/webcontent/html/index.html  
[student@servera ~]$
```

- 2.3. Bestätigen Sie, dass jeder Zugriff auf das Verzeichnis und die Datei **index.html** hat. Der Container verwendet einen unprivilegierten Benutzer, der in der Lage sein muss, die Datei **index.html** zu lesen.

```
[student@servera ~]$ ls -ld webcontent/html/
drwxrwxr-x. 2 student student 24 Aug 28 04:56 webcontent/html/
[student@servera ~]$ ls -l webcontent/html/index.html
-rw-rw-r--. 1 student student 12 Aug 28 04:56 webcontent/html/index.html
```

- 3. Erstellen Sie eine Apache HTTP Server-Container-Instanz mit persistentem Storage.

- 3.1. Melden Sie sich als Benutzer **admin** mit dem Passwort **redhat321** bei der Registry **registry.lab.example.com** an.

```
[student@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 3.2. Erstellen Sie eine getrennte Container-Instanz mit dem Namen **myweb**. Leiten Sie Port 8080 auf dem lokalen Host an den Container-Port 8080 um. Mounten Sie das Verzeichnis **~/webcontent** vom Host in das Verzeichnis **/var/www** im Container. Fügen Sie das Suffix **:Z** zur Volume-Mount-Option hinzu, um den Befehl **podman** anzugeben, das Verzeichnis und seinen Inhalt umzubenennen. Verwenden Sie das Image **registry.lab.example.com/rhel8/httpd-24:1-98**. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run -d --name myweb -p 8080:8080 -v ~/webcontent:/var/www:Z registry.lab.example.com/rhel8/httpd-24:1-98
...output omitted...
```

- 3.3. Führen Sie den Befehl **podman ps** aus, um zu bestätigen, dass der Container ausgeführt wird, und verwenden Sie anschließend den Befehl **curl**, um auf den Webinhalt auf Port 8080 zuzugreifen.

```
[student@servera ~]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND
CREATED      STATUS          PORTS          NAMES
2f4844b376b7  registry.lab.example.com/rhel8/httpd-24:1-98   /usr/bin/run-http...
About a minute ago  Up About a minute ago  0.0.0.0:8080->8080/tcp  myweb
[student@servera ~]$ curl http://localhost:8080/
Hello World
```

- 4. Im vorherigen Schritt haben Sie das Tag **1-98** verwendet, um eine bestimmte Version des Images **httpd-24** auszuwählen. Es gibt eine aktuellere Version dieses Image in der Kursraum-Registry. Verwenden Sie den Befehl **skopeo inspect**, um die Details des Images **registry.lab.example.com/rhel8/httpd-24** abzurufen und um das Tag für diese Version abzurufen. Der folgende **skopeo inspect**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ skopeo inspect docker://registry.lab.example.com/rhel8/
httpd-24
{
  "Name": "registry.lab.example.com/rhel8/httpd-24",
  "Digest": "sha256:bafa...a12a",
  "RepoTags": [
    "1-98",
    "1-104",
    "1-105",
    "latest"
  ],
  ...output omitted...
}
```

Beachten Sie im Abschnitt **RepoTags**, dass eine aktuellere Version mit dem Tag **1-105** vorhanden ist.

- 5. Halten Sie den Container **myweb** an, und löschen Sie ihn. Starten Sie dann einen neuen Container mit dem Image-Tag **1-105**. Bestätigen Sie, dass sich der Webinhalt im persistenten Storage nicht geändert hat.

- 5.1. Halten Sie den Container an, und löschen Sie ihn.

```
[student@servera user]$ podman stop myweb
2f4844b376b78f8f7021fe3a4c077ae52fdc1caa6d877e84106ab783d78e1e1a
[student@servera user]$ podman rm myweb
2f4844b376b78f8f7021fe3a4c077ae52fdc1caa6d877e84106ab783d78e1e1a
```

- 5.2. Führen Sie den Befehl **podman run** erneut aus, den Sie in einem vorherigen Schritt verwendet haben, um den Container **myweb** zu starten. Ersetzen Sie jedoch das Image-Tag **1-98** durch **1-105**. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[student@servera ~]$ podman run -d --name myweb -p 8080:8080 -v ~/webcontent:/var/www:Z registry.lab.example.com/rhel8/httpd-24:1-105
...output omitted...
```

- 5.3. Führen Sie den Befehl **podman ps** aus, um zu verifizieren, dass der Container ausgeführt wird. Verwenden Sie den Befehl **curl**, um zu bestätigen, dass Ihre persistenten Volume-Daten beibehalten werden, auch wenn Sie einen neuen Container gestartet haben.

```
[student@servera ~]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND
CREATED      STATUS          PORTS          NAMES
a648c286c653  registry.lab.example.com/rhel8/httpd-24:1-105  /usr/bin/run-http...
About a minute ago  Up About a minute ago  0.0.0.0:8080->8080/tcp  myweb
[student@servera ~]$ curl http://localhost:8080/
Hello World
```

- 5.4. Beenden Sie **servera**.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-storage finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab containers-storage finish
```

Hiermit ist die angeleitete Übung beendet.

Verwalten von Containern als Services

Zielsetzungen

Nach Abschluss dieses Abschnitts sollten den Status eines Containers als **systemd**-Service starten, anhalten und überprüfen können.

Automatisches Starten von Containern mit dem Server

Wenn Sie Services wie Datenbanken oder Webserver als Container bereitstellen, möchten Sie in der Regel, dass diese Container automatisch mit dem Server gestartet werden.

Durch das Erstellen von **systemd**-Unit-Benutzerdateien für Ihre Container ohne Root können Sie sie mit **systemctl**-Befehlen verwalten, ähnlich wie bei regulären Services. Wenn Sie diese Services aktivieren, stellen Sie sicher, dass die zugeordneten Container gestartet werden, wenn der Hostrechner gestartet wird. Wenn Ihr Container im Modus „Ohne Root“ ausgeführt wird, können Sie diese Services über ein unprivilegiertes Benutzerkonto verwalten, um die Sicherheit zu erhöhen.

Für eine anspruchsvollere Skalierung und Orchestrierung vieler containerbasierter Anwendungen und Services können Sie eine auf Kubernetes basierende Enterprise-Orchestrierungsplattform wie Red Hat OpenShift Container Platform verwenden.

Ausführen von **systemd**-Services als regulärer Benutzer

Zusätzlich zur Verwaltung von Systemservices kann **systemd** auch Benutzerservices verwalten. Mit **systemd**-Benutzerservices können Benutzer Unit-Dateien für Ihre eigenen Services erstellen und diese Services mit **systemctl**-Befehlen verwalten, ohne **root**-Zugriff zu benötigen.

Wenn Sie einen Benutzerservice als Benutzer ohne Root aktivieren, wird dieser Service automatisch gestartet, wenn Sie Ihre erste Sitzung über die Text- oder grafischen Konsolen oder über SSH öffnen. Der Service wird angehalten, wenn Sie die letzte Sitzung schließen. Dieses Verhalten unterscheidet sich von den Systemservices, die gestartet werden, wenn das System gestartet und angehalten wird, wenn das System heruntergefahren wird.

Sie können dieses Standardverhalten jedoch ändern und erzwingen, dass Ihre aktivierten Services mit dem Server gestartet und während des Herunterfahrens angehalten werden, indem Sie den Befehl **logindctl enable-linger** ausführen. Um den Vorgang rückgängig zu machen, verwenden Sie den Befehl **logindctl disable-linger**. Um den aktuellen Status anzuzeigen, verwenden Sie den Befehl **logindctl show-user username** mit Ihrem Benutzernamen als Parameter.

```
[user@host ~]$ loginctl enable-linger
[user@host ~]$ loginctl show-user user
...output omitted...
Linger=yes
[user@host ~]$ loginctl disable-linger
[user@host ~]$ loginctl show-user user
...output omitted...
Linger=no
```

Erstellen und Verwalten von systemd-Benutzerservices

Erstellen Sie zum Definieren von **systemd**-Benutzerservices das Verzeichnis `~/.config/systemd/user/`, um die Unit-Dateien zu speichern. Die Syntax dieser Dateien entspricht den Unit-Systemdateien. Weitere Informationen finden Sie in den Man Pages **systemd.unit(5)** und **systemd.service(5)**.

Verwenden Sie den Befehl **systemctl** mit der Option `--user`, um die neuen Benutzerservices zu steuern. Das folgende Beispiel listet die Unit-Dateien im Verzeichnis `~/.config/systemd/user/` auf, zwingt **systemd**, die Konfiguration neu zu laden, und aktiviert und startet dann den Benutzerservice `myapp`.

```
[user@host ~]$ ls ~/.config/systemd/user/
myapp.service
[user@host ~]$ systemctl --user daemon-reload
[user@host ~]$ systemctl --user enable myapp.service
[user@host ~]$ systemctl --user start myapp.service
```



Anmerkung

Zum Verwenden von **systemctl --user**-Befehlen müssen Sie sich an der Konsole oder direkt über SSH anmelden. Die Verwendung der Befehle **sudo** oder **su** funktioniert nicht.

Der Befehl **systemctl** interagiert mit einem **systemd --user**-Prozess pro Benutzer. Das System startet diesen Prozess nur dann, wenn sich der Benutzer zum ersten Mal über die Konsole oder SSH anmeldet.

In der folgenden Tabelle werden die Unterschiede zwischen **systemd**-System- und Benutzerservices zusammengefasst.

Vergleichen von System- und Benutzerservices

Speichern von benutzerdefinierten Unit-Dateien	Systemservices	<code>/etc/systemd/system/unit.service</code>
	Benutzerservices	<code>~/.config/systemd/user/unit.service</code>
Erneutes Laden von Unit-Dateien	Systemservices	<code># systemctl daemon-reload</code>
	Benutzerservices	<code>\$ systemctl --user daemon-reload</code>

Starten und Anhalten eines Services	Systemservices	<code># systemctl start <i>UNIT</i></code> <code># systemctl stop <i>UNIT</i></code>
	Benutzerservices	<code>\$ systemctl --user start <i>UNIT</i></code> <code>\$ systemctl --user stop <i>UNIT</i></code>
Starten eines Services beim Starten des Rechners	Systemservices	<code># systemctl enable <i>UNIT</i></code>
	Benutzerservices	<code>\$ logindctl enable-linger</code> <code>\$ systemctl --user enable <i>UNIT</i></code>

Verwalten von Containern mit Systemd-Services

Wenn ein einzelner Container-Host eine kleine Anzahl von Containern ausführt, können Sie benutzerbasierte **systemd**-Unit-Dateien einrichten und konfigurieren, um die Container automatisch mit dem Server zu starten. Dies ist ein einfacher Ansatz, der bei sehr einfachen und kleinen Bereitstellungen, die nicht skaliert werden müssen, besonders nützlich ist. Für praktischere Produktionsinstallationen sollten Sie Red Hat OpenShift Container Platform verwenden, die am Ende dieses Abschnitts kurz besprochen wird.

Erstellen eines dedizierten Benutzerkontos zum Ausführen von Containern

Zum Vereinfachen der Verwaltung der Container ohne Root können Sie ein dediziertes Benutzerkonto erstellen, das Sie für alle Ihre Container verwenden. Auf diese Weise können Sie sie über ein einzelnes Benutzerkonto verwalten.



Anmerkung

Das Konto, das Sie für die Gruppierung aller Container erstellen, muss ein reguläres Benutzerkonto sein. Wenn Sie ein Konto mit **useradd** erstellen, reserviert der Befehl einen Bereich der Benutzer-IDs für die Container des Benutzers in der Datei **/etc/subuid**. Wenn Sie jedoch ein Systemkonto mit der **--system**-Option (oder **-r**) von **useradd**, wird vom Befehl kein Bereich reserviert. Folglich können Sie keine Container ohne Root mit Systemkonten starten.

Erstellen der systemd-Unit-Datei

Über einen vorhandenen Container kann mit dem Befehl **podman** die **systemd**-Unit-Datei für Sie erstellt werden. Im folgenden Beispiel wird der Befehl **podman generate systemd** verwendet, um die Unit-Datei für den vorhandenen **Web**-Container zu erstellen:

```
[user@host ~]$ cd ~/.config/systemd/user/
[user@host user]$ podman generate systemd --name web --files --new
/home/user/.config/systemd/user/container-web.service
```

Der Befehl **podman generate systemd** verwendet einen Container als Modell zum Erstellen der Konfigurationsdatei. Nachdem die Datei erstellt wurde, müssen Sie den Container löschen, da **systemd** erwartet, dass der Container zunächst nicht vorhanden ist.

Der Befehl **podman generate systemd** akzeptiert die folgenden Optionen:

--name container_name

Die Option **--name** gibt den Namen eines vorhandenen Containers an, der als Modell verwendet werden soll, um die Unit-Datei zu generieren. Podman verwendet auch diesen Namen, um den Namen der Unit-Datei zu erstellen: **container-container_name.service**.

--files

Die Option **--files** weist Podman an, die Unit-Datei im aktuellen Verzeichnis zu generieren. Ohne die Option zeigt Podman die Datei in der Standardausgabe an.

--new

Die Option **--new** weist Podman an, den **systemd**-Service so zu konfigurieren, dass der Container erstellt wird, wenn der Service gestartet wird, und löscht ihn, wenn der Service angehalten wird. In diesem Modus ist der Container kurzlebig. In der Regel benötigen Sie persistenten Storage, um die Daten beizubehalten. Ohne die Option **--new** konfiguriert Podman den Service so, dass er den vorhandenen Container startet und anhält, ohne ihn zu löschen.

Das folgende Beispiel zeigt die Anweisungen „start“ und „stop“ in der Unit-Datei, wenn Sie den Befehl **podman generate systemd** mit der Option **--new** ausführen:

```
[user@host ~]$ podman run -d --name web -v /home/user/www:/var/www:Z
registry.redhat.io/rhel8/httpd-24:1-105
[user@host ~]$ podman generate systemd --name web --new
...output omitted...
ExecStart=/usr/bin/podman run --common-pidfile %t/%n-pid --cidfile %t/%n-cid --cgroups=no-common -d --name web -v /home/user/webcontent:/var/www:Z ①
registry.redhat.io/rhel8/httpd-24:1-105
ExecStop=/usr/bin/podman stop --ignore --cidfile %t/%n-cid -t 10 ②
ExecStopPost=/usr/bin/podman rm --ignore -f --cidfile %t/%n-cid ③
...output omitted...
```

- ① Beim Start führt **systemd** den Befehl **podman run** aus, um einen neuen Container zu erstellen und zu starten.
- ② Beim Anhalten führt **systemd** den Befehl **podman stop** aus, um den Container anzuhalten.
- ③ Nachdem **systemd** den Container angehalten hat, wird er von **systemd** mit dem Befehl **podman rm** entfernt.

Dagegen sehen Sie im folgenden Beispiel die Anweisungen „start“ und „stop“, wenn Sie den Befehl **podman generate systemd** ohne die Option **--new** ausführen:

```
[user@host ~]$ podman run -d --name web -v /home/user/www:/var/www:Z
registry.redhat.io/rhel8/httpd-24:1-105
[user@host ~]$ podman generate systemd --name web
...output omitted...
ExecStart=/usr/bin/podman start web ①
ExecStop=/usr/bin/podman stop -t 10 web ②
...output omitted...
```

- ① Beim Start führt **systemd** den Befehl **podman start** aus, um einen vorhandenen Container zu starten.
- ② Beim Anhalten führt **systemd** den Befehl **podman stop** aus, um den Container anzuhalten. Beachten Sie, dass **systemd** den Container nicht löscht.

Starten und Anhalten von Containern mit `systemd`

Verwenden Sie den Befehl `systemctl`, um Ihre Container zu steuern.

- Starten des Containers:

```
[user@host ~]$ systemctl --user start container-web
```

- Anhalten des Containers:

```
[user@host ~]$ systemctl --user stop container-web
```

- Abrufen des Status des Containers:

```
[user@host ~]$ systemctl --user status container-web
```



Wichtig

Container, die mit dem Befehl `systemctl` verwaltet werden, werden von `systemd` gesteuert. `systemd` überwacht den Container-Status und startet sie neu, wenn Sie fehlschlagen.

Verwenden Sie nicht den Befehl `podman`, um diese Container zu starten oder anzuhalten. Dadurch kann die `systemd`-Überwachung beeinträchtigt werden.

Konfigurieren von Containern für den Start beim Start des Hostrechners

Standardmäßig werden aktivierte `systemd`-Benutzerservices gestartet, wenn ein Benutzer die erste Sitzung öffnet, und angehalten, wenn der Benutzer die letzte Sitzung schließt. Führen Sie den Befehl `logind`enable-linger` aus, um die Benutzerservices automatisch mit dem Server zu starten:

```
[user@host ~]$ logind`enable-linger
```

Verwenden Sie den Befehl `systemctl`, um zu aktivieren, dass ein Container gestartet wird, wenn der Hostrechner gestartet wird:

```
[user@host ~]$ systemctl --user enable container-web
```

Wenn Sie deaktivieren möchten, dass ein Container beim Starten des Hostrechners gestartet wird, verwenden Sie den Befehl `systemctl` mit der Option `disable`:

```
[user@host ~]$ systemctl --user disable container-web
```

Verwalten von Containern, die als root mit `systemd` ausgeführt werden

Sie können auch Container konfigurieren, die als root ausgeführt werden sollen, um mit `systemd`-Unit-Dateien verwaltet zu werden. Ein Vorteil dieses Ansatzes besteht darin, dass Sie diese Unit-

Kapitel 13 | Ausführen von Containern

Dateien so konfigurieren können, dass sie genau wie normale Unit-Systemdateien funktionieren, und nicht als ein bestimmter Benutzer.

Die Vorgehensweise für die Einrichtung ist vergleichbar mit der, die zuvor für Container ohne Root beschrieben wurde, mit Ausnahme von:

- Es ist nicht erforderlich, einen dedizierten Benutzer einzurichten.
- Wenn Sie die Unit-Datei mit **podman generate systemd** erstellen, führen Sie sie im Verzeichnis **/etc/systemd/system** anstatt im Verzeichnis **~/.config/systemd/user** aus.
- Wenn Sie den Service des Containers mit **systemctl** konfigurieren, verwenden Sie nicht die Option **--user**.
- Sie müssen **logindctl enable-linger** nicht als **root** ausführen.

Eine Demonstration finden Sie im YouTube-Video des Red Hat-Videokanals, der in den Verweisen am Ende dieses Abschnitts aufgeführt ist.

Angemessene Orchestrierung von Containern

In diesem Kapitel haben Sie erfahren, wie Container manuell über die Befehlszeile auf einem einzelnen Host konfiguriert und verwaltet werden und wie systemd konfiguriert wird, sodass Container automatisch mit dem Server gestartet werden. Dies ist in einem sehr kleinen Umfang nützlich, um mehr über Container zu erfahren.

In der Tat benötigen die meisten Unternehmensbereitstellungen jedoch mehr. In der Einführung in dieses Kapitel wurde erwähnt, dass Kubernetes in der Regel zur Verwaltung komplexer Anwendungen verwendet wird, die aus mehreren kooperierenden Containern bestehen. Red Hat OpenShift ist eine Kubernetes-Plattform. Auf der zugehörigen webbasierten Benutzeroberfläche können Sie u. a. Container in einem Cluster von Container-Hosts überwachen und ausführen sowie automatische Skalierungen, Protokollierungen und Auditing vornehmen.

Die Erörterung dieser Tools geht über den Umfang dieses Kurses hinaus. Wenn Sie mehr erfahren möchten, bietet Red Hat Training weitere Kurse an, beginnend mit dem kostenlosen technischen Übersichtskurs *Deploying Containerized Applications* (DO080) und *Red Hat OpenShift I: Containers & Kubernetes* (DO180). Weitere Informationen finden Sie unter <https://www.redhat.com/training>.

Weitere Informationen zu Kubernetes und Red Hat OpenShift finden Sie unter <https://www.openshift.com>. Dort stehen einige Ressourcen zur Verfügung. So können Sie etwa OpenShift mit Tools wie CodeReady Containers [<https://developers.redhat.com/products/codeready-containers/overview>] testen. Weitere Informationen finden Sie in <https://www.openshift.com/try>.



Literaturhinweise

Man Pages **logindctl(1)**, **systemd.unit(5)**, **systemd.service(5)**, **subuid(5)** und **podman-generate-systemd(1)**

Improved systemd integration with Podman 2.0 (Verbesserte systemd-Integration in Podman 2.0)

<https://www.redhat.com/sysadmin/improved-systemd-podman>

Managing Containers in Podman with Systemd Unit Files (Verwalten von Containern in Podman mit Systemd-Unit-Dateien)

<https://www.youtube.com/watch?v=AGkM2jGT61Y>

What is OpenShift (Was ist OpenShift?)

<https://www.openshift.com/learn/what-is-openshift>

Get Started with OpenShift (Erste Schritte in OpenShift)

<https://www.openshift.com/try>

Weitere Informationen finden Sie im Kapitel *Running Containers as Systemd Services with Podman* im Handbuch *Red Hat Enterprise Linux 8 Building, Running, and Managing Containers* unter

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/building_running_and_managing_containers/index#using-systemd-with-containers_building-running-and-managing-containers

► Angeleitete Übung

Verwalten von Containern als Services

In dieser Übung konfigurieren Sie einen Container, der als **systemd**-Service verwaltet wird. Dann verwenden Sie **systemctl**-Befehle, um diesen Container zu verwalten, sodass er automatisch gestartet wird, wenn der Hostrechner gestartet wird.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von **systemd**-Unit-Dateien zum Verwalten von Containern
- Starten und Anhalten von Containern mit **systemctl**-Befehlen
- Konfigurieren von Benutzerkonten für **systemd**-Benutzerservices, sodass diese beim Starten des Hostrechners gestartet werden

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-services start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **servera** im Netzwerk erreichbar ist. Außerdem werden die Container-Tools auf **servera** installiert.

```
[student@workstation ~]$ lab containers-services start
```

Anweisungen

- 1. Melden Sie sich mit dem Befehl **ssh** bei **servera** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **student** lautet **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 3. Erstellen Sie ein Benutzerkonto mit dem Namen **contsvc** mit dem Passwort **redhat**. Konfigurieren Sie das Konto für den Zugriff auf die Container-Image-Registry unter

registry.lab.example.com. Anstelle Ihres regulären Benutzerkontos verwenden Sie dieses Konto, um Container als **systemd**-Services auszuführen.

- 3.1. Verwenden Sie den Befehl **useradd**, um das Konto zu erstellen. Verwenden Sie dann den Befehl **passwd**, um das Passwort auf **redhat** festzulegen.

```
[root@servera ~]# useradd contsvc
[root@servera ~]# 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.
```

- 3.2. Sie müssen sich direkt als der Benutzer **contsvc** anmelden, um die **systemd**-Benutzerservices mit dem Konto **contsvc** verwalten zu können. Sie können die Befehle **su** und **sudo** nicht verwenden.
Melden Sie sich von **servera** ab. Melden Sie sich anschließend mit dem Befehl **ssh** als Benutzer **contsvc** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 3.3. Erstellen Sie das Verzeichnis **~/.config/containers/**.

```
[contsvc@servera ~]$ mkdir -p ~/.config/containers/
[contsvc@servera ~]$
```

- 3.4. Vom Skript **lab** wurde die Datei **registries.conf** im Verzeichnis **/tmp/containers-services/** vorbereitet. Kopieren Sie die Datei nach **~/.config/containers/**. Der folgende **cp**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[contsvc@servera ~]$ cp /tmp/containers-services/registries.conf ~/.config/containers/
```

- 3.5. Führen Sie den Befehl **podman search ubi** testweise aus, um zu bestätigen, dass Sie auf die Registry **registry.lab.example.com** zugreifen können. Wenn alles erwartungsgemäß funktioniert, sollte der Befehl einige Images auflisten.

```
[contsvc@servera ~]$ podman search ubi
INDEX          NAME              DESCRIPTION      STARS      OFFICIAL      AUTOMATED
example.com    registry.lab.example.com/ubi8/ubi
example.com    registry.lab.example.com/ubi7/ubi      0
```

Kapitel 13 | Ausführen von Containern

- 4. Erstellen Sie das Verzeichnis **/home/contsvc/webcontent/html/** und dann die Testseite **index.html**. Sie werden das Verzeichnis als persistenten Storage verwenden, wenn Sie einen Webserver-Container bereitstellen.

- 4.1. Erstellen Sie das Verzeichnis **~/webcontent/html/**.

```
[contsvc@servera ~]$ mkdir -p ~/webcontent/html/
[contsvc@servera ~]$
```

- 4.2. Erstellen Sie die Datei **index.html**. Fügen Sie dann einige Inhalte hinzu.

```
[contsvc@servera ~]$ echo "Hello World" > ~/webcontent/html/index.html
[contsvc@servera ~]$
```

- 4.3. Bestätigen Sie, dass jeder Zugriff auf das Verzeichnis und die Datei **index.html** hat. Der Container verwendet einen unprivilegierten Benutzer, der in der Lage sein muss, die Datei **index.html** zu lesen.

```
[contsvc@servera ~]$ ls -ld webcontent/html/
drwxrwxr-x. 2 contsvc contsvc 24 Aug 28 04:56 webcontent/html/
[contsvc@servera ~]$ ls -l webcontent/html/index.html
-rw-rw-r--. 1 contsvc contsvc 12 Aug 28 04:56 webcontent/html/index.html
```

- 5. Erstellen Sie einen getrennten Container mit dem Namen **myweb**. Leiten Sie Port 8080 auf dem lokalen Host an den Container-Port 8080 um. Mounten Sie das Verzeichnis **~/webcontent** vom Host in das Verzeichnis **/var/www** im Container. Verwenden Sie das Image **registry.lab.example.com/rhel8/httpd-24:1-105**.

- 5.1. Melden Sie sich als Benutzer **admin** mit dem Passwort **redhat321** bei der Registry **registry.lab.example.com** an.

```
[contsvc@servera ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 5.2. Erstellen Sie den Container. Sie können den folgenden Befehl aus der Datei **/tmp/containers-services/start-container.txt** kopieren und einfügen. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[contsvc@servera ~]$ podman run -d --name myweb -p 8080:8080 -v ~/webcontent:/var/www:Z registry.lab.example.com/rhel8/httpd-24:1-105
...output omitted...
```

- 5.3. Verwenden Sie zum Verifizieren Ihrer Arbeit den Befehl **curl**, um auf den Webinhalt auf Port 8080 zuzugreifen.

```
[contsvc@servera ~]$ curl http://localhost:8080/
Hello World
```

- 6. Erstellen Sie die **systemd**-Unit-Datei zum Verwalten des Containers **myweb** mit **systemctl**-Befehlen. Halten Sie den Container **myweb** an, und löschen Sie ihn, sobald Sie fertig sind. Systemd verwaltet den Container und erwartet nicht, dass der Container anfänglich vorhanden ist.

- 6.1. Erstellen Sie das Verzeichnis `~/.config/systemd/user/`.

```
[contsvc@servera ~]$ mkdir -p ~/.config/systemd/user/  
[contsvc@servera ~]$
```

- 6.2. Wechseln Sie in das Verzeichnis `~/.config/systemd/user/`. Führen Sie dann den Befehl **podman generate systemd** aus, um die Unit-Datei für den Container **myweb** zu erstellen. Verwenden Sie die Option `--new`, sodass **systemd** beim Starten des Services einen neuen Container erstellt und den Container beim Beenden des Services löscht.

```
[contsvc@servera ~]$ cd ~/.config/systemd/user  
[contsvc@servera user]$ podman generate systemd --name myweb --files --new  
/home/contsvc/.config/systemd/user/container-myweb.service
```

- 6.3. Halten Sie den Container **myweb** an, und löschen Sie ihn.

```
[contsvc@servera user]$ podman stop myweb  
2f4844b376b78f8f7021fe3a4c077ae52fdc1caa6d877e84106ab783d78e1e1a  
[contsvc@servera user]$ podman rm myweb  
2f4844b376b78f8f7021fe3a4c077ae52fdc1caa6d877e84106ab783d78e1e1a
```

- 7. Zwingen Sie **systemd**, seine Konfiguration neu zu laden. Aktivieren und starten Sie anschließend Ihren neuen Benutzerservice **container-myweb**. Wenn Sie Ihre Arbeit testen möchten, halten Sie den Service an und starten Sie ihn dann und kontrollieren Sie den Container-Status mit den Befehlen **curl** und **podman ps**.

- 7.1. Verwenden Sie den Befehl **systemctl --user daemon-reload** für **systemd**, um die neue Unit-Datei zu berücksichtigen.

```
[contsvc@servera user]$ systemctl --user daemon-reload  
[contsvc@servera user]$
```

- 7.2. Aktivieren und starten Sie den Service **container-myweb**.

```
[contsvc@servera user]$ systemctl --user enable --now container-myweb  
Created symlink /home/contsvc/.config/systemd/user/multi-user.target.wants/  
container-myweb.service → /home/contsvc/.config/systemd/user/container-  
myweb.service.  
Created symlink /home/contsvc/.config/systemd/user/default.target.wants/container-  
myweb.service → /home/contsvc/.config/systemd/user/container-myweb.service.
```

- 7.3. Verifizieren Sie anhand der Befehle **podman ps** und **curl**, dass der Container ausgeführt wird.

```
[contsvc@servera user]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND
CREATED      STATUS          PORTS          NAMES
a648c286c653  registry.lab.example.com/rhel8/httpd-24:1-105  /usr/bin/run-http...
About a minute ago  Up About a minute ago  0.0.0.0:8080->8080/tcp  myweb
[contsvc@servera user]$ curl http://localhost:8080/
Hello World
```

Notieren Sie sich die Container-ID. Anhand dieser Informationen bestätigen Sie, dass **systemd** einen neuen Container erstellt, wenn Sie den Service neu starten.

- 7.4. Halten Sie den Service **container-myweb** an, und bestätigen Sie, dass der Container nicht mehr vorhanden ist. Wenn Sie den Service anhalten, wird **systemd** angehalten und löscht dann den Container.

```
[contsvc@servera user]$ systemctl --user stop container-myweb
[contsvc@servera user]$ podman ps --all
CONTAINER ID  IMAGE  COMMAND  CREATED  STATUS  PORTS  NAMES
```

- 7.5. Starten Sie den Service **container-myweb**. Bestätigen Sie anschließend, dass der Container ausgeführt wird.

```
[contsvc@servera user]$ systemctl --user start container-myweb
[contsvc@servera user]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND
CREATED      STATUS          PORTS          NAMES
6f5148b27726  registry.lab.example.com/rhel8/httpd-24:1-105  /usr/bin/run-http...
5 seconds ago  Up 4 seconds ago  0.0.0.0:8080->8080/tcp  myweb
```

Beachten Sie, dass sich die Container-ID geändert hat. Wenn Sie den Service starten, erstellt **systemd** einen neuen Container.

- 8. Führen Sie den Befehl **logindctl enable-linger** aus, um sicherzustellen, dass Benutzerservices für den Benutzer **contsvc** mit dem Server gestartet werden. Starten Sie **servera** neu, wenn Sie fertig sind.

- 8.1. Führen Sie den Befehl **logindctl enable-linger** aus.

```
[contsvc@servera user]$ logindctl enable-linger
[contsvc@servera user]$
```

- 8.2. Bestätigen Sie, dass die Option **Linger** für den Benutzer **contsvc** festgelegt ist.

```
[contsvc@servera user]$ logindctl show-user contsvc
...output omitted...
Linger=yes
```

- 8.3. Wechseln Sie zum Benutzer **root**, und verwenden Sie dann den Befehl **systemctl reboot**, um **servera** neu zu starten.

```
[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. Warten Sie, bis der Rechner **servera** neu gestartet wird. Dies dauert einige Minuten. Melden Sie sich dann als Benutzer **contsvc** bei **servera** an. Bestätigen Sie, dass **systemd** den Container **myweb** gestartet hat und dass der Webinhalt verfügbar ist.
- 9.1. Melden Sie sich auf **workstation** mit dem Befehl **ssh** bei **servera** als Benutzer **contsvc** an.

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

- 9.2. Bestätigen Sie anhand des Befehls **podman ps**, dass der Container ausgeführt wird.

```
[contsvc@servera ~]$ podman ps  
CONTAINER ID IMAGE COMMAND  
CREATED STATUS PORTS NAMES  
1d174e79f08b registry.lab.example.com/rhel8/httpd-24:1-105 /usr/bin/run-http...  
3 minutes ago Up 3 minutes ago 0.0.0.0:8080->8080/tcp myweb
```

- 9.3. Verwenden Sie den Befehl **curl**, um auf den Webinhalt zuzugreifen.

```
[contsvc@servera ~]$ curl http://localhost:8080/  
Hello World
```

- 9.4. Beenden Sie **servera**.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-services finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab containers-services finish
```

Hiermit ist die angeleitete Übung beendet.

► Praktische Übung

Ausführen von Containern

In dieser praktischen Übung konfigurieren Sie einen Container auf Ihrem Server, der einen MariaDB-Datenbankservice bereitstellt, seine Datenbank auf persistentem Storage speichert und automatisch mit dem Server startet.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von getrennten Containern
- Konfigurieren der Portumleitung und des persistenten Storages
- Konfigurieren von **systemd** für Container für den Start beim Start des Hostrechners

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem werden der MariaDB-Client installiert und das Benutzerkonto **podsvc** erstellt, das Sie zum Ausführen eines MariaDB-Containers verwenden.

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

Anweisungen

1. Installieren Sie die Container-Tools auf **serverb**. Melden Sie sich mit dem Befehl **sudo** bei **serverb** als Benutzer **student** an. Das Passwort für den Benutzer **student** lautet **student**.
2. Die Container-Image-Registry unter **registry.lab.example.com** speichert das Image **rhel8/mariadb-103** mit verschiedenen Tags. Listen Sie diese Tags als Benutzer **podsvc** auf **serverb** auf. Achten Sie dabei auf das Tag mit der *niedrigsten* Versionsnummer. Sie werden dieses Image-Tag verwenden, um einen Container zu einem späteren Zeitpunkt in dieser Übung zu starten.
Das Passwort für den Benutzer **podsvc** lautet **redhat**. Verwenden Sie zum Abfragen der Registry **registry.lab.example.com** das Konto **admin** mit dem Passwort **redhat321**.
3. Erstellen Sie das Verzeichnis **/home/podsvc/db_data** als Benutzer **podsvc** auf **serverb**. Bereiten Sie das Verzeichnis so vor, dass Container Lese-/Schreibzugriff haben. Sie werden dieses Verzeichnis für persistenten Storage verwenden.
4. Erstellen Sie als der Benutzer **podsvc** auf **serverb** einen getrennten MariaDB-Container mit dem Namen **inventorydb**. Verwenden Sie das Image **rhel8/mariadb-103** aus der Registry **registry.lab.example.com**. Geben Sie das Tag mit der niedrigsten Versionsnummer für dieses Image an, das Sie in einem vorherigen Schritt gefunden

haben. Ordnen Sie Port 3306 im Container Port 13306 auf dem Host zu. Mounten Sie das Verzeichnis **/home/podsvc/db_data** auf dem Host als **/var/lib/mysql/data** im Container. Deklarieren Sie die folgenden Variablenwerte:

Variable	Wert
MYSQL_USER	operator1
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

Sie können diese Parameter aus der Datei **/home/podsvc/containers-review/variables** auf **serverb** kopieren und einfügen.

Verwenden Sie den Befehl **mysql**, um zu bestätigen, dass die MariaDB-Datenbank ausgeführt wird. Dieser Befehl befindet sich im Skript **/home/podsvc/containers-review/testdb.sh**. Sie können das Skript auch direkt ausführen, um die Datenbank zu testen.

5. Konfigurieren Sie **systemd** als Benutzer **podsvc** auf **serverb** so, dass der Container **inventorydb** automatisch mit dem Server gestartet wird.

Bewertung

Verwenden Sie als Benutzer **student** auf dem Rechner **workstation** den Befehl **lab**, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-services finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

► Lösung

Ausführen von Containern

In dieser praktischen Übung konfigurieren Sie einen Container auf Ihrem Server, der einen MariaDB-Datenbankservice bereitstellt, seine Datenbank auf persistentem Storage speichert und automatisch mit dem Server startet.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von getrennten Containern
- Konfigurieren der Portumleitung und des persistenten Storages
- Konfigurieren von **systemd** für Container für den Start beim Start des Hostrechners

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab containers-review start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem werden der MariaDB-Client installiert und das Benutzerkonto **podsvc** erstellt, das Sie zum Ausführen eines MariaDB-Containers verwenden.

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

Anweisungen

1. Installieren Sie die Container-Tools auf **serverb**. Melden Sie sich mit dem Befehl **sudo** bei **serverb** als Benutzer **student** an. Das Passwort für den Benutzer **student** lautet **student**.
 - 1.1. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als Benutzer **student** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 1.2. Installieren Sie das Yum-Modul **container-tools** mit dem Befehl **yum**.

```
[student@serverb ~]$ sudo yum module install container-tools
[sudo] password for student: student
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

2. Die Container-Image-Registry unter **registry.lab.example.com** speichert das Image **rhel8/mariadb-103** mit verschiedenen Tags. Listen Sie diese Tags als Benutzer **podsvc** auf **serverb** auf. Achten Sie dabei auf das Tag mit der *niedrigsten* Versionsnummer. Sie werden dieses Image-Tag verwenden, um einen Container zu einem späteren Zeitpunkt in dieser Übung zu starten.

Das Passwort für den Benutzer **podsvc** lautet **redhat**. Verwenden Sie zum Abfragen der Registry **registry.lab.example.com** das Konto **admin** mit dem Passwort **redhat321**.

- 2.1. Beenden Sie das Konto **student** auf **serverb**.

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

- 2.2. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als Benutzer **podsvc** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 2.3. Melden Sie sich über den Befehl **podman login** bei der Container-Registry an.

```
[podsvc@serverb ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 2.4. Verwenden Sie den Befehl **skopeo inspect**, um Informationen zum Image **registry.lab.example.com/rhel8/mariadb-103** anzuzeigen. Der folgende **skopeo inspect**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[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...

```

Das Tag mit der niedrigsten Nummer lautet **1-86**.

3. Erstellen Sie das Verzeichnis **/home/podsvc/db_data** als Benutzer **podsvc** auf **serverb**. Bereiten Sie das Verzeichnis so vor, dass Container Lese-/Schreibzugriff haben. Sie werden dieses Verzeichnis für persistenten Storage verwenden.
- 3.1. Erstellen Sie das Verzeichnis **/home/podsvc/db_data**.

```
[podsvc@serverb ~]$ mkdir /home/podsvc/db_data
[podsvc@serverb ~]$
```

- 3.2. Legen Sie den Zugriffsmodus des Verzeichnisses auf 777 fest, sodass jeder über Lese-/Schreibzugriff verfügt.

```
[podsvc@serverb ~]$ chmod 777 /home/podsvc/db_data
[podsvc@serverb ~]$
```

4. Erstellen Sie als der Benutzer **podsvc** auf **serverb** einen getrennten MariaDB-Container mit dem Namen **inventorydb**. Verwenden Sie das Image **rhel8/mariadb-103** aus der Registry **registry.lab.example.com**. Geben Sie das Tag mit der niedrigsten Versionsnummer für dieses Image an, das Sie in einem vorherigen Schritt gefunden haben. Ordnen Sie Port 3306 im Container Port 13306 auf dem Host zu. Mounten Sie das Verzeichnis **/home/podsvc/db_data** auf dem Host als **/var/lib/mysql/data** im Container. Deklarieren Sie die folgenden Variablenwerte:

Variable	Wert
MYSQL_USER	operator1
MYSQL_PASSWORD	redhat
MYSQL_DATABASE	inventory
MYSQL_ROOT_PASSWORD	redhat

Sie können diese Parameter aus der Datei **/home/podsvc/containers-review/variables** auf **serverb** kopieren und einfügen.

Verwenden Sie den Befehl **mysql**, um zu bestätigen, dass die MariaDB-Datenbank ausgeführt wird. Dieser Befehl befindet sich im Skript **/home/podsvc/containers-review/testdb.sh**. Sie können das Skript auch direkt ausführen, um die Datenbank zu testen.

- 4.1. Verwenden Sie den Befehl **podman run**, um den Container zu erstellen. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[podsvc@serverb ~]$ podman run -d --name inventorydb -p 13306:3306 -v /
home/podsvc/db_data:/var/lib/mysql/data:Z -e MYSQL_USER=operator1 -e
MYSQL_PASSWORD=redhat -e MYSQL_DATABASE=inventory -e MYSQL_ROOT_PASSWORD=redhat
registry.lab.example.com/rhel8/mariadb-103:1-86
...output omitted...
```

- 4.2. Bestätigen Sie, dass die Datenbank ausgeführt wird.

```
[podsvc@serverb ~]$ ~/containers-review/testdb.sh  
Testing the access to the database...  
SUCCESS
```

5. Konfigurieren Sie **systemd** als Benutzer **podsvc** auf **serverb** so, dass der Container **inventorydb** automatisch mit dem Server gestartet wird.
- 5.1. Wenn Sie **sudo** oder **su** bei der Anmeldung als Benutzer **podsvc** verwendet haben, beenden Sie **serverb**. Verwenden Sie dann den Befehl **ssh**, um sich als Benutzer **podsvc** direkt auf **serverb** anzumelden. Beachten Sie, dass **systemd** erfordert, dass der Benutzer eine direkte Sitzung über die Konsole oder über SSH öffnen kann.

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

- 5.2. Erstellen Sie das Verzeichnis **~/.config/systemd/user/**.

```
[podsvc@serverb ~]$ mkdir -p ~/.config/systemd/user/  
[podsvc@serverb ~]$
```

- 5.3. Verwenden Sie den Befehl **podman generate systemd**, um die **systemd**-Unit-Datei aus dem ausgeführten Container zu erstellen.

```
[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. Halten Sie den Container **inventorydb** an, und löschen Sie ihn.

```
[podsvc@serverb user]$ podman stop inventorydb  
0d28f0e0a4118ff019691e34afe09b4d28ee526079b58d19f03b324bd04fd545  
[podsvc@serverb user]$ podman rm inventorydb  
0d28f0e0a4118ff019691e34afe09b4d28ee526079b58d19f03b324bd04fd545
```

- 5.5. Weisen Sie **systemd** an, seine Konfiguration neu zu laden. Aktivieren und starten Sie anschließend den Service **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. Bestätigen Sie, dass der Container ausgeführt wird.

```
[podsvc@serverb user]$ ~/containers-review/testdb.sh
Testing the access to the database...
SUCCESS
[podsvc@serverb user]$ podman ps
CONTAINER ID  IMAGE                                     COMMAND      CREATED
              STATUS          PORTS          NAMES
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. Führen Sie den Befehl **logindctl enable-linger** aus, um die Benutzerservices automatisch mit dem Server zu starten.

```
[podsvc@serverb ~]$ logindctl enable-linger
[podsvc@serverb ~]$
```

- 5.8. Beenden Sie **serverb**.

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

Bewertung

Verwenden Sie als Benutzer **student** auf dem Rechner **workstation** den Befehl **lab**, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

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

Beenden

Führen Sie auf dem Rechner **workstation** das Skript **lab containers-services finish** aus, um diese Übung zu beenden.

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

Hiermit wird die praktische Übung abgeschlossen.

Zusammenfassung

In diesem Kapitel wurden die folgenden Themen behandelt:

- Container bieten eine einfache Möglichkeit, eine Anwendung und ihre Abhängigkeiten zu verteilen und auszuführen, die in Konflikt mit der auf dem Host installierten Software stehen können.
- Container werden aus Container-Images ausgeführt, die Sie aus einer Container-Registry herunterladen oder selbst erstellen können.
- Podman wird von Red Hat Enterprise Linux bereitgestellt und verwaltet Container und Container-Images direkt auf einem einzelnen Host.
- Container können als **root** oder als unprivilegierte Container ohne Root ausgeführt werden, um die Sicherheit zu erhöhen.
- Sie können Netzwerkports auf dem Container-Host zuordnen, um den Datenverkehr an Services zu übergeben, die in den Containern ausgeführt werden. Sie können auch Umgebungsvariablen verwenden, um die Software in Containern zu konfigurieren.
- Container-Storage ist temporär. Sie können jedoch einem Container persistenten Storage zuordnen und dabei beispielsweise die Inhalte eines auf dem Container-Host befindlichen Verzeichnisses verwenden.
- Sie können Systemd so konfigurieren, dass Container beim Systemstart automatisch ausgeführt werden.

Kapitel 14

Ausführliche Wiederholung

Ziel

Wiederholen von Aufgaben aus *Red Hat System Administration II*

Zielsetzungen

- Wiederholen von Aufgaben aus *Red Hat System Administration II*

Abschnitte

- Ausführliche Wiederholung

Praktische Übung

- Praktische Übung: Beheben von Startproblemen und Warten von Servern
- Praktische Übung: Konfigurieren und Verwalten von Dateisystemen und Storage
- Praktische Übung: Konfigurieren und Verwalten der Serversicherheit
- Praktische Übung: Ausführen von Containern

Ausführliche Wiederholung

Ziele

In diesem Abschnitt werden das Wissen und die in *Red Hat System Administration II* vermittelten Kenntnisse überprüft und aufgefrischt.

Überprüfung für Red Hat System Administration II

Bevor Sie mit der ausführlichen Überprüfung für diesen Kurs beginnen, sollten Sie mit den in den jeweiligen Kapiteln behandelten Themen vertraut sein.

Für zusätzliche Übungen stehen Ihnen auch die vorherigen Kapitel dieses Lehrbuchs zur Verfügung.

Kapitel 1, Steigern der Produktivität in der Befehlszeile

Führen Sie die Befehle effizienter aus, indem Sie erweiterte Funktionen der Bash-Shell, Shell-Skripte und verschiedene von Red Hat Enterprise Linux bereitgestellte Dienstprogramme verwenden.

- Befehlsfolgen durch Schreiben eines einfachen Shell-Skripts automatisieren
- Befehle für Listen von Elementen in einem Skript oder aus der Befehlszeile mit Schleifen und Bedingungen ausführen
- Mit dem Befehl **grep** und regulären Ausdrücken in Protokolldateien und der Befehlausgabe nach Text suchen, der mit einem Muster übereinstimmt

Kapitel 2, Terminieren zukünftiger Tasks

Terminieren von Tasks zur automatischen Ausführung in der Zukunft

- Einen Befehl einrichten, der zu einem späteren Zeitpunkt einmal ausgeführt wird
- Die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei eines Benutzers terminieren
- Die Ausführung von Befehlen gemäß einem sich wiederholenden Zeitplan mit der Crontab-Datei des Systems terminieren
- systemd-Timer aktivieren und deaktivieren sowie einen Timer zum Verwalten temporärer Dateien konfigurieren

Kapitel 3, Tuning der Systemleistung

Die Systemleistung durch Festlegen von Tuning-Parametern verbessern und die Planungspriorität von Prozessen festlegen

- Die Systemleistung durch Auswahl eines vom Daemon „tuned“ verwalteten Tuning-Profiles optimieren
- Die Priorität bestimmter Prozesse mit den Befehlen „nice“ und „renice“ festlegen bzw. aufheben

Kapitel 4, Steuern des Dateizugriffs mit ACLs

Access Control Lists (ACLs, Zugriffssteuerungslisten) für Dateien interpretieren und festlegen, um komplexe Benutzer- und Gruppenzugriffsberechtigungen zu steuern

- Anwendungsfälle für ACLs beschreiben, Dateien identifizieren, für die ACLs festgelegt sind, und die Auswirkungen dieser ACLs interpretieren
- ACLs für Dateien festlegen und entfernen sowie Default-ACLs definieren, die für neu erstellte Dateien automatisch durch ein Verzeichnis festgelegt werden

Kapitel 5, Verwalten der SELinux-Sicherheit

Die Sicherheit eines Servers mit SELinux schützen und verwalten

- Beschreiben, wie SELinux Ressourcen schützt und wie der Enforcement-Modus ausgewählt wird
- Den SELinux-Kontext einer Datei konfigurieren, um zu steuern, wie Prozesse mit dieser Datei interagieren
- Boolesche SELinux-Werte konfigurieren, um Änderungen der Laufzeitrichtlinie für unterschiedliche Zugriffsanforderungen zuzulassen
- SELinux-Protokollmeldungen untersuchen und SELinux-AVC-Verweigerungen beheben

Kapitel 6, Verwalten von Basisspeicher

Speichergeräte, Partitionen, Dateisysteme und Swap-Speicher über die Befehlszeile erstellen und verwalten

- Speicherpartitionen erstellen, mit Dateisystemen formatieren und mounten
- Swap-Speicher als Ergänzung zum physischen Arbeitsspeicher erstellen und verwalten

Kapitel 7, Verwalten logischer Volumes

Logische Volumes, die Dateisysteme und Swap-Speicher enthalten, über die Befehlszeile erstellen und verwalten

- Logische Volumes auf Speichergeräten erstellen und verwalten, mit Dateisystemen formatieren oder mit Swap-Speicher vorbereiten
- Volume-Gruppen zugewiesenen Speicher hinzufügen und entfernen sowie zerstörungsfrei die Größe eines mit einem Dateisystem formatierten logischen Volumes erweitern

Kapitel 8, Implementieren erweiterter Storage-Features

Verwalten Sie den Storage mithilfe des lokalen Speicherverwaltungssystems von Stratis, und verwenden Sie die VDO-Volumes, um den verwendeten Speicherplatz zu optimieren.

- Verwalten von mehreren Storage-Ebenen mit der lokalen Speicherverwaltung von Stratis.
- Optimieren der Speicherplatznutzung mittels VDO zum Komprimieren und Deduplizieren von Daten auf Storage-Geräten.

Kapitel 9, Zugreifen auf Network-Attached Storage

Zugreifen auf Network-Attached Storage mit dem NFS-Protokoll.

Kapitel 14 | Ausführliche Wiederholung

- Einhängen, Verwenden und Aushängen eines NFS-Exports über die Befehlszeile und zur Startzeit.
- Konfiguration des Automounters mit direkten und indirekten Zuordnungen, um bei Bedarf automatisch ein NFS-Dateisystem einzubinden (mount) und es zu entfernen (umount), wenn es nicht mehr verwendet wird.

Kapitel 10, Steuern des Boot-Vorgangs

Verwalten Sie den Boot-Vorgang, um die angebotenen Services zu steuern und Probleme zu beheben.

- Beschreiben des Red Hat Enterprise Linux-Boot-Vorgangs, Festlegen des beim Booten verwendeten Standardziels und Booten eines Systems zu einem nicht standardmäßigen Ziel.
- Anmelden beim System und Ändern des Root-Passworts, wenn das aktuelle Root-Passwort verloren gegangen ist.
- Manuelles Reparieren der Dateisystemkonfiguration oder bei Beschädigungsproblemen, die den Boot-Vorgang stoppen.

Kapitel 11, Verwalten der Netzwerksicherheit

Steuern Sie mithilfe der System-Firewall und der SELinux-Regeln die Netzwerkverbindungen zu Services.

- Akzeptieren oder Ablehnen von Netzwerkverbindungen zu Systemservices mithilfe von Firewalld-Regeln.
- Steuern, ob Netzwerkservices bestimmte Netzwerkports verwenden können, indem Sie SELinux-Port-Labels verwalten.

Kapitel 12, Installation von Red Hat Enterprise Linux

Installieren Sie Red Hat Enterprise Linux auf Servern und virtuellen Rechnern.

- Installieren von Red Hat Enterprise Linux auf einem Server.
- Automatisieren des Installationsvorgangs mit Kickstart.
- Installieren eines virtuellen Rechners auf dem Red Hat Enterprise Linux-Server mit Cockpit.

Kapitel 13, Ausführen von Containern

Abrufen, Ausführen und Verwalten einfacher kompakter Services als Container auf einem einzelnen Red Hat Enterprise Linux-Server

- Erklären des Containerkonzepts und dessen Verwendung zum Verwalten und Bereitstellen von Anwendungen mit unterstützenden Softwarebibliotheken und -abhängigkeiten
- Installieren von Container-Managementtools und Ausführen eines einfachen Containers ohne Root
- Suchen, Abrufen, Überprüfen und Verwalten der von einer Remote-Container-Registry abgerufenen und auf Ihrem Server gespeicherten Container-Images
- Ausführen von Containern mit erweiterten Optionen, Auflisten der im System ausgeführten Container sowie Starten, Anhalten und Beenden von Containern

- Bereitstellen von persistentem Storage für Containerdaten durch das Mounten eines Verzeichnisses vom Containerhost in einem ausgeführten Container
- Starten, Anhalten und Überprüfen des Status eines Containers als systemd-Service

► Praktische Übung

Beheben von Startproblemen und Warten von Servern

In dieser Überprüfung beheben und reparieren Sie Probleme beim Starten und aktualisieren das Standardziel des Systems. Sie planen auch, dass Aufgaben nach einem wiederkehrenden Zeitplan als normaler Benutzer ausgeführt werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Diagnostizieren von Problemen und Wiederherstellen des Systems aus dem Notfallmodus.
- Ändern des Standardziels von **graphical.target** in **multi-user.target**.
- Planen wiederkehrender Jobs als ein normaler Benutzer.

Bevor Sie Beginnen

Kopieren Sie alle Dateien oder Arbeiten, die Sie behalten möchten, vor dem Zurücksetzen auf andere Systeme. Setzen Sie jetzt die Systeme **workstation**, **servera** und **serverb** zurück.

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview1 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview1 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** aus, um die ausführliche Überprüfung abzuschließen:

- Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break1** aus. Dieses Abbruchskript bewirkt, dass der Startvorgang auf **serverb** fehlschlägt. Es legt auch einen längeren Timeout im Menü **GRUB2** fest, um den Startvorgang zu unterbrechen und neu **serverb** zu starten.

Beheben Sie die mögliche Ursache, und reparieren Sie den Startfehler. Durch die Korrektur muss sichergestellt werden, dass **serverb** ohne Eingriff neu gebootet wird. Verwenden Sie **redhat** als Passwort des Superusers, falls erforderlich.

- Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break2** aus. Dieses Abbruchskript bewirkt, dass das Standardziel auf **serverb** vom Ziel **multi-user** in das Ziel **graphical** geändert wird. Es legt auch einen längeren Timeout für das Menü **GRUB2** fest, um den Bootvorgang zu unterbrechen, und startet **serverb** neu.

Ändern Sie auf **serverb** das Standardziel so, dass das Ziel **multi-user** verwendet wird. Die Standardzieleinstellungen müssen nach dem Neustart ohne manuelle Eingriffe bestehen bleiben.

Führen Sie den Befehl **sudo** als der Benutzer **student** mit dem Passwort **student** aus, um privilegierte Befehle auszuführen.

- Terminieren Sie als Benutzer **student** einen wiederkehrenden Job, der das Skript **/home/student/backup-home.sh** stündlich zwischen 19 Uhr und 21 Uhr an allen Tagen außer Samstag und Sonntag ausführt.

Laden Sie das Backup-Skript von <http://materials.example.com/labs/backup-home.sh> herunter. Das Backup-Skript **backup-home.sh** sichert das Verzeichnis **/home/student** von **serverb** auf **servera** im Verzeichnis **/home/student/serverb-backup**. Führen Sie das Skript **backup-home.sh** aus, um den wiederkehrenden Job als der Benutzer **student** auf **serverb** zu planen.

- Starten Sie das System neu und warten Sie auf den Abschluss des Startvorgangs, bevor Sie die Auswertung vornehmen.

Bewertung

Führen Sie auf **workstation** das Skript **lab rhcsa-compreview1 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-compreview1 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-compreview1 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-compreview1 finish
```

Speichern Sie vor der nächsten Übung die Dateien oder Arbeiten für die Verwendung auf anderen Systemen, und setzen Sie anschließend **workstation**, **servera** und **serverb** zurück.

Damit ist die ausführliche Überprüfung abgeschlossen.

► Lösung

Beheben von Startproblemen und Warten von Servern

In dieser Überprüfung beheben und reparieren Sie Probleme beim Starten und aktualisieren das Standardziel des Systems. Sie planen auch, dass Aufgaben nach einem wiederkehrenden Zeitplan als normaler Benutzer ausgeführt werden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Diagnostizieren von Problemen und Wiederherstellen des Systems aus dem Notfallmodus.
- Ändern des Standardziels von **graphical.target** in **multi-user.target**.
- Planen wiederkehrender Jobs als ein normaler Benutzer.

Bevor Sie Beginnen

Kopieren Sie alle Dateien oder Arbeiten, die Sie behalten möchten, vor dem Zurücksetzen auf andere Systeme. Setzen Sie jetzt die Systeme **workstation**, **servera** und **serverb** zurück.

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview1 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview1 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** aus, um die ausführliche Überprüfung abzuschließen:

- Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break1** aus. Dieses Abbruchskript bewirkt, dass der Startvorgang auf **serverb** fehlschlägt. Es legt auch einen längeren Timeout im Menü **GRUB2** fest, um den Startvorgang zu unterbrechen und neu **serverb** zu starten.

Beheben Sie die mögliche Ursache, und reparieren Sie den Startfehler. Durch die Korrektur muss sichergestellt werden, dass **serverb** ohne Eingriff neu gebootet wird. Verwenden Sie **redhat** als Passwort des Superusers, falls erforderlich.

- Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break2** aus. Dieses Abbruchskript bewirkt, dass das Standardziel auf **serverb** vom Ziel **multi-user** in das Ziel **graphical** geändert wird. Es legt auch einen längeren Timeout für das Menü **GRUB2** fest, um den Bootvorgang zu unterbrechen, und startet **serverb** neu.

Ändern Sie auf **serverb** das Standardziel so, dass das Ziel **multi-user** verwendet wird. Die Standardzieleinstellungen müssen nach dem Neustart ohne manuelle Eingriffe bestehen bleiben.

Führen Sie den Befehl **sudo** als der Benutzer **student** mit dem Passwort **student** aus, um privilegierte Befehle auszuführen.

- Terminieren Sie als Benutzer **student** einen wiederkehrenden Job, der das Skript **/home/student/backup-home.sh** stündlich zwischen 19 Uhr und 21 Uhr an allen Tagen außer Samstag und Sonntag ausführt.

Laden Sie das Backup-Skript von <http://materials.example.com/labs/backup-home.sh> herunter. Das Backup-Skript **backup-home.sh** sichert das Verzeichnis **/home/student** von **serverb** auf **servera** im Verzeichnis **/home/student/serverb-backup**. Führen Sie das Skript **backup-home.sh** aus, um den wiederkehrenden Job als der Benutzer **student** auf **serverb** zu planen.

- Starten Sie das System neu und warten Sie auf den Abschluss des Startvorgangs, bevor Sie die Auswertung vornehmen.

- Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break1** aus.

1.1.

```
[student@workstation ~]$ lab rhcsa-compreview1 break1
```

- Greifen Sie nach dem Start von **serverb** auf die Konsole zu und beachten Sie, dass der Startvorgang frühzeitig angehalten wurde. Nehmen Sie sich einen Augenblick Zeit, um Vermutungen zu einer möglichen Ursache für dieses Verhalten anzustellen.
 - Suchen Sie entsprechend Ihrer Kursumgebung nach dem Symbol für die **serverb**-Konsole. Öffnen Sie die Konsole.
 - Bei der Betrachtung des Fehlers erscheint es, dass zumindest noch Teile des Systems funktionieren.
 - Drücken Sie **Strg+Alt+Entf**, um **serverb** neu zu starten.
Wenn das Bootloader-Menü angezeigt wird, drücken Sie auf eine beliebige Taste ausgenommen die **Eingabetaste**, um den Zählvorgang zu unterbrechen.
 - Bearbeiten Sie den Bootloader-Standardeintrag im Arbeitsspeicher, um sich im Notfallmodus anzumelden.
Drücken Sie **e**, um den aktuellen Eintrag zu bearbeiten.
 - Navigieren Sie mithilfe der Cursor-Tasten zu der Zeile, die mit **linux** beginnt. Fügen Sie **systemd.unit=emergency.target** an das Ende der Zeile an.
 - Drücken Sie **Strg+x**, um mit der geänderten Konfiguration zu booten.
 - Melden Sie sich im Notfallmodus an. Das **root**-Passwort lautet **redhat**.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@serverb ~]#
```

- Stellen Sie das **/**-Dateisystem im Modus Lesen/Schreiben neu bereit. Führen Sie den Befehl **mount -a** aus, um zu versuchen, alle anderen Dateiesysteme bereitzustellen.

Kapitel 14 | Ausführliche Wiederholung

- 3.1. Stellen Sie das **/**-Dateisystem im Modus Lesen/Schreiben neu bereit, um das Dateisystem zu bearbeiten.

```
[root@serverb ~]# mount -o remount,rw /
```

- 3.2. Führen Sie den Befehl **mount -a** aus, um zu versuchen, alle anderen Dateisysteme bereitzustellen. Beachten Sie, dass eines der Dateisysteme nicht bereitgestellt werden kann.

```
[root@serverb ~]# mount -a
mount: /FakeMount: can't find UUID=fake.
```

- 3.3. Bearbeiten Sie **/etc/fstab**, um das Problem zu beheben. Entfernen Sie die falsche Zeile oder kommentieren Sie sie aus.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=fake      /FakeMount  xfs  defaults    0 0
```

- 3.4. Aktualisieren Sie **systemd**, damit das System die neue **/etc/fstab**-Konfiguration registriert.

```
[root@serverb ~]# systemctl daemon-reload
[ 206.828912] systemd[1]: Reloading.
```

- 3.5. Überprüfen Sie, ob **/etc/fstab** jetzt korrekt ist, indem Sie versuchen, alle Einträge zu mounten.

```
[root@serverb ~]# mount -a
```

- 3.6. Starten Sie **serverb** neu und warten Sie auf den Abschluss des Startvorgangs. Das System sollte nun ordnungsgemäß gebootet werden.

```
[root@serverb ~]# systemctl reboot
```

4. Führen Sie auf **workstation** den Befehl **lab rhcsa-compreview1 break2** aus.

4.1.

```
[student@workstation ~]$ lab rhcsa-compreview1 break2
```

Warten Sie, bis der Bootvorgang abgeschlossen ist, bevor Sie fortfahren.

5. Wechseln Sie auf **serverb** zum Ziel **multi-user**. Legen Sie das Standardziel auf **multi-user** fest. Führen Sie den Befehl **sudo** aus, um erforderliche Administrationsbefehle auszuführen, und verwenden Sie bei Aufforderung **student** als Passwort.

- 5.1. Öffnen Sie auf **workstation** als Benutzer **student** eine SSH-Sitzung zu **serverb**.

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

- 5.2. Bestimmen Sie als der Benutzer **student** auf **serverb** das Standardziel.

```
[student@serverb ~]$ systemctl get-default  
graphical.target
```

- 5.3. Wechseln Sie zum Ziel **multi-user**. Führen Sie den Befehl **sudo** aus. Verwenden Sie bei Aufforderung **student** als Passwort.

```
[student@serverb ~]$ sudo systemctl isolate multi-user.target  
[sudo] password for student: student
```

- 5.4. Legen Sie **serverb** fest, um das Ziel **multi-user** als Standardziel zu verwenden.

```
[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. Starten Sie **serverb** neu, um zu verifizieren, dass das Ziel **multi-user** als Standardziel festgelegt ist.

```
[student@serverb ~]$ sudo systemctl reboot  
Connection to serverb closed by remote host.  
Connection to serverb closed.  
[student@workstation ~]$
```

- 5.6. Öffnen Sie nach dem Neustart eine SSH-Sitzung mit **serverb** als **student**. Verifizieren Sie, dass das Ziel **multi-user** als Standardziel fungiert.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$ systemctl get-default  
multi-user.target
```

6. Terminieren Sie als Benutzer **student** einen wiederkehrenden Job, der das Skript **/home/student/backup-home.sh** stündlich zwischen 19 Uhr und 21 Uhr an allen Tagen außer Samstag und Sonntag ausführt.

Verwenden Sie das Skript **backup-home.sh**, um den wiederkehrenden Job zu planen. Laden Sie das Backup-Skript unter <http://materials.example.com/labs/backup-home.sh> herunter.

- 6.1. Laden Sie auf **serverb** das Backup-Skript von <http://materials.example.com/labs/backup-home.sh> herunter. Wandeln Sie mit **chmod** das Backup-Skript in eine ausführbare Datei um.

```
[student@serverb ~]$ wget http://materials.example.com/labs/backup-home.sh  
...output omitted...  
[student@serverb ~]$ chmod +x backup-home.sh
```

- 6.2. Führen Sie den Befehl **crontab -e** aus, um die Crontab-Datei im Standardtexteditor zu öffnen.

Kapitel 14 | Ausführliche Wiederholung

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

- 6.3. Bearbeiten Sie die Datei, um die folgende Zeile hinzuzufügen:

```
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

Speichern Sie Ihre Änderungen und beenden Sie den Texteditor.

- 6.4. Listen Sie mit dem Befehl **crontab -l** die geplanten wiederkehrenden Jobs auf.

```
[student@serverb ~]$ crontab -l  
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

7. Starten Sie **serverb** neu und warten Sie auf den Abschluss des Startvorgangs, bevor Sie die Auswertung vornehmen.

7.1.

```
[student@serverb ~]$ sudo systemctl reboot  
[sudo] password for student: student  
Connection to serverb closed by remote host.  
Connection to serverb closed.  
[student@workstation ~]$
```

Bewertung

Führen Sie auf **workstation** das Skript **lab rhcsa-compreviw1 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-compreviw1 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-compreviw1 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-compreviw1 finish
```

Speichern Sie vor der nächsten Übung die Dateien oder Arbeiten für die Verwendung auf anderen Systemen, und setzen Sie anschließend **workstation**, **servera** und **serverb** zurück.

Damit ist die ausführliche Überprüfung abgeschlossen.

► Praktische Übung

Konfigurieren und Verwalten von Dateisystemen und Storage

In diesem Test werden Sie ein logisches LVM-Volume erstellen, ein Netzwerk-Dateisystem bereitstellen, eine Swap-Partition erstellen, die beim Starten automatisch aktiviert wird, temporäre, nicht verwendete Dateien für die Bereinigung aus dem System konfigurieren und zum Schutz eines Verzeichnisses ACLs verwenden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines logischen LVM-Volumes.
- Bereitstellen eines Netzwerkdateisystems.
- Erstellen einer Swap-Partition, die beim Starten automatisch aktiviert wird.
- Konfigurieren temporär nicht verwendeter Dateien, die aus dem System entfernt werden sollen.
- Verwenden von ACLs, um ein Verzeichnis zu schützen.

Bevor Sie Beginnen

Kopieren Sie alle Dateien oder Arbeiten, die Sie behalten möchten, vor dem Zurücksetzen auf andere Systeme. Setzen Sie die Systeme **workstation**, **servera** und **serverb** jetzt zurück, sofern Sie sie am Ende der letzten Übung nicht bereits zurückgesetzt haben.

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview2 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** aus, um die ausführliche Überprüfung abzuschließen.

- Konfigurieren Sie ein neues logisches Volume mit 1 GiB namens **vol_home** in einer Volume-Gruppe mit 2 GiB namens **extra_storage**. Verwenden Sie das nicht partitionierte Laufwerk **/dev/vdb**, um Partitionen zu erstellen.
- Das logische Volume **vol_home** sollte mit dem **XFS**-Dateisystemtyp formatiert und persistent auf **/home-directories** bereitgestellt werden.
- Stellen Sie sicher, dass das Network File System **/share** auch nach erneuten Bootvorgängen persistent auf **/local-share** bereitgestellt ist. Der NFS-Server **servera.lab.example.com** exportiert das Network File System **/share**. Der NFS-Exportpfad lautet **servera.lab.example.com:/share**.

Kapitel 14 | Ausführliche Wiederholung

- Erstellen Sie eine neue Partition mit 512 MiB auf dem Laufwerk **/dev/vdc**, die als Swap-Speicher verwendet werden soll. Dieser Swap-Speicher muss beim Booten automatisch aktiviert werden.
- Erstellen Sie eine neue Gruppe mit dem Namen **production**. Erstellen Sie die Benutzer **production1**, **production2**, **production3** und **production4**. Stellen Sie sicher, dass sie die neue Gruppe namens **production** als ihre Ergänzungsgruppe verwenden.
- Konfigurieren Sie Ihr System so, dass es ein neues Verzeichnis mit dem Namen **/run/volatile** verwendet, um temporäre Dateien zu speichern. Dateien in diesem Verzeichnis sollten zeitbasiert bereinigt werden, wenn mehr als 30 Sekunden nicht auf sie zugegriffen wurde. Die oktalen Berechtigungen für das Verzeichnis müssen **0700** lauten. Konfigurieren Sie mit der Datei **/etc/tmpfiles.d/volatile.conf** die zeitbasierte Bereinigung für die Dateien in **/run/volatile**.
- Erstellen Sie ein neues Verzeichnis namens **/webcontent**. Der Besitzer und die Gruppe des Verzeichnisses sollten als **root** festgelegt sein. Die Gruppenmitglieder von **production** sollten dieses Verzeichnis lesen und in es schreiben können. Der Benutzer **production1** sollte nur in der Lage sein, dieses Verzeichnis zu lesen. Diese Berechtigungen sollten für alle Dateien und Verzeichnisse gelten, die im Verzeichnis **/webcontent** neu erstellt werden.

Bewertung

Führen Sie auf **workstation** das Skript **lab rhcsa-compreview2 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-compreview2 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-compreview2 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-compreview2 finish
```

Damit ist die ausführliche Überprüfung abgeschlossen.

► Lösung

Konfigurieren und Verwalten von Dateisystemen und Storage

In diesem Test werden Sie ein logisches LVM-Volume erstellen, ein Netzwerk-Dateisystem bereitstellen, eine Swap-Partition erstellen, die beim Starten automatisch aktiviert wird, temporäre, nicht verwendete Dateien für die Bereinigung aus dem System konfigurieren und zum Schutz eines Verzeichnisses ACLs verwenden.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen eines logischen LVM-Volumes.
- Bereitstellen eines Netzwerkdateisystems.
- Erstellen einer Swap-Partition, die beim Starten automatisch aktiviert wird.
- Konfigurieren temporär nicht verwendeter Dateien, die aus dem System entfernt werden sollen.
- Verwenden von ACLs, um ein Verzeichnis zu schützen.

Bevor Sie Beginnen

Kopieren Sie alle Dateien oder Arbeiten, die Sie behalten möchten, vor dem Zurücksetzen auf andere Systeme. Setzen Sie die Systeme **workstation**, **servera** und **serverb** jetzt zurück, sofern Sie sie am Ende der letzten Übung nicht bereits zurückgesetzt haben.

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview2 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** aus, um die ausführliche Überprüfung abzuschließen.

- Konfigurieren Sie ein neues logisches Volume mit 1 GiB namens **vol_home** in einer Volume-Gruppe mit 2 GiB namens **extra_storage**. Verwenden Sie das nicht partitionierte Laufwerk **/dev/vdb**, um Partitionen zu erstellen.
- Das logische Volume **vol_home** sollte mit dem **XFS**-Dateisystemtyp formatiert und persistent auf **/home-directories** bereitgestellt werden.
- Stellen Sie sicher, dass das Network File System **/share** auch nach erneuten Bootvorgängen persistent auf **/local-share** bereitgestellt ist. Der NFS-Server **servera.lab.example.com** exportiert das Network File System **/share**. Der NFS-Exportpfad lautet **servera.lab.example.com:/share**.

Kapitel 14 | Ausführliche Wiederholung

- Erstellen Sie eine neue Partition mit 512 MiB auf dem Laufwerk **/dev/vdc**, die als Swap-Speicher verwendet werden soll. Dieser Swap-Speicher muss beim Booten automatisch aktiviert werden.
- Erstellen Sie eine neue Gruppe mit dem Namen **production**. Erstellen Sie die Benutzer **production1**, **production2**, **production3** und **production4**. Stellen Sie sicher, dass sie die neue Gruppe namens **production** als ihre Ergänzungsgruppe verwenden.
- Konfigurieren Sie Ihr System so, dass es ein neues Verzeichnis mit dem Namen **/run/volatile** verwendet, um temporäre Dateien zu speichern. Dateien in diesem Verzeichnis sollten zeitbasiert bereinigt werden, wenn mehr als 30 Sekunden nicht auf sie zugegriffen wurde. Die oktalen Berechtigungen für das Verzeichnis müssen **0700** lauten. Konfigurieren Sie mit der Datei **/etc/tmpfiles.d/volatile.conf** die zeitbasierte Bereinigung für die Dateien in **/run/volatile**.
- Erstellen Sie ein neues Verzeichnis namens **/webcontent**. Der Besitzer und die Gruppe des Verzeichnisses sollten als **root** festgelegt sein. Die Gruppenmitglieder von **production** sollten dieses Verzeichnis lesen und in es schreiben können. Der Benutzer **production1** sollte nur in der Lage sein, dieses Verzeichnis zu lesen. Diese Berechtigungen sollten für alle Dateien und Verzeichnisse gelten, die im Verzeichnis **/webcontent** neu erstellt werden.

1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **serverb**.

1.1.

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

2. Wechseln Sie zum Benutzer **root**.

2.1.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

3. Erstellen Sie eine Partition mit 2 GiB auf **/dev/vdb**.

3.1.

```
[root@serverb ~]# parted /dev/vdb mklabel msdos  
[root@serverb ~]# parted /dev/vdb mkpart primary 1GiB 3GiB  
[root@serverb ~]# parted /dev/vdb set 1 lvm on
```

4. Erstellen Sie ein logisches Volume namens **vol_home** mit der Partition mit 2 GiB, die Sie auf **/dev/vdb** erstellt haben. Benennen Sie die Volume-Gruppe **extra_storage**.

4.1. Deklarieren Sie das Blockgerät **/dev/vdb1** als ein physisches Volume.

```
[root@serverb ~]# pvcreate /dev/vdb1  
...output omitted...
```

- 4.2. Erstellen Sie die Volume-Gruppe **extra_storage** mit **/dev/vdb1**.

```
[root@serverb ~]# vgcreate extra_storage /dev/vdb1  
...output omitted...
```

- 4.3. Erstellen Sie ein logisches Volume mit 1 GiB namens **vol_home**.

```
[root@serverb ~]# lvcreate -L 1GiB -n vol_home extra_storage  
...output omitted...
```

5. Formatieren Sie **vol_home** mit dem **XFS**-Dateisystemtyp und stellen Sie es auf **/home-directories** bereit.

- 5.1. Erstellen Sie das Verzeichnis **/home-directories**.

```
[root@serverb ~]# mkdir /home-directories
```

- 5.2. Formatieren Sie **/dev/extra_storage/vol_home** mit dem Dateisystemtyp **XFS**.

```
[root@serverb ~]# mkfs -t xfs /dev/extra_storage/vol_home  
...output omitted...
```

- 5.3. Stellen Sie **/dev/extra_storage/vol_home** persistent auf **/home-directories** bereit. Verwenden Sie die UUID der Struktur, wenn Sie den Eintrag in **/etc/fstab** erstellen.

```
[root@serverb ~]# lsblk -o UUID /dev/extra_storage/vol_home  
UUID  
988cf149-0667-4733-abca-f80c6ec50ab6  
[root@serverb ~]# echo "UUID=988cf149-0667-4733-abca-f80c6ec50ab6 /home-directories \  
xfs defaults 0 0" >> /etc/fstab  
[root@serverb ~]# mount -a
```

6. Stellen Sie sicher, dass das Network File System **/share** auch nach erneuten Bootvorgängen persistent auf **/local-share** bereitgestellt ist. Der NFS-Server **servera.lab.example.com** exportiert das Network File System **/share**. Der NFS-Exportpfad lautet **servera.lab.example.com:/share**.

- 6.1. Erstellen Sie das Verzeichnis **/local-share**.

```
[root@serverb ~]# mkdir /local-share
```

- 6.2. Hängen Sie den unter **servera.lab.example.com:/share** verfügbaren entsprechenden Eintrag an **/etc/fstab** an, sodass das Network File System auf **/local-share** auch nach erneuten Bootvorgängen persistent bereitgestellt ist.

```
[root@serverb ~]# echo "servera.lab.example.com:/share /local-share \  
nfs rw,sync 0 0" >> /etc/fstab
```

- 6.3. Stellen Sie das Network File System basierend auf dem Eintrag in **/etc/fstab** auf **/local-share** bereit.

```
[root@serverb ~]# mount /local-share
```

7. Erstellen Sie eine neue Partition mit 512 MiB auf dem Laufwerk **/dev/vdc**, die als Swap-Speicher verwendet werden soll. Dieser Swap-Speicher muss zur Startzeit automatisch aktiviert werden.

- 7.1. Erstellen Sie eine Partition mit 512 MiB auf **/dev/vdc**.

```
[root@serverb ~]# parted /dev/vdc mklabel msdos  
[root@serverb ~]# parted /dev/vdc mkpart primary linux-swap 1MiB 513MiB
```

- 7.2. Aktivieren Sie den Swap-Speicher **/dev/vdc1**.

```
[root@serverb ~]# mkswap /dev/vdc1  
...output omitted...
```

- 7.3. Aktivieren Sie den Swap-Speicher, damit er auch nach erneuten Bootvorgängen erhalten bleibt. Verwenden Sie die UUID der Struktur, wenn Sie den Eintrag in **/etc/fstab** erstellen.

```
[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
```

8. Erstellen Sie die Benutzer **production1**, **production2**, **production3** und **production4**. Stellen Sie sicher, dass sie die neue Gruppe namens **production** als ihre Ergänzungsgruppe verwenden.

- 8.1.

```
[root@serverb ~]# groupadd production  
[root@serverb ~]# for i in 1 2 3 4; do useradd -G production production$i; done
```

9. Konfigurieren Sie Ihr System so, dass es ein neues Verzeichnis namens **/run/volatile** verwendet, um die temporären Dateien zu speichern. Dateien in diesem Verzeichnis sollten zeitbasiert bereinigt werden, wenn mehr als 30 Sekunden nicht auf sie zugegriffen wurde. Die oktalen Berechtigungen für das Verzeichnis müssen **0700** lauten. Konfigurieren Sie mit der Datei **/etc/tmpfiles.d/volatile.conf** die zeitbasierte Bereinigung für die Dateien in **/run/volatile**.

- 9.1. Erstellen Sie eine neue Datei namens **/etc/tmpfiles.d/volatile.conf** mit folgendem Inhalt:

```
d /run/volatile 0700 root root 30s
```

- 9.2. Führen Sie den Befehl **systemctl-tmpfiles --create** aus, um das Verzeichnis **/run/volatile** zu erstellen, falls es noch nicht vorhanden ist.

```
[root@servera ~]# systemctl-tmpfiles --create /etc/tmpfiles.d/volatile.conf
```

10. Erstellen Sie ein neues Verzeichnis namens **/webcontent**. Der Eigentümer sowie der Gruppeneigentümer des Verzeichnisses sollten als **root** festgelegt sein. Die Gruppenmitglieder von **production** sollten dieses Verzeichnis lesen und in es schreiben können. Der Benutzer **production1** sollte nur in der Lage sein, dieses Verzeichnis zu lesen. Diese Berechtigungen sollten für alle Dateien und Verzeichnisse gelten, die im Verzeichnis **/webcontent** neu erstellt werden.

- 10.1. Erstellen Sie das Verzeichnis **/webcontent**.

```
[root@serverb ~]# mkdir /webcontent
```

- 10.2. Verwenden Sie **setfacl**, um Berechtigungen auf **/webcontent** zu konfigurieren, damit die Gruppenmitglieder von **production** über Lese- und Schreibberechtigungen dafür verfügen, mit Ausnahme des Benutzers **production1**, dem nur die Leseberechtigung gewährt werden soll.

```
[root@serverb ~]# setfacl -m u:production1:rx /webcontent
[root@serverb ~]# setfacl -m g:production:rw /webcontent
```

- 10.3. Verwenden Sie **setfacl**, um die Standardberechtigungen auf **/webcontent** festzulegen. Damit gelten die Berechtigungen, die Sie im vorherigen Schritt angewendet haben, auch für alle neuen Dateien und Verzeichnisse, die unter dem Verzeichnis **/webcontent** erstellt wurden.

```
[root@serverb ~]# setfacl -m d:u:production1:rx /webcontent
[root@serverb ~]# setfacl -m d:g:production:rw /webcontent
```

- 10.4. Beenden Sie die Shell des Benutzers **root**.

```
[root@serverb ~]# exit
logout
```

- 10.5. Melden Sie sich von **serverb** ab.

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

Bewertung

Führen Sie auf **workstation** das Skript **lab_rhcsa-compreview2 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab_rhcsa-compreview2 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-compreview2 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-compreview2 finish
```

Damit ist die ausführliche Überprüfung abgeschlossen.

► Praktische Übung

Konfigurieren und Verwalten der Serversicherheit

In dieser Überprüfung konfigurieren Sie die auf dem SSH-Schlüssel basierende Authentifizierung, ändern die Firewall-Einstellungen, passen den SELinux-Modus und einen booleschen SELinux-Wert an und beheben SELinux-Probleme.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Konfigurieren von SSH-Schlüssels für die schlüsselbasierte SSH-Authentifizierung.
- Konfigurieren von Firewall-Einstellungen.
- Anpassen des SELinux-Modus und der booleschen SELinux-Werte.
- Beheben von SELinux-Problemen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview3 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview3 start
```

Anweisungen

Führen Sie die folgenden Aufgaben aus, um die ausführliche Überprüfung abzuschließen:

- Generieren Sie SSH-Schlüssel für den Benutzer **student** auf **serverb**. Schützen Sie den Private Key nicht mit einer Passphrase.
- Konfigurieren Sie auf **servera** den Benutzer **student** so, dass die Anmeldeauthentifizierung mit dem SSH-Schlüsselpaar akzeptiert wird, das für **student** auf **serverb** erstellt wurde. Der Benutzer **student** auf **serverb** sollte in der Lage sein, sich mittels SSH ohne Eingabe eines Passworts bei **servera** anzumelden. Verwenden Sie bei Bedarf **student** als Passwort des Benutzers **student**.
- Ändern Sie auf **servera** den SELinux-Standardmodus in den Modus **permissive**.
- Konfigurieren Sie **serverb** so, dass das Benuterverzeichnis des Benutzers **production5** automatisch bereitgestellt wird, wenn sich der Benutzer mit dem Network File System **/home-directories/production5** anmeldet. Dieses Network File System wird von **servera.lab.example.com** exportiert. Passen Sie den entsprechenden booleschen SELinux-Wert an, sodass **production5** das in NFS bereitgestellte Benuterverzeichnis auf **serverb** verwenden kann, nachdem die Authentifizierung über die SSH-Schlüssel-basierte Authentifizierung vorgenommen wurde. Das Passwort für den Benutzer **production5** lautet **redhat**.

Kapitel 14 | Ausführliche Wiederholung

- Passen Sie auf **serverb** die Firewalleinstellungen an, sodass von **servera** stammende SSH-Verbindungen abgelehnt werden.
- Untersuchen und beheben Sie auf **serverb** das Problem mit dem Apache HTTPD-Daemon, der für die Überwachung des Ports **30080/TCP** konfiguriert ist, aber nicht gestartet wird. Passen Sie die Firewalleinstellungen entsprechend an, sodass Port **30080/TCP** für eingehende Verbindungen geöffnet ist.

Bewertung

Führen Sie auf **workstation** das Skript **lab rhcsa-comprevew3 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-comprevew3 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-comprevew3 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-comprevew3 finish
```

Speichern Sie die Dateien oder Arbeiten für die Verwendung auf anderen Systemen, und setzen Sie anschließend **workstation**, **servera** und **serverb** zurück.

Damit ist die ausführliche Überprüfung abgeschlossen.

► Lösung

Konfigurieren und Verwalten der Serversicherheit

In dieser Überprüfung konfigurieren Sie die auf dem SSH-Schlüssel basierende Authentifizierung, ändern die Firewall-Einstellungen, passen den SELinux-Modus und einen booleschen SELinux-Wert an und beheben SELinux-Probleme.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Konfigurieren von SSH-Schlüssels für die schlüsselbasierte SSH-Authentifizierung.
- Konfigurieren von Firewall-Einstellungen.
- Anpassen des SELinux-Modus und der booleschen SELinux-Werte.
- Beheben von SELinux-Problemen.

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** bei **workstation** an.

Führen Sie **lab rhcsa-compreview3 start** auf **workstation** aus, um die ausführliche Überprüfung zu beginnen. Dieses Skript erstellt die erforderlichen Dateien, um die Umgebung korrekt einzurichten.

```
[student@workstation ~]$ lab rhcsa-compreview3 start
```

Anweisungen

Führen Sie die folgenden Aufgaben aus, um die ausführliche Überprüfung abzuschließen:

- Generieren Sie SSH-Schlüssel für den Benutzer **student** auf **serverb**. Schützen Sie den Private Key nicht mit einer Passphrase.
- Konfigurieren Sie auf **servera** den Benutzer **student** so, dass die Anmeldeauthentifizierung mit dem SSH-Schlüsselpaar akzeptiert wird, das für **student** auf **serverb** erstellt wurde. Der Benutzer **student** auf **serverb** sollte in der Lage sein, sich mittels SSH ohne Eingabe eines Passworts bei **servera** anzumelden. Verwenden Sie bei Bedarf **student** als Passwort des Benutzers **student**.
- Ändern Sie auf **servera** den SELinux-Standardmodus in den Modus **permissive**.
- Konfigurieren Sie **serverb** so, dass das Benuterverzeichnis des Benutzers **production5** automatisch bereitgestellt wird, wenn sich der Benutzer mit dem Network File System **/home-directories/production5** anmeldet. Dieses Network File System wird von **servera.lab.example.com** exportiert. Passen Sie den entsprechenden booleschen SELinux-Wert an, sodass **production5** das in NFS bereitgestellte Benuterverzeichnis auf **serverb** verwenden kann, nachdem die Authentifizierung über die SSH-Schlüssel-basierte Authentifizierung vorgenommen wurde. Das Passwort für den Benutzer **production5** lautet **redhat**.

Kapitel 14 | Ausführliche Wiederholung

- Passen Sie auf **serverb** die Firewalleinstellungen an, sodass von **servera** stammende SSH-Verbindungen abgelehnt werden.
- Untersuchen und beheben Sie auf **serverb** das Problem mit dem Apache HTTPD-Daemon, der für die Überwachung des Ports **30080/TCP** konfiguriert ist, aber nicht gestartet wird. Passen Sie die Firewalleinstellungen entsprechend an, sodass Port **30080/TCP** für eingehende Verbindungen geöffnet ist.

1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **serverb**.

1.1.

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

2. Führen Sie den Befehl **ssh-keygen** aus, um die SSH-Schlüssel für den Benutzer **student** auf **serverb** zu generieren. Schützen Sie den Private Key nicht mit einer Passphrase.

2.1.

```
[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:  
SHA256:1TPZ4TXYWiGWfExUGtRTHgfKQbF9hVuLa+VmH4vgkFY student@serverb.lab.example.com  
The key's randomart image is:  
+---[RSA 2048]---+  
| .+@B0** |  
| .=.#+B* |  
| . X.*o= |  
| . E +.+ |  
| S o + |  
| + . o = |  
| . o o + +|  
| . . . . |  
| |  
+---[SHA256]-----+
```

3. Konfigurieren Sie auf **servera** den Benutzer **student** so, dass die Anmeldeauthentifizierung mit dem SSH-Schlüsselpaar akzeptiert wird, das Sie für **student** auf **serverb** erstellt haben. Der Benutzer **student** auf **serverb** sollte in der Lage sein, sich mittels SSH ohne Eingabe eines Passworts bei **servera** anzumelden. Verwenden Sie bei Bedarf **student** als Passwort des Benutzers **student**.

- 3.1. Führen Sie den Befehl **ssh-copy-id** aus, um den Public Key des SSH-Schlüsselpaares von **student** auf **serverb** an **student** auf **servera** zu übertragen. Verwenden Sie bei Bedarf **student** als Passwort des Benutzers **student**.

```
[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.
ECDSA key fingerprint is SHA256:g/fIMtVzDWTbTi1l00WC30sL6cHmro9Tf563NxmeyyE.
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
student@servera's password: student

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'student@servera'"
and check to make sure that only the key(s) you wanted were added.
```

- 3.2. Führen Sie den Befehl **ssh** aus, um zu verifizieren, dass sich der Benutzer **student** über **serverb** bei **servera** anmelden kann, ohne ein Passwort einzugeben.

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

4. Ändern Sie auf **servera** den SELinux-Standardmodus in den Modus **permissive**.

- 4.1. Bearbeiten Sie **/etc/sysconfig/selinux**, um den Wert des Parameters **SELINUX** auf **permissive** festzulegen. Sie können den Befehl **sudo vi /etc/sysconfig/
selinux** ausführen, um die Konfigurationsdatei als Superuser zu bearbeiten. Verwenden Sie bei Aufforderung **student** als Passwort.

```
...output omitted...
#SELINUX=enforcing
SELINUX=permissive
...output omitted...
```

- 4.2. Führen Sie den Befehl **sudo systemctl reboot** aus, um das System als Superuser neu zu starten.

```
[student@servera ~]$ sudo systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@serverb ~]$
```

5. Konfigurieren Sie **serverb** so, dass das Benutzerverzeichnis des Benutzers **production5** automatisch bereitgestellt wird, wenn sich der Benutzer mit dem Network File System / **home-directories/production5** anmeldet. Dieses Network File System wird von **servera.lab.example.com** exportiert. Passen Sie den entsprechenden booleschen SELinux-Wert an, sodass **production5** das in NFS bereitgestellte Benutzerverzeichnis auf **serverb** verwenden kann, nachdem die Authentifizierung über die SSH-Schlüssel-basierte Authentifizierung vorgenommen wurde. Das Passwort für den Benutzer **production5** lautet **redhat**.

Kapitel 14 | Ausführliche Wiederholung

- 5.1. Führen Sie auf **serverb** den Befehl **sudo -i** aus, um zum Benutzerkonto **root** zu wechseln.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 5.2. Installieren Sie das Paket **autofs**.

```
[root@serverb ~]# yum install autofs  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Installed:  
  autofs-1:5.1.4-29.el8.x86_64  
  
Complete!
```

- 5.3. Erstellen Sie die Master-Map-Datei **autofs** namens **/etc/auto.master.d/production5.autofs** mit dem folgenden Inhalt.

```
/- /etc/auto.production5
```

- 5.4. Rufen Sie die Details des Benutzers **production5** ab, um den Pfad für das Benutzerverzeichnis abzurufen.

```
[root@serverb ~]# getent passwd production5  
production5:x:5001:5001::/localhome/production5:/bin/bash
```

- 5.5. Erstellen Sie die Datei **/etc/auto.production5** mit dem folgenden Inhalt.

```
/localhome/production5 -rw servera.lab.example.com:/home-directories/production5
```

- 5.6. Starten Sie den Service **autofs** neu.

```
[root@serverb ~]# systemctl restart autofs
```

6. Vergewissern Sie sich auf **servera**, dass sich der Benutzer **production5** nicht bei **serverb** über die SSH-Public-Key-Authentifizierung anmelden kann. Ein boolescher SELinux-Wert verursacht dieses Problem, das Sie in den folgenden Schritten beheben werden.

- 6.1. Öffnen Sie auf **workstation** als **student** eine SSH-Sitzung zu **servera**.

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

- 6.2. Wechseln Sie zum Benutzer **production5** und verwenden Sie als Passwort **redhat**.

```
[student@servera ~]$ su - production5  
Password: redhat  
[production5@servera ~]$
```

- 6.3. Generieren Sie mit dem Befehl **ssh-keygen** die SSH-Schlüssel als **production5**.

```
[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:zmin1nmCt4H8LA+4FPimtdg81n17ATbInUFW3HSPxk4  
production5@servera.lab.example.com  
The key's randomart image is:  
+---[RSA 2048]---+  
|       .00.0. . |  
|       ... .0 0 |  
|     . o o     E . |  
|   . o *     +   |  
| .. .So       . |  
| . + =     . |  
| *.*+=. . |  
| 0o+***.o |  
| o.=o.=** |  
+---[SHA256]---+
```

- 6.4. Führen Sie den Befehl **ssh-copy-id** aus, um den Public Key des SSH-Schlüsselpaares von **production5** auf **servera** an **production5** auf **serverb** zu übertragen. Wenn Sie dazu aufgefordert werden, geben Sie **redhat** als Passwort des Benutzers **production5** ein.

```
[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 into the machine, with: "ssh 'production5@serverb'"  
and check to make sure that only the key(s) you wanted were added.
```

Kapitel 14 | Ausführliche Wiederholung

- 6.5. Verwenden Sie die SSH-Public-Key-Authentifizierung statt der passwortbasierten Authentifizierung, um sich bei **serverb** als **production5** anzumelden. Dieser Befehl muss fehlschlagen.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \  
-o passwordauthentication=no production5@serverb  
production5@serverb: Permission denied (publickey,gssapi-keyex,gssapi-with-  
mic,password).
```

7. Legen Sie die entsprechende boolesche SELinux-Einstellung auf **serverb** fest, damit **production5** sich bei **serverb** mit der SSH-Public-Key-Authentifizierung anmelden und das Benutzerverzeichnis verwenden kann.

- 7.1. Legen Sie auf **serverb** als **root** den booleschen SELinux-Wert **use_nfs_home_dirs** auf **true** fest.

```
[root@serverb ~]# setsebool -P use_nfs_home_dirs true
```

- 7.2. Verwenden Sie die SSH-Public-Key-Authentifizierung statt der passwortbasierten Authentifizierung, um sich bei **serverb** als **production5** anzumelden. Dieser Befehl sollte erfolgreich sein.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \  
-o passwordauthentication=no production5@serverb  
...output omitted...  
[production5@serverb ~]$
```

8. Passen Sie auf **serverb** die Firewalleinstellungen an, sodass von **servera** stammende SSH-Verbindungen abgelehnt werden. Das System **servera** verwendet die IPv4-Adresse **172.25.250.10**.

- 8.1. Führen Sie den Befehl **firewall-cmd** aus, um der **firewalld**-Zone namens **block** die IPv4-Adresse von **servera** hinzuzufügen.

```
[root@serverb ~]# firewall-cmd --add-source=172.25.250.10/32 \  
--zone=block --permanent  
success
```

- 8.2. Führen Sie den Befehl **firewall-cmd --reload** aus, um die Änderungen in den Firewalleinstellungen neu zu laden.

```
[root@serverb ~]# firewall-cmd --reload  
success
```

9. Untersuchen und beheben Sie auf **serverb** das Problem mit dem Apache HTTPD-Daemon, der für die Überwachung des Ports **30080/TCP** konfiguriert ist, aber nicht gestartet wird. Passen Sie die Firewalleinstellungen entsprechend an, sodass Port **30080/TCP** für eingehende Verbindungen geöffnet ist.

- 9.1. Führen Sie den Befehl **systemctl** aus, um den **httpd**-Service neu zu starten. Dieser Befehl kann den Service nicht neu starten.

```
[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 -xe" for details.
```

- 9.2. Führen Sie den Befehl **systemctl status** aus, um den Grund für den Fehler des **httpd**-Services zu untersuchen.

```
[root@serverb ~]# systemctl status httpd.service
● 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 2019-04-15 06:42:41 EDT; 5min ago
    Docs: man:httpd.service(8)
   Process: 27313 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND (code=exited, status=1/FAILURE)
   Main PID: 27313 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."

Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Starting The Apache HTTP Server...
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission denied:
AH00072: make_sock: could not bind to address [::]:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission denied:
AH00072: make_sock: could not bind to address 0.0.0.0:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: no listening sockets available, shutting down
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: AH00015: Unable to open logs
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service: Main process exited, code=exited, status=1/FAILURE
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service: Failed with result 'exit-code'.
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Failed to start The Apache HTTP Server.
```

Beachten Sie den Berechtigungsfehler in der vorhergehenden Ausgabe, der angibt, dass der **httpd**-Daemon keine Verbindung mit Port **30080/TCP** herstellen konnte. Die SELinux-Richtlinie kann eine potenzielle Einschränkung für eine Anwendung sein, die eine Verbindung zu einem Port herstellt. Drücken Sie **q**, um den vorherigen **systemctl**-Befehl zu beenden.

- 9.3. Führen Sie den Befehl **sealert** aus, um zu ermitteln, ob eine SELinux-Richtlinie **httpd** daran hindert, eine Verbindung zu Port **30080/TCP** herzustellen.

```
[root@serverb ~]# 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 30080.

***** Plugin bind_ports (92.2 confidence) suggests *****
```

Kapitel 14 | Ausführliche Wiederholung

```
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...
```

Die vorhergehende Protokollmeldung zeigt, dass der Port **30080/TCP** nicht den entsprechenden SELinux-Kontext **http_port_t** aufweist. Deshalb verhindert SELinux, dass **httpd** eine Verbindung zu diesem Port herstellt. Die Protokollmeldung generiert auch die Syntax des Befehls **semanage port**, sodass Sie das Problem leicht beheben können.

- 9.4. Führen Sie den Befehl **semanage port** aus, um den entsprechenden SELinux-Kontext für den Port **30080/TCP** für die Verbindung mit **httpd** festzulegen.

```
[root@serverb ~]# semanage port -a -t http_port_t -p tcp 30080
```

- 9.5. Führen Sie den Befehl **systemctl** aus, um **httpd** neu zu starten. Dieser Befehl sollte den Service erfolgreich neu starten.

```
[root@serverb ~]# systemctl restart httpd
```

- 9.6. Fügen Sie der Standardzone **firewalld** namens **public** den Port **30080/TCP** hinzu.

```
[root@serverb ~]# firewall-cmd --add-port=30080/tcp --permanent  
success  
[root@serverb ~]# firewall-cmd --reload  
success
```

- 9.7. Beenden Sie die Shell des Benutzers **root**.

```
[root@serverb ~]# exit  
logout
```

- 9.8. Melden Sie sich von **serverb** ab.

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

Bewertung

Führen Sie auf **workstation** das Skript **lab_rhcsa-compreview3 grade** aus, um den Erfolg dieser Übung zu bestätigen. Beheben Sie sämtliche gemeldeten Fehler und führen Sie das Skript so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab_rhcsa-compreview3 grade
```

Beenden

Führen Sie auf der **workstation** den Befehl **lab rhcsa-compreview3 finish** aus, um diese Übung zu beenden. Dieses Skript löscht die während der Übung erstellten Dateien und Ressourcen und sorgt für eine bereinigte Umgebung.

```
[student@workstation ~]$ lab rhcsa-compreview3 finish
```

Speichern Sie die Dateien oder Arbeiten für die Verwendung auf anderen Systemen, und setzen Sie anschließend **workstation**, **servera** und **serverb** zurück.

Damit ist die ausführliche Überprüfung abgeschlossen.

► Praktische Übung

Ausführen von Containern

In dieser Wiederholung konfigurieren Sie einen Container auf Ihrem Server, der Webinhalt über den persistenten Storage bereitstellt und automatisch mit dem Server gestartet wird.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von getrennten Containern ohne Root
- Konfigurieren der Portumleitung und des persistenten Storages
- Konfigurieren von **systemd** für Container für den Start beim Start des Hostrechners

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab rhcsa-compreview4 start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem wird eine Archivdatei mit einigen Webinhalten und dem Benutzerkonto **container** erstellt, mit dem Sie einen Apache HTTP Server-Container ausführen.

```
[student@workstation ~]$ lab rhcsa-compreview4 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** als Benutzer **containers** aus, um die ausführliche Wiederholung abzuschließen:

- Erstellen Sie das Verzeichnis **/srv/web/** auf **serverb**. Extrahieren Sie dann das Archiv **/home/containers/rhcsa-compreview4/web-content.tgz** in dieses Verzeichnis. Konfigurieren Sie das Verzeichnis so, dass es von einem Container ohne Root für persistenten Storage verwendet werden kann.
- Installieren Sie die Container-Tools auf **serverb**.
- Erstellen Sie als Benutzer **containers** auf **serverb** einen getrennten Apache HTTP Server-Container mit dem Namen **web**. Verwenden Sie das Image **rhel8/httpd-24** mit dem Tag **1-105** aus der Registry **registry.lab.example.com**. Ordnen Sie Port 8080 im Container Port 8888 auf dem Host zu. Mounten Sie das Verzeichnis **/srv/web** auf dem Host als **/var/www** im Container. Deklarieren Sie die Umgebungsvariable **HTTPD_MPM** mit **event** für den Wert.
- Konfigurieren Sie **systemd** als Benutzer **containers** auf **serverb** so, dass der Container **web** automatisch mit dem Server gestartet wird.

Das Passwort für den Benutzer **containers** lautet **redhat**. Verwenden Sie für den Zugriff auf die unter **registry.lab.example.com** befindliche Container-Image-Registry das Konto **admin** mit dem Passwort **redhat321**. Sie können die **web**-Container-Parameter aus

der Datei **/home/containers/rhcsa-compreview4/variables** auf **serverb** kopieren und einfügen.

Bewertung

Verwenden Sie als Benutzer **student** auf dem Rechner **workstation** den Befehl **lab rhcsa-compreview4 grade**, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-compreview4 grade
```

Beenden

Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab rhcsa-compreview4 finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab rhcsa-compreview4 finish
```

Damit ist die umfassende Wiederholung abgeschlossen.

► Lösung

Ausführen von Containern

In dieser Wiederholung konfigurieren Sie einen Container auf Ihrem Server, der Webinhalt über den persistenten Storage bereitstellt und automatisch mit dem Server gestartet wird.

Ergebnisse

Es werden folgende Fähigkeiten vermittelt:

- Erstellen von getrennten Containern ohne Root
- Konfigurieren der Portumleitung und des persistenten Storages
- Konfigurieren von **systemd** für Container für den Start beim Start des Hostrechners

Bevor Sie Beginnen

Melden Sie sich als Benutzer **student** mit dem Passwort **student** auf dem Rechner **workstation** an.

Führen Sie auf dem Rechner **workstation** den Befehl **lab rhcsa-compreview4 start** aus. Dieser Befehl führt ein Startskript aus, das ermittelt, ob der Rechner **serverb** im Netzwerk erreichbar ist. Außerdem wird eine Archivdatei mit einigen Webinhalten und dem Benutzerkonto **container** erstellt, mit dem Sie einen Apache HTTP Server-Container ausführen.

```
[student@workstation ~]$ lab rhcsa-compreview4 start
```

Anweisungen

Führen Sie die folgenden Aufgaben auf **serverb** als Benutzer **containers** aus, um die ausführliche Wiederholung abzuschließen:

- Erstellen Sie das Verzeichnis **/srv/web/** auf **serverb**. Extrahieren Sie dann das Archiv **/home/containers/rhcsa-compreview4/web-content.tgz** in dieses Verzeichnis. Konfigurieren Sie das Verzeichnis so, dass es von einem Container ohne Root für persistenten Storage verwendet werden kann.
- Installieren Sie die Container-Tools auf **serverb**.
- Erstellen Sie als Benutzer **containers** auf **serverb** einen getrennten Apache HTTP Server-Container mit dem Namen **web**. Verwenden Sie das Image **rhel8/httpd-24** mit dem Tag **1-105** aus der Registry **registry.lab.example.com**. Ordnen Sie Port 8080 im Container Port 8888 auf dem Host zu. Mounten Sie das Verzeichnis **/srv/web** auf dem Host als **/var/www** im Container. Deklarieren Sie die Umgebungsvariable **HTTPD_MPM** mit **event** für den Wert.
- Konfigurieren Sie **systemd** als Benutzer **containers** auf **serverb** so, dass der Container **web** automatisch mit dem Server gestartet wird.

Das Passwort für den Benutzer **containers** lautet **redhat**. Verwenden Sie für den Zugriff auf die unter **registry.lab.example.com** befindliche Container-Image-Registry das Konto **admin** mit dem Passwort **redhat321**. Sie können die **web**-Container-Parameter aus

der Datei **/home/containers/rhcsa-comprevew4/variables** auf **serverb** kopieren und einfügen.

1. Erstellen Sie das Verzeichnis **/srv/web/** auf **serverb**. Extrahieren Sie dann das Archiv **/home/containers/rhcsa-comprevew4/web-content.tgz** in dieses Verzeichnis. Konfigurieren Sie das Verzeichnis so, dass es von einem Container ohne Root für persistenten Storage verwendet werden kann.
 - 1.1. Melden Sie sich mit dem Befehl **ssh** bei **serverb** als Benutzer **containers** an. Die Systeme sind für die Verwendung von SSH-Schlüsseln zur Authentifizierung konfiguriert. Daher ist kein Passwort erforderlich.

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

- 1.2. Verwenden Sie den Befehl **sudo -i**, um zum Benutzer **root** zu wechseln. Das Passwort für den Benutzer **containers** lautet **redhat**.

```
[containers@serverb ~]$ sudo -i
[sudo] password for containers: redhat
[root@serverb ~]#
```

- 1.3. Erstellen Sie das Verzeichnis **/srv/web/**.

```
[root@serverb ~]# mkdir /srv/web/
[root@serverb ~]#
```

- 1.4. Extrahieren Sie das Archiv **/home/containers/rhcsa-comprevew4/web-content.tgz** in das Verzeichnis **/srv/web/**.

```
[root@serverb ~]# cd /srv/web/
[root@serverb web]# tar xvf /home/containers/rhcsa-comprevew4/web-content.tgz
html/
html/index.html
[root@serverb web]#
```

- 1.5. Container ohne Root benötigen Lesezugriff auf das Verzeichnis **/srv/web/** und auf dessen Inhalte. Zudem muss der über den Benutzer **containers** ausgeführte Befehl **podman** das Verzeichnis für SELinux umbenennen können. Legen Sie den Verzeichniseigentümer auf **container** fest. Bestätigen Sie dann, dass jeder Zugriff auf den Inhalt hat.

```
[root@serverb web]# chown -R containers: /srv/web
[root@serverb web]# ls -ld /srv/web/
drwxr-xr-x. 3 containers containers 18 Sep  7 04:43 /srv/web/
[root@serverb web]# ls -ld /srv/web/html/
drwxr-xr-x. 2 containers containers 24 Sep  7 04:01 /srv/web/html/
[root@serverb web]# ls -l /srv/web/html/index.html
-rw-r--r--. 1 containers containers 546 Sep  7 04:01 /srv/web/html/index.html
```

Kapitel 14 | Ausführliche Wiederholung

2. Installieren Sie die Container-Tools auf **serverb**.

- 2.1. Installieren Sie das Yum-Modul **container-tools** mit dem Befehl **yum**.

```
[root@serverb web]# yum module install container-tools
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- 2.2. Beenden Sie das **root**-Konto.

```
[root@serverb web]# exit
logout
[containers@serverb ~]$
```

3. Erstellen Sie als Benutzer **containers** auf **serverb** einen getrennten Container mit dem Namen **web**. Verwenden Sie das Image **rhe18/httpd-24** mit dem Tag **1-105** aus der Registry **registry.lab.example.com**. Ordnen Sie Port 8080 im Container Port 8888 auf dem Host zu. Mounten Sie das Verzeichnis **/srv/web** auf dem Host als **/var/www** im Container. Deklarieren Sie die Umgebungsvariable **HTTPD_MPM** mit dem Wert **event**.
Sie können diese Parameter aus der Datei **/home/containers/rhcsa-compreview4/variables** auf **serverb** kopieren und einfügen.

- 3.1. Melden Sie sich bei der unter **registry.lab.example.com** befindliche Container-Image-Registry über das Konto **admin** mit dem Passwort **redhat321** an.

```
[containers@serverb ~]$ podman login registry.lab.example.com
Username: admin
Password: redhat321
Login Succeeded!
```

- 3.2. Verwenden Sie den Befehl **podman run**, um den Container zu erstellen. Der folgende **podman run**-Befehl ist sehr lang und sollte als eine einzelne Zeile eingegeben werden.

```
[containers@serverb ~]$ podman run -d --name web -p 8888:8080 -v /srv/web:/var/www:Z -e HTTPD_MPM=event registry.lab.example.com/rhel8/httpd-24:1-105
...output omitted...
```

- 3.3. Bestätigen Sie mit dem Befehl **curl**, dass Apache HTTP Server ausgeführt wird.

```
[containers@serverb ~]$ curl http://localhost:8888/
Comprehensive Review Web Content Test

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sit amet lacus vestibulum, varius magna sit amet, tempus neque.
...output omitted...
```

Dieser Inhalt stammt aus dem zuvor von Ihnen extrahierten Archiv **web-content.tgz**.

4. Konfigurieren Sie **systemd** als Benutzer **containers** auf **serverb** so, dass der Container **web** automatisch mit dem Server gestartet wird.

- 4.1. Wenn Sie **sudo** oder **su** bei der Anmeldung als Benutzer **containers** verwendet haben, beenden Sie **serverb**. Verwenden Sie dann den Befehl **ssh**, um sich als Benutzer **containers** direkt auf **serverb** anzumelden.

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

- 4.2. Erstellen Sie das Verzeichnis **~/.config/systemd/user/**.

```
[containers@serverb ~]$ mkdir -p ~/.config/systemd/user/  
[containers@serverb ~]$
```

- 4.3. Verwenden Sie den Befehl **podman generate systemd**, um die **systemd**-Unit-Datei aus dem ausgeführten Container zu erstellen.

```
[containers@serverb ~]$ cd ~/.config/systemd/user/  
[containers@serverb user]$ podman generate systemd --name web --files --new  
/home/containers/.config/systemd/user/container-web.service
```

- 4.4. Halten Sie den Container **web** an, und löschen Sie ihn.

```
[containers@serverb user]$ podman stop web  
d16a826c936efc7686d8d8e5617b727f5d272361c54f8a0ca65c57d012347784  
[containers@serverb user]$ podman rm web  
d16a826c936efc7686d8d8e5617b727f5d272361c54f8a0ca65c57d012347784
```

- 4.5. Weisen Sie **systemd** an, seine Konfiguration neu zu laden. Aktivieren und starten Sie anschließend den Service **container-web**.

```
[containers@serverb user]$ systemctl --user daemon-reload  
[containers@serverb user]$ systemctl --user enable --now container-web.service  
Created symlink /home/containers/.config/systemd/user/multi-user.target.wants/  
container-web.service → /home/containers/.config/systemd/user/container-  
web.service.  
Created symlink /home/containers/.config/systemd/user/default.target.wants/  
container-web.service → /home/containers/.config/systemd/user/container-  
web.service.
```

- 4.6. Bestätigen Sie, dass der Container ausgeführt wird.

```
[containers@serverb user]$ curl http://localhost:8888/  
Comprehensive Review Web Content Test  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Sed sit amet lacus vestibulum, varius magna sit amet, tempus neque.  
...output omitted...
```

- 4.7. Führen Sie den Befehl **logindctl enable-linger** aus, um die Benutzerservices automatisch mit dem Server zu starten.

```
[containers@serverb ~]$ logindctl enable-linger
[containers@serverb ~]$
```

4.8. Beenden Sie **serverb**.

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

Bewertung

Verwenden Sie als Benutzer **student** auf dem Rechner **workstation** den Befehl **lab rhcsa-compreview4 grade**, um Ihre Arbeit zu bewerten. Beheben Sie sämtliche gemeldeten Fehler, und führen Sie den Befehl so lange erneut aus, bis die Durchführung erfolgreich ist.

```
[student@workstation ~]$ lab rhcsa-compreview4 grade
```

Beenden

Führen Sie auf dem Rechner **workstation** als Benutzer **student** den Befehl **lab rhcsa-compreview4 finish** aus, um diese Übung zu beenden.

```
[student@workstation ~]$ lab rhcsa-compreview4 finish
```

Damit ist die umfassende Wiederholung abgeschlossen.