








Leseprobe

Michael Koflers Standardwerk gibt Ihnen einen vollständigen Überblick über das große Thema »Linux«. In dieser Leseprobe lernen Sie die Grundlagen der Netzwerkverwaltung und die Administration von Webservern und virtuellen Maschinen kennen. Außerdem können Sie einen Blick in das Inhaltsverzeichnis und das gesamte Stichwortverzeichnis werfen.

-  **»Was ist Linux?«**
»Netzwerk-Tools«
»Apache«
»VirtualBox und Vagrant«
-  **Inhaltsverzeichnis**
-  **Index**
-  **Der Autor**
-  **Leseprobe weiterempfehlen**

Michael Kofler

Linux – Das umfassende Handbuch

1.450 Seiten, gebunden, 15. Auflage, September 2017
49,90 Euro, ISBN 978-3-8362-5854-8

 www.rheinwerk-verlag.de/4465

Kapitel 1

Was ist Linux?

Um die einleitende Frage zu beantworten, erkläre ich in diesem Kapitel zuerst einige wichtige Begriffe, die im gesamten Buch immer wieder verwendet werden: Betriebssystem, Unix, Distribution, Kernel etc. Ein knapper Überblick über die Merkmale von Linux und die verfügbaren Programme macht deutlich, wie weit die Anwendungsmöglichkeiten von Linux reichen. Es folgt ein kurzer Ausflug in die Geschichte von Linux: Sie erfahren, wie Linux entstanden ist und auf welchen Komponenten es basiert.

Von zentraler Bedeutung ist dabei natürlich die *General Public License* (kurz GPL), die angibt, unter welchen Bedingungen Linux weitergegeben werden darf. Erst die GPL macht Linux zu einem freien System, wobei »frei« mehr heißt als einfach »kostenlos«.

1.1 Einführung

Linux ist ein Unix-ähnliches Betriebssystem. Der wichtigste Unterschied gegenüber historischen Unix-Systemen besteht darin, dass Linux zusammen mit dem vollständigen Quellcode frei kopiert werden darf.

Ein Betriebssystem ist ein Bündel von Programmen, mit denen die grundlegendsten Funktionen eines Rechners realisiert werden: die Schnittstelle zwischen Mensch und Maschine (also konkret: die Verwaltung von Tastatur, Bildschirm etc.) und die Verwaltung der Systemressourcen (CPU-Zeit, Speicher etc.). Sie benötigen ein Betriebssystem, damit Sie ein Anwendungsprogramm überhaupt starten und eigene Daten in einer Datei speichern können. Populäre Betriebssysteme sind Windows, Linux, BSD, macOS und iOS.

Betriebssystem

Schon lange vor Windows, Linux oder macOS gab es Unix. Dieses Betriebssystem war technisch gesehen seiner Zeit voraus: echtes Multitasking, eine Trennung der Prozesse voneinander, klar definierte Zugriffsrechte für Dateien, ausgereifte Netzwerkfunktionen etc. Allerdings bot Unix anfänglich nur eine spartanische Benutzeroberfläche und stellte hohe Hardware-Anforderungen. Das erklärt, warum Unix fast ausschließlich im wissenschaftlichen und industriellen Bereich eingesetzt wurde.

Unix versus Linux

In seiner Verbreitung hat Linux Unix fast vollständig verdrängt: Große Teile des Internets (z. B. Google) werden heute von Linux getragen. Linux läuft auf herkömmlichen Rechnern, in Form von Android auf Smartphones und Tablets, auf Embedded Devices (z. B. ADSL-Routern, NAS-Festplatten) und in Supercomputern: Mehr als 99 Prozent der 500 schnellsten Rechner der Welt laufen heute unter Linux (<https://top500.org/statistics/list>).

Kernel Genau genommen bezeichnet der Begriff Linux nur den Kernel: Er ist der innerste Teil (Kern) eines Betriebssystems mit ganz elementaren Funktionen, wie Speicherverwaltung, Prozessverwaltung und Steuerung der Hardware. Die Informationen in diesem Buch beziehen sich auf den Kernel 4.*n*.

1.2 Hardware-Unterstützung

Linux unterstützt beinahe die gesamte gängige PC-Hardware und läuft darüber hinaus auch auf anderen Hardware-Plattformen, z. B. auf Smartphones mit ARM-CPU. Dennoch müssen Sie beim Kauf eines neuen Rechners aufpassen. Es gibt einige Hardware-Komponenten, die im Zusammenspiel mit Linux oft Probleme machen:

- ▶ **Grafikkarten:** Fast alle auf dem Markt vertretenen Grafikkarten bzw. in die CPU integrierten Grafik-Cores funktionieren unter Linux. Für viele Linux-Anwender ohne besondere Anforderungen an das Grafiksystem sind Intel-CPU mit eingebautem Grafik-Core die optimale Lösung. Neue Grafikkarten von NVIDIA und ATI/AMD erfordern hingegen oft Zusatztreiber, damit die Karte perfekt genutzt werden kann. Die Installation dieser Treiber bereitet oft Probleme.
- ▶ **Hybrid-Grafiksysteme:** Besonders problematisch sind Grafiksysteme, bei denen ein energiesparender interner Grafik-Core mit einer schnelleren externen Grafikkarte kombiniert wird. Mit geeigneten Windows- oder macOS-Treibern wechselt das Betriebssystem im laufenden Betrieb zwischen dem Grafik-Core und der Grafikkarte. Unter Linux funktioniert das – wenn überhaupt – nur nach einer komplizierten Konfiguration.
- ▶ **Hochauflösende Displays:** Hardware-technisch kommt Linux mit nahezu allen auf dem Markt erhältlichen Bildschirmen zurecht. Relativ schlecht ist leider die Software-Unterstützung von hochauflösenden Displays, die je nach Marketing-Sprech Retina-, XHD- oder HiDPI-Displays, 4k- oder 5k-Monitore genannt werden. Wenn Sie das Display in der vollen Auflösung betreiben, bleiben Teile der Benutzeroberfläche unleserlich klein.
- ▶ **WLAN- und Netzwerkadapter:** WLAN- und LAN-Controller machen selten Probleme. Nur ganz neue Modelle werden von Linux mitunter noch nicht unterstützt.

- ▶ **SSD-Cache:** Manche Notebooks kombinieren eine herkömmliche Festplatte mit einer kleinen SSD. In der Theorie erhalten Sie damit das Beste aus beiden Welten, also viel Speicherplatz und hohe Geschwindigkeit für wenig Geld. Die Praxis sieht schon unter Windows weit weniger rosig aus. Wenn dann auch noch Linux ins Spiel kommt, ist es mit den Vorteilen des SSD-Caches ganz vorbei. Im besten Fall ignoriert Linux den SSD-Cache ganz einfach und läuft so, als gäbe es nur eine herkömmliche Festplatte; im ungünstigsten Fall verursachen Sie ein defektes Dateisystem, wenn Sie unter Linux in eine Windows-Partition schreiben, deren Daten sich teilweise im SSD-Cache befinden. Investieren Sie ein paar Euro mehr in ein Notebook, das nur eine SSD enthält – es lohnt sich!
- ▶ **Energiesparfunktionen:** Gerade neue Notebooks haben unter Linux oft deutlich kürzere Akku-Laufzeiten als unter Windows. Dieses Ärgernis resultiert daraus, dass das Zusammenspiel diverser Energiesparfunktionen optimale Treiber voraussetzt, die für Linux oft gar nicht oder erst ein, zwei Jahre nach der Markteinführung verfügbar sind.

Stellen Sie also vor dem Kauf eines neuen Rechners bzw. einer Hardware-Erweiterung sicher, dass alle Komponenten von Linux unterstützt werden. Auch eine Internet-suche nach *linux <hardwarename>* kann nicht schaden. Lesenswert sind außerdem Testberichte der Zeitschrift c't: Deren Redakteure machen sich bei den meisten Geräten die Mühe, auch die Linux-Kompatibilität zu testen – zuletzt z. B. in diesem Artikel über Business-Notebooks:

<http://heise.de/-3678376> (kostenpflichtig)

Checkliste für das ideale Linux-Notebook bzw. den idealen Linux-PC

Wenn ich mir einen neuen Rechner kaufe, achte ich zumeist auf die folgenden Punkte:

- ▶ **CPU und Grafik:** Für mich kommt nur eine in die 64-Bit-CPU integrierte Grafiklösung infrage, die mit Open-Source-Treibern gut funktioniert. Diese Voraussetzungen erfüllen die meisten Intel-CPU.
- ▶ **Display:** Bei Notebooks erspare ich mir den Aufpreis für ein Display mit mehr als 1920 × 1080 Pixel – eine optimale Nutzung ist unter Linux leider schwer möglich. Immerhin ist gerade beim Desktop-System Gnome Besserung in Sicht (im wahren Sinne des Wortes); man kann also argumentieren, dass ein XHD-Display eine Investition für die Zukunft ist.
- ▶ **Speicher:** Es muss eine SSD sein. Größere Datenmengen speichere ich extern auf einem NAS-Gerät, in einem Cloud-Speicher etc.
- ▶ **Kein Windows:** Nach Möglichkeit kaufe ich Geräte ohne vorinstalliertes Windows, auch wenn die Preisersparnis gering ist.
- ▶ **Lieber etwas älter:** Um ganz neue Geräte mache ich nach Möglichkeit einen großen Bogen, auch wenn die Spezifikationen noch so verlockend sind.

1.3 Distributionen

Noch immer ist die einleitende Frage – Was ist Linux? – nicht ganz beantwortet. Viele Anwender interessiert der Kernel nämlich herzlich wenig, sofern er nur läuft und die vorhandene Hardware unterstützt. Für sie umfasst der Begriff Linux, wie er umgangssprachlich verwendet wird, neben dem Kernel auch das riesige Bündel von Programmen, das mit Linux mitgeliefert wird: Dazu zählen neben unzähligen Kommandos die Desktop-Systeme KDE und Gnome, das Office-Paket LibreOffice, der Webbrowser Firefox, das Zeichenprogramm GIMP sowie zahllose Programmiersprachen und Server-Programme (Webserver, Mail-Server, File-Server etc.).

Als Linux-Distribution wird die Einheit bezeichnet, die aus dem eigentlichen Betriebssystem (Kernel) und seinen Zusatzprogrammen besteht. Eine Distribution ermöglicht eine rasche und bequeme Installation von Linux. Die meisten Distributionen können kostenlos aus dem Internet heruntergeladen werden.

Distributionen unterscheiden sich vor allem durch folgende Punkte voneinander:

- **Umfang, Aktualität:** Die Anzahl, Auswahl und Aktualität der mitgelieferten Programme und Bibliotheken variiert stark. Manche Distributionen setzen bewusst auf etwas ältere, stabile Versionen – z. B. Debian.
- **Installations- und Konfigurationswerkzeuge:** Die mitgelieferten Programme zur Installation, Konfiguration und Wartung des Systems helfen dabei, die Konfigurationsdateien einzustellen. Das kann viel Zeit sparen.
- **Konfiguration des Desktops (KDE, Gnome):** Manche Distributionen lassen dem Anwender die Wahl zwischen KDE, Gnome und anderen Desktop-Systemen. Auch die Detailkonfiguration und optische Gestaltung variiert je nach Distribution.
- **Hardware-Unterstützung:** Linux kommt mit den meisten PC-Hardware-Komponenten zurecht. Dennoch gibt es im Detail Unterschiede zwischen den Distributionen, insbesondere wenn es darum geht, Nicht-Open-Source-Treiber (z. B. für NVIDIA-Grafikkarten) in das System zu integrieren.
- **Updates:** Sie können eine Linux-Distribution nur so lange sicher betreiben, wie Sie Updates bekommen. Danach ist aus Sicherheitsgründen ein Wechsel auf eine neue Version der Distribution erforderlich. Deswegen ist es bedeutsam, wie lange es für eine Distribution Updates gibt. Hier gilt meist die Grundregel: je teurer der Support, desto länger der Zeitraum. Einige Beispiele (Stand: Frühjahr 2017):

Fedora:	13 Monate
openSUSE Leap:	ca. 18 bis 24 Monate
Red Hat Enterprise Linux:	10 Jahre (mit Einschränkungen sogar 13 Jahre)
SUSE Enterprise Server:	10 Jahre (mit Einschränkungen sogar 13 Jahre)
Ubuntu LTS:	3 bis 5 Jahre
Ubuntu (sonstige Versionen):	9 Monate

- **Live-System:** Viele Distributionen ermöglichen den Linux-Betrieb direkt von einer CD/DVD oder von einem USB-Stick. Das ermöglicht ein einfaches Ausprobieren von Linux. Außerdem bieten Live-Systeme eine ideale Möglichkeit, um ein defektes Linux-System zu reparieren.
- **Zielplattform (CPU-Architektur):** Viele Distributionen sind nur für Intel- und AMD-kompatible Prozessoren erhältlich, in der Regel in einer 32- und in einer 64-Bit-Variante. Es gibt aber auch Distributionen für andere Prozessorplattformen, z. B. für ARM- oder für PowerPC-CPU's.
- **Support:** Bei kommerziellen Distributionen bekommen Sie Hilfe bei der Installation (via E-Mail und/oder per Telefon).
- **Lizenz:** Die meisten Distributionen sind kostenlos erhältlich. Bei einigen Distributionen gibt es hier aber Einschränkungen: Beispielsweise ist bei den Enterprise-Distributionen von Red Hat und SUSE ein Zugriff auf das Update-System nur für registrierte Kunden möglich. Sie zahlen hier nicht für die Software an sich, sondern für das Service-Angebot rund herum.

So belebend die Konkurrenz vieler Distributionen für deren Weiterentwicklung ist, so lästig ist sie bei der Installation von Programmen, die nicht mit der Distribution mitgeliefert werden: Eine fehlende oder veraltete Programmbibliothek kann die Ursache dafür sein, dass ein Programm nicht läuft. Abhilfe versucht das Linux-Standard-Base-Projekt (LSB) zu schaffen: Die LSB-Spezifikation definiert Regeln, die einen gemeinsamen Nenner aller am LSB-Projekt beteiligten Distributionen sicherstellen:

Linux Standard Base (LSB)

<https://wiki.linuxfoundation.org/lsb/start>

Gängige Linux-Distributionen

Der folgende Überblick über die wichtigsten verfügbaren Distributionen soll Ihnen eine erste Orientierungshilfe geben. Die Liste ist alphabetisch geordnet und erhebt keinen Anspruch auf Vollständigkeit.

Android ist eine von Google entwickelte Plattform für Mobilfunkgeräte und Tablets. Android hat damit Linux zu der Weltdominanz verholfen, über die Linux-Entwickler in der Vergangenheit gescherzt haben. Aber Android ist natürlich keine typische, PC-taugliche Distribution.

Android

Arch Linux ist eine für technische Anwender optimierte Linux-Distribution. Die manuell im Textmodus durchzuführende Installation stellt sicher, dass Einsteiger einen großen Bogen um Arch Linux machen. Dafür zählen <https://wiki.archlinux.org> und <https://wiki.archlinux.de> zu den besten Quellen für Linux-Konfigurationsdetails im Netz. Arch-Linux-Derivate wie **Manjaro** und **Antergos** mit grafischen Installations-

Arch Linux

	und Konfigurationsprogrammen haben Arch-Linux zuletzt sogar in die Top-10-Liste von <i>distrowatch.com</i> gebracht.
CentOS und Scientific Linux	CentOS und Scientific Linux sind zwei kostenlose Varianten zu Red Hat Enterprise Linux (RHEL). Beide Distributionen sind binärkompatibel zu RHEL, es fehlen aber alle Red-Hat-Markenzeichen, -Logos etc. Die Distributionen sind vor allem für Server-Betreiber interessant, die kompatibel zu RHEL sein möchten, sich die hohen RHEL-Kosten aber nicht leisten können.
Chrome OS	Das Chrome OS wird wie Android von Google entwickelt. Es ist für Notebooks optimiert und setzt zur Nutzung eine aktive Internetverbindung voraus. Die Benutzeroberfläche basiert auf dem Google Chrome Webbrowser. Chrome OS spielt aktuell in Europa keine große Rolle, wohl aber auf dem Bildungsmarkt in den USA: Dort werden billige Chrome-Books (also Notebooks mit Chrome OS) häufig in Schulen eingesetzt.
Debian	Debian ist die älteste vollkommen freie Distribution. Sie wird von engagierten Linux-Entwicklern zusammengestellt, wobei die Einhaltung der Spielregeln »freier« Software eine hohe Priorität genießt. Die strikte Auslegung dieser Philosophie hat in der Vergangenheit mehrfach zu Verzögerungen geführt. Debian richtet sich an fortgeschrittene Linux-Anwender und hat einen großen Marktanteil bei Server-Installationen. Im Vergleich zu anderen Distributionen ist Debian stark auf maximale Stabilität hin optimiert und enthält deswegen oft relativ alte Programmversionen. Dafür steht Debian für viele Hardware-Plattformen zur Verfügung, unter anderem für AMD64, ARM64, ARMEL, ARMHF, i386, Mips, Mipsel, PowerPC, PPC64EL und S390X. Es gibt zahlreiche Distributionen, die sich von Debian ableiten, z. B. Raspbian und Ubuntu.
Fedora	Fedora ist der kostenlose Entwicklungszweig von Red Hat Linux. Die Entwicklung wird von Red Hat unterstützt und gelenkt. Für Red Hat ist Fedora eine Art Spielwiese, auf der neue Funktionen ausprobiert werden können, ohne die Stabilität der Enterprise-Versionen zu gefährden. Programme, die sich unter Fedora bewähren, werden später in die Enterprise-Versionen integriert. Bei technisch interessierten Linux-Fans ist Fedora beliebt, weil diese Distribution oft eine Vorreiterrolle spielt: Neue Linux-Funktionen finden sich oft zuerst in Fedora und erst später in anderen Distributionen. Neue Fedora-Versionen erscheinen alle sechs Monate. Updates werden einen Monat nach dem Erscheinen der übernächsten Version eingestellt, d. h., die Lebensdauer ist mit 13 Monaten sehr kurz.
Kali Linux	Das auf Debian basierende Kali Linux enthält eine riesige Sammlung von Hacking- und Pen-Testing-Werkzeugen. Die Distribution gilt als <i>der</i> Werkzeugkasten für Hacker und Sicherheits-Experten.

openSUSE ist eine kostenlose Linux-Distribution. Beginnend mit der seit November 2015 verfügbaren Version »Leap 42.n« basiert openSUSE auf den Enterprise-Versionen von SUSE, ersetzt aber viele Programme durch aktuellere Versionen. Voraussichtlich ab 2018 sollen die Versionsnummern von SUSE Enterprise und openSUSE Leap zusammengeführt werden. openSUSE wird dann einen Sprung zurück zu Version 16 machen.	openSUSE
Oracle bietet unter dem Namen Oracle Linux eine Variante zu Red Hat Enterprise Linux (RHEL) an. Das ist aufgrund der Open-Source-Lizenzen eine zulässige Vorgehensweise. Technisch gibt es nur wenige Unterschiede zu RHEL, die Oracle-Variante ist aber billiger und ohne Support sogar kostenlos verfügbar. Dennoch ist die Verbreitung von Oracles Linux-Variante verhältnismäßig gering.	Oracle
Raspbian ist die Standard-Distribution für den beliebten Minicomputer Raspberry Pi. Raspbian basiert auf Debian, wurde für den Raspberry Pi aber speziell adaptiert und erweitert.	Raspbian
Red Hat ist die international bekannteste und erfolgreichste Linux-Firma. Red-Hat-Distributionen dominieren insbesondere den amerikanischen Markt. Die Paketverwaltung auf der Basis des RPM-Formats (einer Eigenentwicklung von Red Hat) wurde von vielen anderen Distributionen übernommen. Red Hat ist überwiegend auf Unternehmenskunden ausgerichtet. Die Enterprise-Versionen (RHEL = Red Hat Enterprise Linux) sind vergleichsweise teuer. Sie zeichnen sich durch hohe Stabilität und einen zehnjährigen Update-Zeitraum aus. Für Linux-Enthusiasten und -Entwickler, die ein Red-Hat-ähnliches System zum Nulltarif suchen, bieten sich CentOS und Fedora an.	Red Hat
SUSE gilt weltweit als die Nummer zwei auf dem kommerziellen Linux-Markt. SUSE Enterprise ist vor allem im europäischen Markt verankert.	SUSE
Ubuntu ist die zurzeit populärste Distribution für Privatanwender. Ubuntu verwendet als Basis Debian, ist aber besser für Desktop-Anwender optimiert (Motto: <i>Linux for human beings</i>). Die kostenlose Distribution erscheint im Halbjahresrhythmus. Für gewöhnliche Versionen werden Updates über neun Monate zur Verfügung gestellt. Für die alle zwei Jahre erscheinenden LTS-Versionen gibt es sogar 3 bzw. 5 Jahre lang Updates (für Desktop- bzw. Server-Pakete). Finanziell wird Ubuntu Linux durch die Firma Canonical unterstützt.	Ubuntu
Zu Ubuntu gibt es eine Menge offizieller und inoffizieller Varianten. Etabliert und weit verbreitet sind Ubuntu Server , Kubuntu , Xubuntu , Ubuntu MATE und Linux Mint . Relativ neue Ubuntu-Derivate mit modernen Desktop-Systemen sind Budgie (Solus Desktop), elementary OS (macOS-ähnlicher Desktop), Neon und Zorin OS (Windows-ähnlicher Desktop). Besonders interessant ist Neon: Diese Distribution	Ubuntu-Derivate

Andere Distributionen

kombiniert Ubuntu LTS mit stets aktuellen KDE-Paketen und etabliert sich damit aktuell als *die* Distribution für KDE-Fans.

Neben den oben aufgezählten »großen« Distributionen gibt es im Internet zahlreiche Zusammenstellungen von Miniatursystemen. Sie sind vor allem für Spezialaufgaben konzipiert, etwa für Wartungsarbeiten (Emergency-Systeme) oder um ein Linux-System ohne eigentliche Installation verwenden zu können (Live-Systeme). Populäre Vertreter dieser Linux-Gattung sind **Devil Linux**, **Parted Magic** und **TinyCore**.

Einen ziemlich guten Überblick über alle momentan verfügbaren Linux-Distributionen, egal ob kommerziellen oder anderen Ursprungs, finden Sie im Internet auf der folgenden Seite:

<https://distrowatch.com>

Die Qual der Wahl

Eine Empfehlung für eine bestimmte Distribution ist schwierig. Für Linux-Einsteiger ist es zumeist von Vorteil, sich vorerst für eine weitverbreitete Distribution wie Debian, Fedora, openSUSE oder Ubuntu zu entscheiden. Eine gute Wahl ist auch Linux Mint. Zu diesen Distributionen sind sowohl im Internet als auch im Buch- und Zeitschriftenhandel viele Informationen verfügbar. Bei Problemen ist es vergleichsweise leicht, Hilfe zu finden.

Kommerzielle Linux-Anwender bzw. Server-Administratoren müssen sich entscheiden, ob sie bereit sind, für professionellen Support Geld auszugeben. In diesem Fall spricht wenig gegen die Marktführer Red Hat und SUSE. Andernfalls sind CentOS, Debian und Ubuntu attraktive kostenlose Alternativen.

1.4 Open-Source-Lizenzen (GPL & Co.)

Die Grundidee von »Open Source« besteht darin, dass der Quellcode von Programmen frei verfügbar ist und von jedem erweitert bzw. geändert werden darf. Allerdings ist damit auch eine Verpflichtung verbunden: Wer Open-Source-Code zur Entwicklung eigener Produkte verwendet, muss den gesamten Code ebenfalls wieder frei weitergeben.

Die Open-Source-Idee verbietet übrigens keinesfalls den Verkauf von Open-Source-Produkten. Auf den ersten Blick scheint das ein Widerspruch zu sein. Tatsächlich bezieht sich die Freiheit in »Open Source« mehr auf den Code als auf das fertige Produkt. Zudem regelt die freie Verfügbarkeit des Codes auch die Preisgestaltung von Open-Source-Produkten: Nur wer neben dem Kompatibel eines Open-Source-Programms weitere Zusatzleistungen anbietet (Handbücher, Support etc.), wird überleben. Sobald der Preis in keinem vernünftigen Verhältnis zu den Leistungen steht, werden sich andere Firmen finden, die es günstiger machen.

General Public License (GPL)

1

Das Ziel der Open-Source-Entwickler ist es, Software zu schaffen, deren Quellen frei verfügbar sind und es auch bleiben. Um einen Missbrauch auszuschließen, sind viele Open-Source-Programme durch die *GNU General Public License* (kurz GPL) geschützt. Hinter der GPL steht die *Free Software Foundation* (FSF). Diese Organisation wurde von Richard Stallman gegründet, um hochwertige Software frei verfügbar zu machen. Richard Stallman ist übrigens auch der Autor des Editors Emacs, der in Kapitel 16 beschrieben wird.

Die Kernaussage der GPL besteht darin, dass zwar jeder den Code verändern und sogar die resultierenden Programme verkaufen darf, dass aber gleichzeitig der Anwender/Käufer das Recht auf den vollständigen Code hat und diesen ebenfalls verändern und wieder kostenlos weitergeben darf. Jedes GNU-Programm muss zusammen mit dem vollständigen GPL-Text weitergegeben werden. Die GPL schließt damit aus, dass jemand ein GPL-Programm weiterentwickeln und verkaufen kann, *ohne* die Veränderungen öffentlich verfügbar zu machen. Jede Weiterentwicklung ist somit ein Gewinn für *alle* Anwender. Den vollständigen Text der GPL finden Sie hier:

<https://gnu.org/licenses/gpl.html>

Das Konzept der GPL ist recht einfach zu verstehen, im Detail treten aber immer wieder Fragen auf. Viele davon werden hier beantwortet:

<https://gnu.org/licenses/gpl-faq.html>

Wenn Sie glauben, dass Sie alles verstanden haben, sollten Sie das GPL-Quiz ausprobieren:

<https://gnu.org/cgi-bin/license-quiz.cgi>

Neben der GPL existiert noch die Variante LGPL (Lesser GPL). Der wesentliche Unterschied zur GPL besteht darin, dass eine derart geschützte Bibliothek auch von kommerziellen Produkten genutzt werden darf, deren Code *nicht* frei verfügbar ist. Ohne die LGPL könnten GPL-Bibliotheken nur wieder für GPL-Programme genutzt werden, was in vielen Fällen eine unerwünschte Einschränkung für kommerzielle Programmierer wäre.

Lesser General Public License (LGPL)

Durchaus nicht alle Teile einer Linux-Distribution unterliegen den gleichen Copyright-Bedingungen! Obwohl der Kernel und viele Tools der GPL unterliegen, gelten für manche Komponenten und Programme andere rechtliche Bedingungen:

Andere Lizenzen

- **MIT- und BSD-Lizenz:** Die MIT- und BSD-Lizenzen erlauben die kommerzielle Nutzung des Codes *ohne* die Verpflichtung, Änderungen öffentlich weiterzugeben. Die Lizenzen sind damit wesentlich liberaler als die GPL und eher mit der LGPL vergleichbar.

- **Doppellizenzen:** Für einige Programme gelten Doppellizenzen. Beispielsweise können Sie den Datenbank-Server MySQL für Open-Source-Projekte, auf einem eigenen Webserver bzw. für die innerbetriebliche Anwendung gemäß der GPL kostenlos einsetzen. Wenn Sie hingegen ein kommerzielles Produkt auf der Basis von MySQL entwickeln und samt MySQL verkaufen möchten, ohne Ihren Quellcode zur Verfügung zu stellen, dann kommt die kommerzielle Lizenz zum Einsatz. Die Weitergabe von MySQL wird in diesem Fall kostenpflichtig.
- **Kommerzielle Lizenzen:** Einige Programme unterstehen zwar einer kommerziellen Lizenz, dürfen aber dennoch kostenlos genutzt werden. Ein bekanntes Beispiel ist das Flash-Plugin von Adobe: Zwar ist das Programm unter Linux kostenlos erhältlich (und darf auch in Firmen kostenlos eingesetzt werden), aber der Quellcode zu diesem Programm ist nicht verfügbar.

Manche Distributionen kennzeichnen die Produkte, bei denen die Nutzung oder Weitergabe eventuell lizenzrechtliche Probleme verursachen könnte. Bei Debian befinden sich solche Programme in der Paketquelle *non-free*.

Das Dickicht der zahllosen, mehr oder weniger »freien« Lizenzen ist schwer zu durchschauen. Die Bandbreite zwischen der manchmal fundamentalistischen Auslegung von »frei« im Sinne der GPL und den verklausulierten Bestimmungen mancher Firmen, die ihr Software-Produkt zwar frei nennen möchten (weil dies gerade modern ist), in Wirklichkeit aber uneingeschränkte Kontrolle über den Code behalten möchten, ist groß.

Eine gute Einführung in das Thema geben die beiden folgenden Websites. Das Ziel von *opensource.org* ist es, unabhängig von Einzel- oder Firmeninteressen die Idee (oder das Ideal) von Software mit frei verfügbarem Quellcode zu fördern. Dort finden Sie auch eine Liste von Lizenzen, die der Open-Source-Idee entsprechen.

<https://heise.de/-221957>
<https://opensource.org>

Lizenzkonflikte zwischen Open- und Closed-Source-Software

Open-Source-Lizenzen für Entwickler

Wenn Sie Programme entwickeln und diese zusammen mit Linux bzw. in Kombination mit Open-Source-Programmen oder -Bibliotheken verkaufen möchten, müssen Sie sich in die bisweilen verwirrende Problematik der unterschiedlichen Software-Lizenzen tiefer einarbeiten. Viele Open-Source-Lizenzen erlauben die Weitergabe nur, wenn auch Sie Ihren Quellcode im Rahmen einer Open-Source-Lizenz frei verfügbar machen. Auf je mehr Open-Source-Komponenten mit unterschiedlichen Lizenzen Ihr Programm basiert, desto komplizierter wird die Weitergabe.

Es gibt aber auch Ausnahmen, die die kommerzielle Nutzung von Open-Source-Komponenten erleichtern: Beispielsweise gilt für Apache und PHP sinngemäß, dass Sie diese Programme auch in Kombination mit einem Closed-Source-Programm frei weitergeben dürfen.

Manche proprietäre Treiber für Hardware-Komponenten (z.B. für NVIDIA-Grafikkarten) bestehen aus einem kleinen Kernelmodul (Open Source) und diversen externen Programmen oder Bibliotheken, deren Quellcode nicht verfügbar ist (Closed Source). Das Kernelmodul hat nur den Zweck, eine Verbindung zwischen dem Kernel und dem Closed-Source-Treiber herzustellen.

GPL-Probleme mit Hardware-Treibern

Diese Treiber sind aus Sicht vieler Linux-Anwender eine gute Sache: Sie sind kostenlos verfügbar und ermöglichen es, diverse Hardware-Komponenten zu nutzen, zu denen es entweder gar keine oder zumindest keine vollständigen Open-Source-Treiber für Linux gibt. Die Frage ist aber, ob bzw. in welchem Ausmaß die Closed-Source-Treiber wegen der engen Verzahnung mit dem Kernel, der ja der GPL untersteht, diese Lizenz verletzen. Viele Open-Source-Entwickler dulden die Treiber nur widerwillig. Eine direkte Weitergabe mit GPL-Produkten ist nicht zulässig, weswegen der Benutzer die Treiber in der Regel selbst herunterladen und installieren muss.

1.5 Die Geschichte von Linux

Da Linux ein Unix-ähnliches Betriebssystem ist, müsste ich an dieser Stelle eigentlich mit der Geschichte von Unix beginnen – aber dazu fehlt hier der Platz. Stattdessen beginnt diese Geschichtsstunde mit der Gründung des GNU-Projekts durch Richard Stallman. GNU steht für *GNU is not Unix*. In diesem Projekt wurden seit 1982 Open-Source-Werkzeuge entwickelt. Dazu zählen der GNU-C-Compiler, der Texteditor Emacs sowie diverse GNU-Utilities wie *find* und *grep* etc.

1982: GNU

Erst sieben Jahre nach dem Start des GNU-Projekts war die Zeit reif für die erste Version der *General Public License*. Diese Lizenz stellt sicher, dass freier Code frei bleibt.

1989: GPL

Die allerersten Teile des Linux-Kernels (Version 0.01) entwickelte Linus Torvalds. Er gab seinen Code im September 1991 über das Internet frei. Schnell fanden sich weltweit Programmierer, die an der Idee Interesse hatten und Erweiterungen dazu programmierten. Als der Kernel von Linux die Ausführung des GNU-C-Compiler erlaubte, stand auch die gesamte Palette der GNU-Tools zur Verfügung. Weitere Komponenten waren das Dateisystem Minix, Netzwerk-Software von BSD-Unix, das X Window System des MIT und dessen Portierung XFree86 etc.

1991: Linux-Kernel 0.01

	Linux ist also nicht nur Linus Torvalds zu verdanken. Hinter Linux stehen vielmehr eine Menge engagierter Menschen, die in ihrer Freizeit, im Rahmen ihres Studiums oder bezahlt von Firmen wie Google, IBM oder HP freie Software produzieren.
1994: Erste Distributionen	Informatik-Freaks an Universitäten konnten sich Linux und seine Komponenten selbst herunterladen, kompilieren und installieren. Eine breite Anwendung fand Linux aber erst mit Linux-Distributionen, die Linux und die darum entstandene Software auf Disketten bzw. CD-ROMs verpackten und mit einem Installationsprogramm versahen. Vier der zu dieser Zeit entstandenen Distributionen existieren heute noch: Debian, Red Hat, Slackware und SUSE.
1996: Pinguin	1996 wurde der Pinguin zum Linux-Logo.
1998: Microsoft nimmt Linux wahr	Mit dem rasanten Siegeszug des Internets stieg auch die Verbreitung von Linux, vor allem auf Servern. Gewissermaßen zum Ritterschlag für Linux wurde der legendäre Ausspruch von Steve Ballmer: <i>Microsoft is worried about free software</i> ... Ein Jahr später ging Red Hat spektakulär an die Börse.
2009: Android	Mit der Android-Plattform brachte Google Linux zuerst auf das Handy (2009), danach auch auf Tablets und in TV-Geräte.
2012: Raspberry Pi	2012 eroberte der Minicomputer Raspberry Pi die Herzen von Elektronikbastlern. Für nur rund 40 EUR können Sie mit dem Raspberry Pi selbst Hardware-Experimente durchführen, in die Welt der Heimautomation einsteigen, ein Medien-Center oder einen Home-Server betreiben. Der Raspberry Pi macht Embedded Linux zu einem Massenphänomen.

1.6 Software-Patente und andere Ärgernisse

Patente schützen in den USA und anderen Ländern Software-Ideen, -Konzepte und Algorithmen. Alles Mögliche und Unmögliche ist patentiert, triviale Dinge wie die Darstellung eines Fortschrittsbalkens oder die 1-Click-Bestellung (Amazon). Der Missbrauch derartiger Trivialpatente und die für die schnelllebige Software-Branche sehr langen Laufzeiten von 20 Jahren tragen zum Widerwillen gegen Software-Patente bei.

Beispielsweise verzichteten viele Distributionen jahrelang aus Angst vor Klagen darauf, Bibliotheken zum Abspielen von MP3-Dateien mitzuliefern; die darin eingesetzten Algorithmen sind durch Patente geschützt. Die Anwender mussten sich die zum Abspielen von MP3-Dateien erforderlichen Bibliotheken selbst installieren. Glücklicherweise laufen die MP3-Patente 2017 aus, sodass zumindest dieses Problem jetzt aus der Welt geschafft ist.

Während Patente selten ein Risiko für einzelne Software-Entwickler sind, spielen sie im Kampf um Marktanteile eine immer größere Rolle, besonders im heiß umkämpften Smartphone- und Tablet-Markt. Jeder große Hersteller verklagt jeden anderen – mit ungewissem Ausgang, aber auf jeden Fall zur Freude der beteiligten Rechtsanwälte und Kanzleien.

Ganz aussichtslos ist die Lage zum Glück nicht. Das liegt vor allem daran, dass einige Linux nahestehende Firmen wie IBM selbst über riesige Patent-Pools verfügen. Diverse Linux-Firmen haben zudem begonnen, selbst Patente zu sammeln, die teilweise von anderen Firmen gleichsam für Open-Source-Zwecke »gespendet« wurden. Das Absurde an der Situation besteht darin, dass ein verfehltes Patentrecht die Open-Source-Gemeinde dazu zwingt, selbst Patente einzusetzen, um sich gegen eventuelle Klagen zu schützen. Details über Patent-Pools der Open-Source-Gemeinde finden Sie hier:

<https://openinventionnetwork.com>

Auch abseits der MP3-Dateien ist der Multimedia-Markt ein Problemfeld. Beispielsweise können Sie unter Linux DVDs nicht ohne Weiteres abspielen. Diverse Gesetze verbieten in vielen Ländern sowohl die Weitergabe der erforderlichen Bibliotheken als auch die bloße Beschreibung, wie diese zu installieren sind – z. B. das Urheberrechtsgesetz in Deutschland.

Nicht besser sieht es mit online erworbenen Daten (Videos, E-Books etc.) aus, die durch DRM geschützt sind. DRM steht für *Digital Rights Management* und bezeichnet diverse Verfahren, um die Nutzung der Daten so einzuschränken, dass sie nur auf einem ganz bestimmten Rechner möglich ist. Sozusagen nebenbei werden Sie dadurch auf eine bestimmte Hardware (z. B. iPod oder iPhone) bzw. auf ein bestimmtes Betriebssystem (z. B. Windows, macOS) beschränkt. DRM-Gegner bezeichnen das System nicht umsonst als *Digital Restriction Management*.

Patent-Pools der Open-Source-Gemeinde

Multimedia

Digital Rights Management

Kapitel 14

Netzwerk-Tools

Dieses Kapitel stellt Kommandos zur Benutzung, Steuerung und Analyse elementarer Netzwerkdienste vor. Sie lernen hier, wie Sie sich mit `ssh` auf einem anderen Rechner im Netzwerk einloggen und mit `wget` Dateien übertragen. Mit den Programmen `Lynx` und `Mutt` können Sie im Textmodus sogar Webseiten besuchen und Mails lesen und verfassen.

Weitere Kommandos zur Analyse des Netzwerkstatus sowie zur Suche nach offenen Ports auf fremden Rechnern, `netstat` und `nmap`, stelle ich Ihnen in Kapitel 37, »Firewalls«, vor.

14.1 Netzwerkstatus ermitteln

Dieser Abschnitt gibt einen Überblick über Kommandos zum Test der Grundfunktionen des Netzwerks. Weitere Informationen zu den hier vorgestellten Kommandos folgen in Abschnitt 26.4, »Manuelle LAN- und WLAN-Konfiguration«.

Das Kommando `ip addr` liefert eine Liste aller bekannten Netzwerkschnittstellen:

```
root# ip addr
ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc ...
    link/ether 00:1c:42:55:4f:0e brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.42/24 brd 10.0.0.255 scope global eth0
    inet6 fe80::21c:42ff:fe55:4f0e/64 scope link
        valid_lft forever preferred_lft forever
```

Netzwerk-
schnittstellen
ermitteln

Typische Schnittstellen sind `ethN` oder `enpNsM` (Ethernet), `wlanN` (WLAN) und `pppN` (Internetzugang via UMTS-Modem, ADSL oder VPN). Bei den meisten gängigen Distributionen fließt in den Namen der Ethernet-Schnittstelle die interne Bus-Nummer ein – z. B. `enp0s1`. Auf PCs und Notebooks mit nur einer Ethernet-Schnittstelle wirkt der

Name umständlich. Aber die bus-spezifische Nummerierung stellt sicher, dass sich auf Servern mit vielen Netzwerkschnittstellen die Nummerierung auch dann nicht ändert, wenn weitere Netzwerkadapter hinzugefügt werden.

Eine Sonderrolle nimmt die Schnittstelle `lo` ein: Sie ermöglicht es lokalen Programmen, über das Netzwerkprotokoll zu kommunizieren. Das funktioniert selbst dann, wenn ein Rechner nicht nach außen hin mit einem Netzwerk verbunden ist.

Wenn `ip addr` nur bei der Schnittstelle `lo` eine IP-Adresse angibt, wurde noch keine Netzwerkschnittstelle aktiviert. Abhilfe schafft das von Ihrer Distribution vorgesehene Werkzeug zur Netzwerkkonfiguration. Sie können die Netzwerkschnittstelle mit dem `ip`-Kommando auch manuell aktivieren. Details dazu sowie zur IPv6-Konfiguration finden Sie in Kapitel 26, »Netzwerkkonfiguration«.

Erreichbarkeit von localhost testen

`ping` sendet einmal pro Sekunde ein kleines Netzwerkpaket an die angegebene Adresse. Wenn sich dort ein Rechner befindet, sendet dieser eine Antwort, es sei denn, eine Firewall verhindert das. `ping` läuft so lange, bis es mit `[Strg]+[C]` beendet wird. `ping localhost` überprüft, ob das Loopback-Interface und damit die elementaren Netzwerkfunktionen des eigenen Rechners funktionieren:

```
user$ ping localhost
PING localhost (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=0.152 ms
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=0.114 ms
...
```

Erreichbarkeit des lokalen Netzes testen

Indem Sie an `ping` statt `localhost` die IP-Nummer eines anderen Rechners im lokalen Netz übergeben, testen Sie, ob das lokale Netz funktioniert. `-c 2` bewirkt, dass `ping` nicht endlos läuft, sondern nach zwei Paketen endet:

```
user$ ping -c 2 192.168.0.99
PING 192.168.0.99 (192.168.0.99): 56 data bytes
64 bytes from 192.168.0.99: icmp_seq=0 ttl=255 time=0.274 ms
64 bytes from 192.168.0.99: icmp_seq=1 ttl=255 time=0.150 ms
...
```

Wenn es im lokalen Netz einen Nameserver gibt, der der IP-Nummer `192.168.0.99` einen Namen zuordnet, oder wenn die Datei `/etc/hosts` diese Aufgabe übernimmt, können Sie bei `ping` statt der IP-Nummer den Rechnernamen angeben:

```
user$ ping -c 2 mars
PING mars.sol (192.168.0.99) 56(84) bytes of data.
64 bytes from mars.sol (192.168.0.99): icmp_seq=1 ttl=64 time=0.281 ms
64 bytes from mars.sol (192.168.0.99): icmp_seq=2 ttl=64 time=0.287 ms

--- mars.sol ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.281/0.284/0.287/0.003 ms
```

Als Nächstes können Sie testen, ob die Verbindung zum Internet gelingt. Das folgende Kommando testet gleichzeitig zwei Aspekte der Netzwerkkonfiguration: die Erreichbarkeit des Nameservers und die Funktion des Gateways.

Internetzugang testen

```
user$ ping -c 2 www.yahoo.com
PING www.yahoo-ht2.akadns.net (209.73.186.238) 56(84) bytes of data.
64 bytes from f1.www.vip.re3.yahoo.com (209.73.186.238): icmp_seq=1 time=122 ms
64 bytes from f1.www.vip.re3.yahoo.com (209.73.186.238): icmp_seq=2 time=123 ms

--- www.yahoo-ht2.akadns.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 122.731/123.256/123.782/0.631 ms
```

Wenn das nicht funktioniert, sind mehrere Ursachen denkbar:

- ▶ Vielleicht ist der Server von Yahoo gerade unerreichbar, oder der Server hat aus Sicherheitsgründen die Antwort auf `ping` deaktiviert. Probieren Sie eine andere bekannte Internetadresse aus.
- ▶ Für die Ermittlung der IP-Adresse zu `yahoo.com` ist der Nameserver verantwortlich. Wenn Sie die Fehlermeldung *unknown host yahoo.com* erhalten, gibt es Probleme mit dem Nameserver. Überprüfen Sie, ob `/etc/resolv.conf` dessen Adresse enthält.
- ▶ Das Gateway ist dafür zuständig, IP-Pakete aus dem lokalen Netzwerk an das Internet weiterzuleiten. Wenn das nicht funktioniert, erhalten Sie die Fehlermeldung *connect: Network is unreachable*. Die Gateway-Konfiguration können Sie mit `ip route` überprüfen. Das Kommando liefert normalerweise mehrere Zeilen. Die Gateway-Adresse befindet sich in der dritten Spalte der Zeile, die mit `default` beginnt:

```
user$ ip route
default via 10.0.0.138 dev eth0
10.0.0.0/24 dev eth0 proto kernel scope link src 10.0.0.42
```
- ▶ Falls Sie in einem lokalen Netz einen eigenen Rechner als Gateway eingerichtet haben, besteht die Möglichkeit, dass Sie die Masquerading-Funktion vergessen haben. In diesem Fall würde der Internetzugang für das gesamte lokale Netzwerk nicht funktionieren. Eine detaillierte Anleitung zur Konfiguration eines eigenen LAN- oder WLAN-Routers finden Sie in Kapitel 27, »Internet-Gateway«.

Mit `traceroute` finden Sie heraus, welchen Weg ein Netzwerkpaket von Ihrem Rechner zu einem anderen Rechner nimmt und wie viele Millisekunden die Laufzeit bis zur jeweiligen Zwischenstation beträgt. Standardmäßig unternimmt das Kommando drei Versuche und liefert daher entsprechend drei Zeiten. Das Kommando funktioniert nicht, wenn sich auf einer der Zwischenstationen eine Firewall befindet, die den von `traceroute` genutzten UDP-Port 33434 blockiert. In diesem Fall liefert `traceroute` für diese und alle weiteren Stationen nur noch drei Sterne.

Den Weg von IP-Paketen verfolgen

Die folgenden Zeilen zeigen den Weg von meinem Arbeitsrechner zu google.at. Zeile 1 beschreibt mein Internet-Gateway (den Rechner mars.sol), Zeile 2 den ADSL-Router und Zeile 3 das Gateway meines Internet-Providers.

```
user$ traceroute google.at
traceroute to google.at (66.102.9.104), 30 hops max, 40 byte packets
 1 mars.sol.0.168.192.in-addr.arpa (192.168.0.1) 0.277 ms ...
 2 192.168.1.1 (192.168.1.1) 0.373 ms ...
 3 N704P030.adsl.highway.telekom.at (62.47.31.254) 8.598 ms ...
 4 172.19.90.193 (172.19.90.193) 11.864 ms ...
 ...
14 66.102.9.104 (66.102.9.104) 52.741 ms ...
```

Firewalls umgehen

Mitunter behindern Sicherheitseinstellungen und Firewalls die Arbeit von traceroute. Anstelle von IP-Adressen zeigt das Kommando dann nur * * * an. In solchen Fällen können Sie versuchen, mit den Optionen -T oder -I andere Verfahren zu verwenden, um den Weg von Paketen zu verfolgen. Beide Optionen erfordern root-Rechte.

mtr Das Kommando `mtr` sendet regelmäßig Netzwerkpakete zum angegebenen Host und analysiert die Antworten. Die Ergebnisliste kombiniert Daten von ping und traceroute. Beachten Sie, dass es zwei Versionen dieses Programms gibt: das hier beschriebene Textkommando sowie eine Variante mit grafischer Benutzeroberfläche. Bei Desktop-Installationen von Debian und Ubuntu ist standardmäßig die GTK-Variante installiert. Um stattdessen die Textversion zu installieren, führen Sie `apt-get install mtr-tiny` aus.

```
user$ mtr -c 10 -r google.de
HOST: michael's-computer      Loss%  Snt  Last  Avg  Best  Wrst StDev
 1 |-- speedtouch.lan         0.0%   10   42.6  48.5   6.0  95.9  28.9
 2 |-- 178-191-207-254.adsl.hi 0.0%   10   18.9  20.4  18.6  23.2   1.9
 3 |-- 195.3.74.129           0.0%   10   19.4  18.8  17.9  19.4   0.5
 4 |-- AUX10-GRAZBC10.highway.te 0.0%   10   21.2  21.3  20.7  22.0   0.3
 5 |-- 195.3.70.154           0.0%   10   21.2  27.3  20.9  81.2  18.9
 6 |-- 62.47.120.150          0.0%   10   25.3  25.5  24.9  26.0   0.4
 7 |-- 209.85.243.119         0.0%   10   25.6  25.9  25.2  28.2   0.8
 8 |-- 216.239.46.88          0.0%   10   25.8  26.3  25.8  27.7   0.6
 9 |-- bud01s08-in-f23.1e100.net 0.0%   10   25.7  25.8  25.0  26.8   0.5
```

gnome-nettool Wer unter Gnome arbeitet, kann einen Großteil der oben aufgezählten Informationen ganz komfortabel mit dem Programm `gnome-nettool` ermitteln (siehe Abbildung 14.1). Bei einigen Distributionen steht das Programm aus Platzgründen standardmäßig nicht zur Verfügung – dann müssen Sie das gleichnamige Paket zuerst installieren.

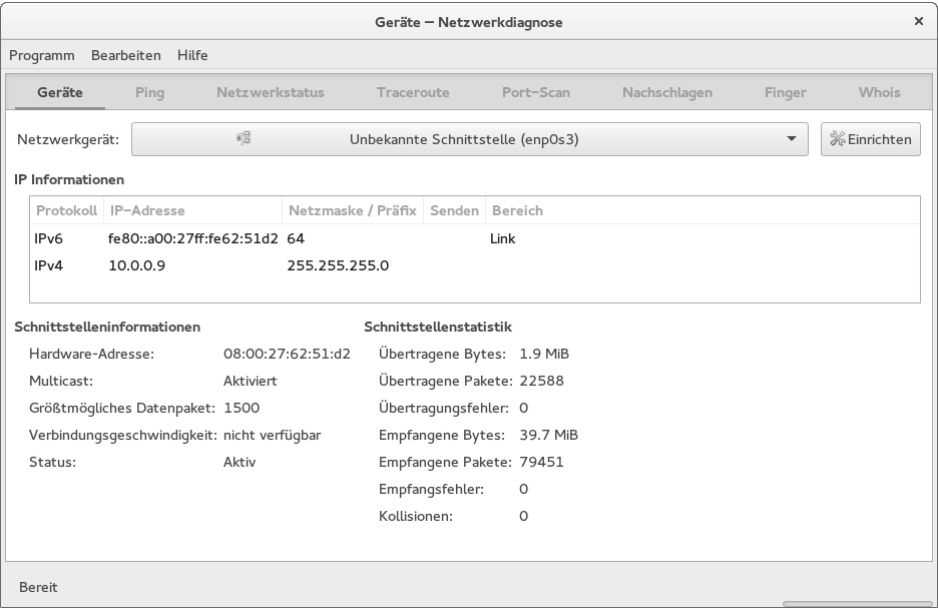


Abbildung 14.1 Netzwerkdiagnose unter Gnome

14.2 Auf anderen Rechnern arbeiten (SSH)

Die Programme `telnet`, `rlogin` und `ssh` ermöglichen es, so auf einem anderen Rechner zu arbeiten, als stünde er vor Ihnen. Das funktioniert sowohl für kommandoorientierte Programme als auch für X-Programme. Dieser Abschnitt beschränkt sich auf die Beschreibung von `ssh` (Secure Shell). Die älteren Programme `telnet` und `rlogin` sollten aus Sicherheitsgründen nicht mehr eingesetzt werden. Sie übertragen die Login-Informationen inklusive des Passworts unverschlüsselt.

Die Grundvoraussetzung für die Anwendung von `ssh` besteht darin, dass auf dem zweiten Rechner ein SSH-Server läuft, also das Programm `sshd`. Bei manchen Linux-Distributionen ist dies standardmäßig der Fall, bei anderen muss das Programm (zumeist als Paket `openssh-server`) zuerst installiert werden. Wenn auf den Rechnern Firewalls laufen, dürfen diese den Port 22 nicht blockieren.

Einen eigenen SSH-Server einrichten

Informationen zur Installation, Konfiguration und Absicherung eines SSH-Servers folgen in Kapitel 31, »Secure Shell (SSH)«. Dort erfahren Sie auch, wie Sie den SSH-Server absichern.

Gewöhnliche Shell-Session Wenn Sie auf dem Rechner `uranus` arbeiten und nun eine Shell-Session auf dem Rechner `mars` starten möchten, führen Sie zum Verbindungsaufbau das folgende Kommando aus:

```
user@uranus$ ssh mars
user@mars's password: *****
```

Beim ersten Verbindungsaufbau zu einem neuen Rechner erscheint eine Warnung nach dem folgenden Muster:

```
The authenticity of host 'mars (192.168.0.10)' can't be established.
RSA1 key fingerprint is 1e:0e:15:ad:6f:64:88:60:ec:21:f1:4b:b7:68:f4:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'mars,192.168.0.10' (RSA1) to the list
of known hosts.
```

Das bedeutet, dass `ssh` sich nicht sicher ist, ob es dem Rechner `mars` mit der IP-Adresse `192.168.0.10` vertrauen darf. Es könnte sein, dass ein fremder Rechner vortäuscht, `mars` zu sein. Wenn Sie die Rückfrage mit `yes` beantworten, speichert `ssh` den Namen, die Adresse und den RSA-Fingerprint (einen Code zur eindeutigen Identifizierung des Partnerrechners) in `~/.ssh/known_hosts`.

Falls Sie auf `mars` unter einem anderen Login-Namen als auf `uranus` arbeiten möchten (z. B. als `root`), geben Sie den Namen mit der Option `-l` an:

```
user@uranus$ ssh -l root mars
root@mars's password: *****
```

SSH-Authentifizierung mit Schlüsseln

Wesentlich sicherer als ein Login mit Passwort ist die Authentifizierung durch einen Schlüssel. Die Vorgehensweise wird im Detail in Abschnitt 31.4 beschrieben. Die Verwendung von Schlüsseln ermöglicht es auch, auf SSH basierende Kommandos und Scripts automatisch per Script auszuführen.

Kommandos ausführen Statt `ssh` interaktiv zu nutzen, können Sie auf dem entfernten Rechner auch einfach nur ein Kommando ausführen. Das Kommando und seine Parameter werden einfach als weitere Parameter an `ssh` übergeben. `ssh` endet nach diesem Kommando.

```
user@uranus$ ssh mars kommando optionen
user@mars's password: *****
```

Aus dieser scheinbar trivialen Funktion ergeben sich weitreichende Möglichkeiten: Sie können nun beispielsweise auf dem entfernten Rechner `tar` starten, das damit erstellte Archiv an die Standardausgabe weiterleiten (geben Sie dazu einen Bindestrich - nach der Option `-f` ein, also `-f -`) und die Standardausgabe mit `|` als Eingabe

für ein zweites `tar`-Kommando verwenden, das lokal läuft. Damit können Sie einen ganzen Verzeichnisbaum sicher via SSH kopieren.

Das folgende Kommando zeigt, wie ich den gesamten `/var/www`-Verzeichnisbaum meines Webserver `kofler.info` in das lokale Verzeichnis `~/bak` kopiere. Das Kommando setzt dabei voraus, dass alle Dateien in `/var/www` vom Benutzer `username` gelesen werden können.

```
user$ ssh -l username kofler.info tar -cf - /var/www | tar -xC ~/bak/ -f -
username@kofler.info's password: *****
```

Wenn Sie in einem Script mehrere Kommandos via SSH ausführen möchten, verwenden Sie am besten die Heredoc-Syntax (siehe auch Abschnitt 10.11, »Code-Strukturierung in bash-Scripts«):

```
#!/bin/bash
pw=strengeheim
...
ssh -T root@host <<ENDSSH
echo root:$pw | chpasswd
rm -f /etc/file1
cp /root/file2 /userxy/file3
ENDSSH
...
```

Damit führt `ssh` alle Kommandos aus, bis im Script die mit `ENDSSH` markierte Zeile erreicht wird. Die Option `-T` verhindert dabei, dass SSH versucht, ein Pseudo-Terminal zu öffnen. Das ist hier unerwünscht, weil die Kommandoausführung nicht interaktiv erfolgen soll.

Beim ersten SSH-Verbindungsaufbau zu einem neuen Host fragt `ssh`, ob Sie dem Host vertrauen. Normalerweise ist diese Rückfrage sinnvoll. Wenn Sie aber mit `ssh` automatisiert auf mehreren Hosts (oft in virtuellen Maschinen) Arbeiten durchführen möchten, stört die Rückfrage. Abhilfe schafft in solchen Fällen die Option `-o StrictHostKeyChecking=no`.

Diese und andere Optionen können Sie auch global in `/etc/ssh/ssh_config` oder individuell für einen Benutzer in `.ssh/config` einstellen. Verwechseln Sie aber `/etc/ssh/ssh_config` nicht mit `/etc/ssh/sshd_config`! Die erste Datei enthält SSH-Client-Optionen, die zweite Datei Optionen für den SSH-Server.

Sofern als Grafiksystem `X` und nicht `Wayland` verwendet wird (sowohl auf dem Client als auch auf dem Server!), können Sie in einer SSH-Verbindung, die Sie mit `ssh -X` initiiert haben, auch Grafikprogramme ausführen. Die Option `-X` ist erforderlich, damit sich `ssh` um die korrekte Einstellung der `DISPLAY`-Variablen kümmert.

```
user@localhost$ ssh -X otheruser@otherhost
otheruser@otherhost$ firefox & (Firefox läuft extern, wird aber lokal angezeigt)
```

Mehrere Kommandos ausführen (Scripts)

Rückfrage bei erstem Verbindungsaufbau verhindern

SSH und X

Dateien sicher kopieren mit scp Um eine Datei via SSH über das Netzwerk zu kopieren, gibt es das Kommando `scp`. Die Syntax sieht so aus:

```
user$  scp [[user1@]host1:]filename1 [[user2@]host2:][filename2]
user2@host2's password:  *****
```

Damit wird die Datei `filename1` vom Rechner `host1` zum Rechner `host2` übertragen und dort in der Datei `filename2` gespeichert. Einige Anmerkungen zu den vielen optionalen Bestandteilen der Kopieranweisung:

- ▶ `host1` und `host2` müssen nicht angegeben werden, wenn der lokale Rechner (also `localhost`) gemeint ist.
- ▶ `user1` muss nicht angegeben werden, wenn der aktive Benutzer gemeint ist.
- ▶ `user2` muss nicht angegeben werden, wenn auf dem Rechner `host2` der aktuelle Benutzername von `host1` bzw. `user1` verwendet werden soll.
- ▶ `filename1` darf auch ein Verzeichnis sein. Sie müssen dann die Option `-r` angeben, damit das gesamte Verzeichnis mit allen Unterverzeichnissen übertragen wird.
- ▶ `filename2` muss nicht angegeben werden, wenn der Dateiname unverändert bleiben soll. Die Datei wird dann in das Home-Verzeichnis von `user2` kopiert.
Statt `filename2` kann auch das Zielverzeichnis angegeben werden, wobei wie üblich `~` für das Home-Verzeichnis von `user2` verwendet wird.

Zum Abschluss noch ein Beispiel: Nehmen Sie an, die Benutzerin `gabi` arbeitet auf dem Rechner `uranus`. Sie will die Datei `abc.txt` in das Verzeichnis `~/efg` auf dem Rechner `mars` übertragen. Das `scp`-Kommando sieht so aus:

```
gabi@uranus$  scp abc.txt mars:~/efg/
gabi@mars's password:  *****
```

Falls Sie beim `scp`-Kommando eine IPv6-Adresse angeben wollen, müssen Sie diese in eckige Klammern stellen. Andernfalls kommt `scp` bei den vielen Doppelpunkten durcheinander.

```
user$  scp kofler@[2001:1234:5678::1]:datei.txt .
```

SFTP SFTP (*Secure FTP*) ist eine auf SSH basierende sichere Variante zum Protokoll FTP. Details zu SFTP folgen im nächsten Abschnitt, der die Übertragung von Dateien via FTP und HTTP zum Thema hat.

SSH-Tunnel Eine SSH-Anwendungsmöglichkeit für fortgeschrittene Linux-Anwender ist der Tunnelbau. Derartige Tunnel eignen sich zwar nicht als Transportmöglichkeit für Autos oder Züge, sie ermöglichen aber die Übertragung aller IP-Pakete, die an einen bestimmten Port gerichtet sind. SSH-Tunnel bieten damit einen sicheren Weg, um IP-Pakete zwischen zwei Rechnern zu übertragen – und das selbst dann, wenn sich

zwischen den beiden Rechnern eine Firewall befindet, die den Port eigentlich blockiert. Eine Einführung in die Welt der IP-Pakete und eine Erklärung des Begriffs *Port* finden Sie in Kapitel 37, »Firewalls«.

Wenn der Tunnelbau vom Client-Rechner aus erfolgt, kommt die Option `-L localhost:localhost:remoteport` zum Einsatz. Beispielsweise bewirkt das folgende Kommando, dass der Port 3306 des Rechners `mars` über den Port 3307 des lokalen Rechners zugänglich ist. Durch das Kommando wird gleichzeitig eine SSH-Session gestartet, was Sie durch `-N` aber verhindern können (wenn Sie nur den Tunnel, aber keine Shell benötigen). Falls der Login bei `mars` unter einem anderen Namen erfolgen soll, müssen Sie den Login-Namen wie üblich durch `-l name` oder durch `name@remotehost` angeben.

```
user@uranus$  ssh -L 3307:localhost:3306 username@mars
user@mars's password:  *****
```

Der Tunnel bleibt so lange offen, bis die SSH-Session mit `[Strg]+[D]` beendet wird. Falls Sie `ssh` mit der Option `-N` gestartet haben, muss das Programm mit `[Strg]+[C]` gestoppt werden.

3306 ist der übliche Port von MySQL. Sie können nun auf dem Rechner `uranus` über dessen Port 3307 auf den MySQL-Server zugreifen, der auf `mars` läuft. Beim `mysql`-Kommando müssen Sie den Port 3307 und den Hostname `127.0.0.1` angeben, damit der SSH-Tunnel tatsächlich benutzt wird. Standardmäßig stellt `mysql` lokale Verbindungen über eine Socket-Datei her.

```
user@uranus$  mysql -u mysqllogin -P 3307 -h 127.0.0.1 -p
Enter password:  *****
```

Damit der MySQL-Login funktioniert, müssen zwei Voraussetzungen erfüllt sein:

- ▶ Erstens muss der MySQL-Server auf dem Rechner `mars` grundsätzlich IP-Verbindungen akzeptieren. Der MySQL-Server kann aus Sicherheitsgründen auch so konfiguriert sein, dass Verbindungen nur über eine Socket-Datei möglich sind. Dann hilft ein Tunnel nicht weiter, weil ein Tunnel nur Ports verbinden kann.
- ▶ Zweitens muss der MySQL-Server die Kombination aus Login-Name und Hostname akzeptieren. Als Hostname wird der Name des Rechners verwendet, zu dem `ssh` den Tunnel errichtet hat – hier also `mars` bzw. `mars.sol`, wenn die Domain `sol` lautet.

Es gibt noch weit mehr und oft viel komplexere Anwendungsmöglichkeiten für SSH-Tunnel. Beispielsweise können Sie die Tunnel dazu verwenden, um ein Virtual Private Network zu bilden. Weiterführende Dokumentation finden Sie z. B. hier:

<http://www.tldp.org/HOWTO/VPN-HOWTO>

SSH-Dateisystem Mit dem Kommando `sshfs`, das sich bei vielen Distributionen im gleichnamigen Paket befindet, können Sie das Dateisystem eines externen Rechners in den lokalen Verzeichnisbaum integrieren. Das kann beispielsweise die Durchführung von Backups vereinfachen.

```
root# mkdir /media/ext-host
root# sshfs user@hostname /media/ext-host
root# ...
root# umount /media/ext-host
```

Beachten Sie aber, dass Sie im SSH-Dateisystem wegen der Verschlüsselung aller Daten zumeist einen geringeren Durchsatz als mit Samba oder NFS erzielen werden. Das SSH-Dateisystem ist deswegen für den Einsatz in lokalen Netzwerken nur bedingt geeignet. Ich habe zudem die Erfahrung gemacht, dass `sshfs` auf kurzzeitige Netzwerk-ausfälle allergisch reagiert und dann hängen bleibt. Ich bin deswegen vom Einsatz dieses an sich praktischen Dateisystems wieder abgekommen.

telnet

Ein Vorgänger von `ssh` war `telnet`. Da `telnet` keine Daten verschlüsselt, sollte das Kommando auf keinen Fall dazu verwendet werden, um auf externen Rechnern zu arbeiten. Aktuelle Linux-Distributionen lassen dies standardmäßig ohnedies nicht zu, aber man stößt immer wieder auf Router, ADSL-Modems etc., die diese Art der Kommunikation zulassen.

Der Grund, warum ich Ihnen hier `telnet` überhaupt präsentiere, ist ein anderer: `telnet` eignet sich gut dazu, um zu überprüfen, ob auf einem externen Rechner auf einem bestimmten Port ein Netzwerkdienst läuft und auf einen Verbindungsaufbau wartet. Beispielsweise können Sie mit `telnet` sicherstellen, dass der zuvor eingerichtete Mail-Server tatsächlich läuft. Dazu übergeben Sie an `telnet` den Namen oder die IP-Adresse des Servers sowie die Port-Nummer:

```
user$ telnet kofler.info 25
Trying 5.9.22.29...
Connected to kofler.info.
Escape character is '^]'.
220 kofler.info ESMTP Postfix (Ubuntu)
helo kofler.info
250 kofler.info
^]      (Verbindung mit Strg+] beenden)
```

14.3 Dateien übertragen (FTP)

FTP steht für *File Transfer Protocol* und bezeichnet ein recht altes Verfahren zur Übertragung von Dateien über ein Netzwerk. Seine große Popularität verdankt FTP der Spielart Anonymous FTP: Viele große Internet-Server bieten allen Anwendern Zugang zu sogenannten FTP-Archiven. Dieser Zugang ist (im Gegensatz zum sonstigen FTP) nicht durch ein Passwort versperrt.

Grundlagen

Ein großer Nachteil von FTP besteht darin, dass beim Login-Prozess der Benutzername und das Passwort unverschlüsselt übertragen werden. Eine sichere Alternative ist SFTP (Secure FTP) auf der Basis von SSH (siehe Kapitel 31, »Secure Shell (SSH)«). Auch HTTP, also das Protokoll zur Übertragung von Webseiten, wird oft als Alternative zu FTP eingesetzt.

In diesem Kapitel geht es nur um die Nutzung von FTP, also um die Client-Sichtweise. Damit FTP funktioniert, muss auf der Gegenstelle ein FTP-Server laufen. Dessen Konfiguration ist in Abschnitt 32.8, »FTP-Server (vsftpd)«, beschrieben.

Der Urahn aller FTP-Clients ist das interaktive Textkommando `ftp`. Da es Dateien normalerweise aus dem aktuellen Verzeichnis bzw. in das aktuelle Verzeichnis überträgt, sollten Sie vor dem Start von `ftp` mit `cd` in das gewünschte Arbeitsverzeichnis wechseln. Die FTP-Sitzung wird dann mit dem Kommando `ftp user@ftpservername` oder einfach `ftp ftpservername` eingeleitet. Falls Sie Anonymous FTP nutzen möchten, geben Sie als Benutzernamen `anonymous` ein.

FTP-Kommando

Nach dem Verbindungsaufbau und der Eingabe des Passworts kann es losgehen: Mit den Kommandos `cd`, `pwd` und `ls`, die dieselbe Bedeutung wie unter Linux haben, können Sie sich durch die Verzeichnisse des FTP-Archivs bewegen. Um eine Datei vom FTP-Archiv in das aktuelle Verzeichnis Ihres Rechners zu übertragen, führen Sie `get datei` aus. Der Dateiname bleibt dabei unverändert.

Umgekehrt können Sie mit `put` eine Datei aus Ihrem aktuellen Verzeichnis in ein Verzeichnis des FTP-Archivs übertragen. Das geht freilich nur dann, wenn Sie eine Schreiberlaubnis für das Verzeichnis haben. Bei Anonymous FTP ist das zumeist nur für ein Verzeichnis mit einem Namen wie `/pub/incoming` der Fall. Die FTP-Sitzung wird mit dem Kommando `quit` oder `bye` beendet. Eine Referenz der wichtigsten FTP-Kommandos finden Sie in Tabelle 14.1.

Text- versus Binärmodus

Bevor Sie eine Datei übertragen, müssen Sie mit `binary` in den Binärmodus umschalten. Im Textmodus interpretiert FTP die Dateien als Texte und versucht, diese in das Format des jeweiligen Rechners zu konvertieren. Binärdateien werden durch so eine Konvertierung unbrauchbar. Die meisten FTP-Server sind glücklicherweise so konfiguriert, dass `binary` als Grundeinstellung gilt.

Kommando	Funktion
?	zeigt eine Liste aller FTP-Kommandos an.
!	ermöglicht die Ausführung von Shell-Kommandos.
ascii	wechselt in den Textmodus.
binary	wechselt in den Binärmodus.
bye	beendet FTP.
cd verz	wechselt in das angegebene FTP-Verzeichnis.
close	beendet die Verbindung zum FTP-Server.
get datei	überträgt die Datei vom FTP-Archiv in das aktuelle Verzeichnis.
help kommando	zeigt eine kurze Info zum angegebenen Kommando an.
lcd verz	wechselt das aktuelle Verzeichnis auf dem lokalen Rechner.
ls	zeigt die Liste der Dateien auf dem FTP-Server an.
lls	zeigt die Liste der Dateien auf dem lokalen Rechner an.
mget *.muster	überträgt alle passenden Dateien vom FTP-Archiv in das aktuelle Verzeichnis (siehe auch prompt).
open	stellt die Verbindung zum fremden Rechner her (wenn es beim ersten Versuch nicht geklappt hat).
prompt	aktiviert/deaktiviert die automatische Rückfrage vor der Übertragung jeder Datei durch mget.
put datei	überträgt die Datei in das FTP-Archiv (<i>upload</i>).
quit	beendet FTP.
reget datei	setzt die Übertragung einer bereits teilweise übertragenen Datei fort.
user	ermöglicht einen neuen Login.

Tabelle 14.1 ftp-Kommandos

Andere
FTP-Programme

Das Kommando `ftp` ist nicht komfortabel zu bedienen. Zum Glück gibt es unzählige Alternativen:

- ▶ Webbrowser, Dateimanager: Alle unter Linux verfügbaren Webbrowser und Dateimanager können auch zum FTP-Download verwendet werden. Manche Programme ermöglichen sogar einen komfortablen Upload.
- ▶ Grafische FTP-Clients: Programme wie `gftp` (Gnome) sind speziell für typische FTP-Aufgaben optimiert. Sie bieten Spezialfunktionen wie Bookmark- und Passwortverwaltung, die parallele Übertragung mehrerer Dateien, die Synchronisation von Verzeichnissen etc.

- ▶ `ncftp`: Diese Alternative zu `ftp` hat zwar eine textbasierte Benutzeroberfläche, ist aber komfortabler als das Original zu bedienen.
- ▶ `sftp`: Dieses Programm ist ähnlich minimalistisch wie `ftp`, aber dafür deutlich sicherer. Allerdings muss an der Gegenstelle ein SSH-Server laufen (kein FTP-Server). `sftp` wird im folgenden Abschnitt beschrieben.
- ▶ `wget`, `curl`, `lftp`: Diese Kommandos helfen bei der automatisierten Übertragung von Dateien bzw. ganzer Verzeichnisbäume via FTP.

Wenn Sie das Protokoll FTP nicht als Benutzer `anonymous` nutzen möchten, sondern sich mit Name und Passwort anmelden können, gilt bei den meisten FTP-Clients die folgende Syntax:

```
ftp://benutzername:password@servername
```

Manche FTP-Clients funktionieren nicht richtig, wenn sich zwischen Ihrem Rechner und dem FTP-Server eine Firewall befindet oder wenn Sie in einem lokalen Netzwerk arbeiten, das mittels Masquerading mit dem Internet verbunden ist. In solchen Fällen hilft es fast immer, den Client in einen sogenannten passiven Modus zu versetzen. Leider gibt es dafür kein einheitliches Kommando – werfen Sie also einen Blick in die Dokumentation! Die meisten Clients erkennen derartige Situationen selbstständig und aktivieren den passiven Modus automatisch.

SFTP (Secure FTP)

Das Kommando `sftp` ist Teil des `openssh`-Pakets. `sftp` verwendet intern ein ganz anderes Protokoll als `ftp` und kann wie `ssh` nur eingesetzt werden, wenn auf der Gegenstelle ein SSH-Server läuft. Anonymous FTP ist mit `sftp` nicht möglich. Davon abgesehen, erfolgt die Bedienung des Programms wie die von `ftp`. Mit `sftp -b batch-datei` können Sie SFTP-Downloads automatisieren.

Vielen ist `sftp` zu spartanisch. Die Auswahl komfortablerer SFTP-Clients ist allerdings kleiner als bei FTP. Außerdem ist manchmal etwas Überredungskunst erforderlich, bis der Verbindungsaufbau klappt:

- ▶ `gftp`: `gftp` bietet vielseitige SFTP-Konfigurationsmöglichkeiten (FTP • OPTIONEN • SSH). Wenn es Probleme gibt, achten Sie darauf, dass Sie den richtigen Port verwenden (22 für SSH, nicht 21 wie bei FTP). Häufig müssen Sie außerdem VERWENDE SSH2 SFTP FUNKTIONEN im Optionsdialog aktivieren.
- ▶ KDE, Gnome: Mit Dolphin oder Nautilus initiieren Sie eine SFTP-Verbindung, indem Sie die Adresse `sftp://user@servername` eingeben. Nach der Passwortabfrage zeigen die Programme das FTP-Verzeichnis wie ein lokales Verzeichnis an. Beide Dateimanager unterstützen auch direkt das SSH-Protokoll, das selbst dann funktioniert, wenn `sftp` nicht zur Verfügung steht. Dazu geben Sie die Adresse in der Form `fish://user@servername` an.

FTP-Adresse mit
Passwort

Passiver Modus

SFTP-Alternative

wget

Der interaktive Ansatz des Kommandos `ftp` ist zur Automatisierung von Downloads – beispielsweise in einem Script – ungeeignet. Auch sonst ist `ftp` reichlich inflexibel. Beispielsweise ist es unmöglich, einen unterbrochenen Download selbstständig wieder aufzunehmen. Abhilfe schafft das Kommando `wget`, das speziell zur Durchführung großer Downloads bzw. zur Übertragung ganzer Verzeichnisse konzipiert ist. `wget` unterstützt gleichermaßen die Protokolle FTP, HTTP und HTTPS.

Beispiele In der Grundform lädt `wget` die angegebene Datei einfach herunter:

```
user$ wget ftp://myftpserver.de/name.abc
```

Wenn der Download aus irgendeinem Grund unterbrochen wird, kann er mit `-c` ohne Umstände wieder aufgenommen werden:

```
user$ wget -c ftp://myftpserver.de/name.abc
```

Downloads von großen Dateien, beispielsweise von ISO-Images von Linux-Distributionen, dauern bei einem nicht so guten Internetzugang mehrere Stunden. Da bietet es sich an, den Download über Nacht durchzuführen. Das folgende Kommando stellt nahezu sicher, dass sich die Datei am nächsten Morgen tatsächlich auf dem Rechner befindet. Wegen `-t 20` wird der Download nach einem Verbindungsabbruch bis zu 20-mal neu aufgenommen. `--retry-connrefused` bewirkt, dass selbst nach dem Fehler *connection refused* ein neuer Versuch gestartet wird. Das ist dann zweckmäßig, wenn der Download-Server bekanntermaßen unzuverlässig ist und immer wieder für kurze Zeit unerreichbar ist.

```
user$ wget -t 20 --retry-connrefused http://mydownloadserver.de/name.iso
```

Das folgende Kommando lädt sämtliche Dateien herunter, die notwendig sind, um die angegebene Webseite später in unverändertem Zustand offline zu lesen. Kurz zur Bedeutung der Optionen: `-p` lädt auch CSS-Dateien und Bilder herunter. `-k` verändert in den heruntergeladenen Dateien die Links, sodass diese auf lokale Dateien verweisen. `-E` fügt heruntergeladenen Script-Dateien (ASP, PHP etc.) die Kennung `.html` hinzu. `-H` verfolgt auch Links auf externe Websites.

```
user$ wget -p -k -E -H http://mywebsite.de/seite.html
```

Wenn Sie eine ganze Website offline lesen möchten, hilft das folgende rekursive Download-Kommando (Option `-r`). Die Rekursionstiefe wird durch `-l 4` auf vier Ebenen limitiert.

```
user$ wget -r -l 4 -p -E -k http://mywebsite.de
```

curl

Das Kommando `curl` hilft dabei, Dateien von oder zu FTP-, HTTP- oder sonstigen Servern zu übertragen. Die `man`-Seite listet eine beeindruckende Palette von Protokollen auf, die `curl` beherrscht. In diesem Abschnitt beschränke ich mich allerdings auf FTP-Uploads. Für die Script-Programmierung besonders praktisch ist, dass `curl` auch Daten aus der Standardeingabe verarbeiten bzw. zur Standardausgabe schreiben kann. Sie müssen also nicht zuerst eine `*.tar.gz`-Datei erstellen und diese dann zum FTP-Server übertragen, sondern können beide Operationen mittels einer Pipe gleichzeitig ausführen.

Das folgende Kommando überträgt die angegebene Datei zum FTP-Server `backupserver` und speichert sie im Verzeichnis `verz`:

```
user$ curl -T datei -u username:password ftp://backupserver/verz
```

Um Daten aus dem Standardeingabekanal zu verarbeiten, geben Sie mit `-T` als Dateinamen einen Bindestrich an. Das folgende Kommando speichert das aus dem `tar`-Kommando resultierende Ergebnis direkt in der Datei `name.tgz` auf dem FTP-Server:

```
user$ tar czf - verz/ | curl -T - -u usern:pw ftp://bserver/name.tgz
```

lftp

`lftp` ist ein komfortabler interaktiver FTP-Client. Das Kommando eignet sich aber auch gut, um FTP-Uploads oder andere Kommandos in einem Script auszuführen. Dazu können Sie an `lftp` entweder mit `-c` mehrere durch Strichpunkte getrennte FTP-Kommandos übergeben oder mit `-f` eine Datei angeben, die diese Kommandos zeilenweise enthält. Das erste Kommando wird dabei immer `user benutzername,password servername` lauten, um die Verbindung zum FTP-Server herzustellen. Das folgende Kommando demonstriert einen Datei-Upload:

```
root# lftp -c "open -u username,password backupserver; put www.tgz"
```

Wenn Sie der Datei auf dem FTP-Server einen anderen Namen geben möchten, geben Sie zusätzlich die Option `-o <neuerName>` an. `lftp` zeigt während des Uploads den aktuellen Fortschritt an.

Um statt einer Datei ein ganzes Verzeichnis zum Backup-Server zu übertragen, verwenden Sie das Kommando `mirror -R`. (`mirror` kopiert normalerweise Verzeichnisse vom FTP-Server auf den lokalen Rechner. `-R` dreht die Übertragungsrichtung um.) Auch hierzu ein Beispiel:

```
root# lftp -c "open -u usern,passw bserver; mirror -R verzeichnis"
```

Im Unterschied zu anderen FTP-Clients unterstützt `lftp` das Kommando `du`, mit dem Sie feststellen können, wie viel Speicherplatz Ihre Backup-Dateien bereits belegen.

Das ist dann wichtig, wenn Ihr Speicherplatz auf dem Backup-Server streng limitiert ist. Das folgende Kommando zeigt, wie Sie ohne interaktiven Eingriff den bereits belegten Speicherplatz ermitteln. Die Option -s gibt an, dass Sie nur an der Endsumme interessiert sind. -m bewirkt, dass als Maßeinheit MiB verwendet wird.

```
user$ lftp -c "open -u username,password bserver; du -s -m"
2378 .
```

Wenn Sie das Ergebnis für eine Berechnung verwenden möchten, stört die zweite Spalte (also der Punkt, der angibt, dass sich der Zahlenwert auf das aktuelle Verzeichnis bezieht). Stellen Sie dem Kommando einfach cut -f 1 hintan, um die erste Spalte zu extrahieren:

```
user$ lftp -c "open -u usern,passw bserver; du -s -m" | cut -f 1
2378
```

rsync, mirror, sitecopy

rsync hilft dabei, ganze Verzeichnisbäume zu kopieren bzw. zu synchronisieren. Eine ausführliche Beschreibung dieses Kommandos finden Sie in Abschnitt 36.4, »Verzeichnisse synchronisieren (rsync)«. Sofern auf dem Partnerrechner weder ein SSH-noch ein rsync-Server läuft, können Sie anstelle von rsync auf die Kommandos mirror oder sitecopy zurückgreifen. Das Perl-Script mirror aus dem gleichnamigen Paket kopiert ganze Verzeichnisbäume von einem FTP-Server auf den lokalen Rechner. Das Kommando sitecopy ist hingegen dahingehend optimiert, einen Verzeichnisbaum auf einen Webserver hochzuladen, wobei der Datentransfer wahlweise via FTP oder WebDAV erfolgt.

14.4 Lynx

Webbrowser im Textmodus

Webbrowser wie Firefox oder Chrome sind in einer Textkonsole oder in einem Terminalfenster unbrauchbar. Um dennoch auch im Textmodus rasch eine Webseite zu besuchen oder ein HTML-Dokument zu lesen, helfen Programme wie ELinks, Lynx oder w3m. Nebenbei können Sie mit diesen Programmen einfache HTML-Dokumente in reinen Text umwandeln. Alle drei Programme sind ähnlich zu bedienen. Zahlreiche Optionen sowie Tastenkürzel sind in den man-Seiten bzw. im integrierten Hilfesystem dokumentiert. Aus Platzgründen stelle ich hier nur exemplarisch das bekannteste Programm Lynx näher vor.

Lynx Die Bedienung von Lynx ist einfach: Sie starten das Programm im Regelfall dadurch, dass Sie eine WWW-Adresse oder den Namen einer HTML-Datei als Parameter angeben. Lynx lädt das Dokument und zeigt die erste Seite an, wobei Überschriften und Links durch unterschiedliche Farben gekennzeichnet sind. Wenn Sie Lynx mit der

Option -use_mouse starten, können Sie das Programm auch per Maus bedienen: Mit der linken Taste folgen Sie einem Link, die mittlere Taste zeigt ein Kontextmenü an, und die rechte Taste führt zur vorherigen Seite zurück.

Lynx verwendet zur Ausgabe standardmäßig den Latin-1-Zeichensatz. Damit Sonderzeichen in Unicode-Konsolen richtig dargestellt werden, geben Sie die Option -display_charset=utf-8 an. Das folgende Kommando zeigt, wie Sie Lynx als Konverter von HTML in reinen Text einsetzen:

```
user$ lynx -dump quelle.html > ziel.txt
```

14.5 Mutt

Zum Lesen lokaler E-Mails bietet sich das textbasierte E-Mail-Programm Mutt an (siehe Abbildung 14.2). Vor dem ersten Einsatz muss das zumeist gleichnamige Paket installiert werden. In einem Konsolenfenster führen Sie zuerst su -l aus, um sich als root anzumelden, und starten das Programm dann mit dem Kommando mutt.

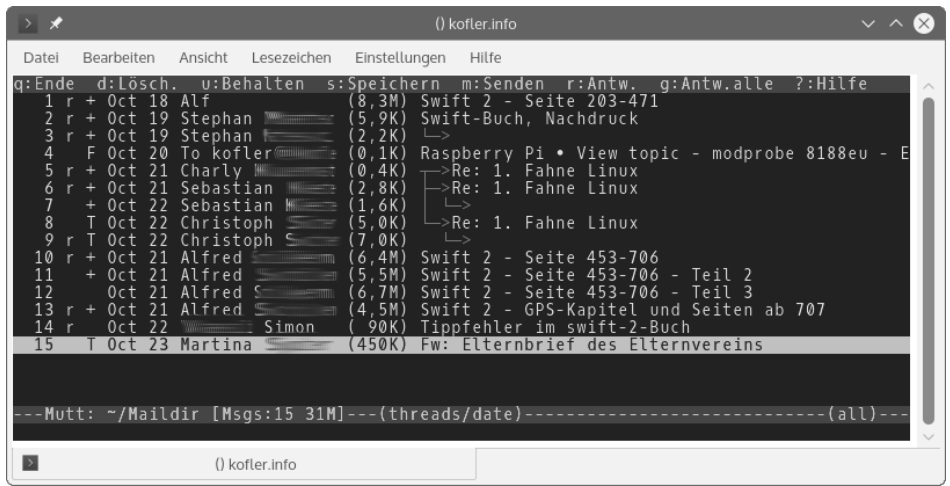


Abbildung 14.2 Lokale E-Mails mit Mutt lesen

Das Programm zeigt auf der Startseite die Titelzeilen aller E-Mails an. Wenn der aktive Benutzer noch keine einzige E-Mail empfangen hat, beklagt sich Mutt darüber, dass es die Datei /var/mail/benutzer noch nicht gibt. Diese Warnung können Sie ignorieren. Sie tritt nicht mehr auf, sobald die erste E-Mail eingetroffen ist.

Mit den Cursortasten bewegen Sie sich durch die Inbox. ← zeigt den Text der ausgewählten E-Mail an. Mit der Leertaste blättern Sie durch die Nachricht. J führt zur nächsten Nachricht, I zurück in die Inbox. ? zeigt einen Hilfetext mit allen wichtigen Tastenkürzeln an.

Um eine neue E-Mail zu verfassen, drücken Sie **[M]** und geben den Empfänger und die Subject-Zeile an. Anschließend startet Mutt den Editor, den Sie mit der Umgebungsvariable `$EDITOR` oder mit dem Link `/etc/alternatives/editor` ausgewählt haben. Dort schreiben Sie den Nachrichtentext, speichern ihn und verlassen den Editor. Anschließend versenden Sie die E-Mail in Mutt durch **[Y]**.

[Q] beendet das Programm. Beim Verlassen stellt Mutt zwei Fragen: Sollen mit **[D]** als gelöscht markierte E-Mails endgültig gelöscht werden? Und sollen gelesene Nachrichten nach `/home/username/mbox` verschoben werden? Wenn Sie vorhaben, die E-Mails später noch mit einem anderen Programm zu bearbeiten, sollten Sie beide Fragen mit **[N]** beantworten. Besonders die zweite Frage ist kritisch: In der lokalen `mbox`-Datei findet nur noch Mutt die E-Mails, nicht aber ein externes Programm wie z. B. der POP-Server Dovecot.

Konfiguration Mutt funktioniert auf Anhieb, wenn sich Ihre E-Mail in einer `mbox`-Datei im Verzeichnis `/var/mail/name` befindet. Wenn Ihre E-Mails hingegen im Maildir-Format im Verzeichnis Maildir gespeichert werden, müssen Sie die Konfigurationsdatei `.muttrc` mit dem folgenden Inhalt einrichten:

```
# Datei .muttrc
set mbox_type=Maildir
set folder=~/.Maildir
set mask="!^\\.[^.]"
set mbox=~/.Maildir
set record="+.Sent"
set postponed="+.Drafts"
set spoolfile=~/.Maildir
```

Weitere Maildir-Konfigurationstipps für diverse Spezialfälle finden Sie hier:

<https://dev.mutt.org/trac/wiki/MuttFaq/Maildir>

<https://eising.wordpress.com/mutt-maildir-mini-howto>

Kapitel 32

Apache

In diesem Kapitel beschreibe ich, wie Sie Ihren eigenen Webserver aufsetzen. Im Mittelpunkt des Kapitels steht das Programm Apache und seine Basiskonfiguration inklusive der Verwendung der kostenlosen HTTPS-Schlüssel von *Let's Encrypt*. Darüber hinaus gehe ich auch auf einige beliebte Erweiterungen ein, unter anderem auf die Programmiersprache PHP und auf das Programm GoAccess zur Erstellung von Zugriffsstatistiken. Das Kapitel endet mit Informationen zu FTP – und der Empfehlung, auf einen FTP-Server möglichst zu verzichten.

Ich gehe in diesem Kapitel davon aus, dass Sie einen öffentlichen Webserver im Internet betreiben möchten. Grundsätzlich ist es natürlich auch möglich, Apache nur innerhalb eines LANs einzusetzen, beispielsweise als firmeninternes Kommunikationszentrum mit einem Wiki und Seiten zur Projektplanung, Zeiterfassung etc. In diesem Fall müssen Sie aber unbedingt sicherstellen, dass die hier gesammelten Daten tatsächlich intern bleiben und dass kein ungeschützter Webzugriff aus dem Internet möglich ist (siehe auch Abschnitt 32.2, »Webverzeichnisse einrichten und absichern«).

Generell gilt: Dieses Kapitel ist lediglich eine Einführung in die Konfiguration von Apache und beschreibt bestenfalls ein Prozent der Schlüsselwörter zur Apache-Konfiguration. Der professionelle Einsatz von Apache setzt das Studium weiterführender Dokumentation voraus, sei es in Buchform oder aus dem Internet.

32.1 Apache

Apache ist der beliebteste Webserver der Open-Source-Welt. Im März 2017 liefen laut <https://netcraft.com> ca. 46 Prozent aller aktiven Websites unter Apache. Wird nur die Million der am meisten besuchten Websites betrachtet, beträgt der Marktanteil ca. 41 Prozent. Aktuelle Informationen sowie eine umfassende Dokumentation zu Apache finden Sie auf der Apache-Website:

<https://httpd.apache.org>

Versionen und Alternativen Die aktuelle Apache-Version ist 2.4.n. Version 2.2.n wird zwar ebenfalls noch gewartet, kommt aber nur noch auf Langzeit-Server-Installationen zum Einsatz. Aktuelle Linux-Distributionen verwenden durchwegs Apache 2.4, weswegen ich in diesem Buch auf Version 2.2 nicht mehr eingehe.

Zunehmend beliebt mit einem Marktanteil von rund 20 Prozent ist der Webserver nginx: Auch dabei handelt es sich um ein Open-Source-Programm, das speziell im Hinblick auf hohe Geschwindigkeit und Skalierbarkeit optimiert wurde.

Installation Eine typische Apache-Installation besteht aus zahlreichen zusammengehörenden Paketen: dem Server an sich, diversen Bibliotheken, Plugins, Programmiersprachen etc. Um Ihnen die Installation zu erleichtern, können Sie bei einigen Distributionen jeweils eine ganze Gruppe von Paketen zur Installation auswählen. Damit werden neben Apache auch die wichtigsten MySQL- und PHP-Pakete installiert.

```
root# taskel install web-server           (Debian)
root# yum groupinstall 'Web-Server'      (CentOS, RHEL)
root# dnf groupinstall 'Web-Server'      (Fedora)
root# zypper intall -t pattern lamp_server (SUSE)
user$ sudo apt install taskel           (Ubuntu)
user$ sudo taskel install lamp-server
```

Das richtige Multi-Processing-Modul (MPM)

Apache stellt vier unterschiedliche Multi-Processing-Module zur Auswahl, nämlich perchild, prefork, worker und event. Diese Multi-Threading-Verfahren haben Einfluss darauf, wie effizient Apache mehrere Anfragen gleichzeitig verarbeiten kann. Beim Einrichten von Apache müssen Sie sich für eine dieser Varianten entscheiden, indem Sie das entsprechende apache2-mpm-xxx-Paket installieren.

Wenn Sie zusammen mit Apache die Programmiersprache PHP einsetzen möchten, ist das Verfahren prefork die sicherste Wahl. Bei den anderen Varianten sind Fehler aufgrund von nicht threadsicheren PHP-Bibliotheken möglich:

<https://php.net/manual/en/faq.installation.php>

Start/Stopp Apache ist ein Dämon, der je nach Distribution explizit gestartet werden muss. Eine Zusammenfassung der erforderlichen Kommandos finden Sie in Abschnitt 12.5, »Systemprozesse (Dämonen)«. Der Name des Init-Scripts variiert je nach Distribution: Er lautet apache2 bei Debian, SUSE und Ubuntu bzw. httpd bei CentOS, Fedora und RHEL.

Firewall Unter CentOS, Fedora, RHEL und (open)SUSE blockiert die standardmäßig aktive Firewall den Zugriff auf den Webserver von außen. Sie können Apache also vorerst nur direkt auf dem Rechner ausprobieren, auf dem der Webserver läuft (<http://localhost>). Damit der Webserver auch von außen erreichbar ist, müssen Sie in der Firewall

Ausnahmeregeln für die Protokolle HTTP und HTTPS definieren, also für die Port-Nummern 80 und 443.

Unter SUSE verwenden Sie zur Firewall-Konfiguration am besten YaST. Unter CentOS/Fedora/RHEL können Sie wie folgt vorgehen: Sie stellen zuerst fest, welche Firewall-Zone für die Netzwerkschnittstelle zum Internet gilt (häufig public, hier aber FedoraWorkstation), und aktivieren dann für diese Zone Ausnahmeregeln:

```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=http
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=https
root# firewall-cmd --reload
```

Alternative Verfahren zur Firewall-Konfiguration sowie eine Menge Hintergrundinformationen zu diesem Thema folgen in Kapitel 37, »Firewalls«.

Der Service-Name für das Init-System variiert je nach Distribution. Aus Sicherheitsgründen wird der Webserver wie die meisten anderen Netzwerkdämonen nicht unter dem Account root ausgeführt, sondern unter einem anderen Account. Dessen Namen stellen Sie am einfachsten mit ps axu fest. Tabelle 32.1 fasst zusammen, unter welchem Namen Apache dem Init-System bekannt ist, unter welchem Account das Programm läuft und wo sich standardmäßig die HTML-Dateien befinden.

Name und Account

Distribution	Prozessname	Account	DocumentRoot
Debian, Ubuntu	apache2	www-data	/var/www/html
CentOS, Fedora, RHEL	httpd	apache	/var/www/html
SUSE	httpd-threadverfahren	wwwrun	/srv/www/htdocs

Tabelle 32.1 Programmname, Account und DocumentRoot-Verzeichnis von Apache

Um zu testen, ob alles funktioniert, starten Sie auf dem lokalen Rechner einen Webbrowser und geben als Adresse <http://localhost/> oder <http://servername/> ein. Sie sollten nun eine Testseite des Webserver sehen (siehe Abbildung 32.1).

Test

Damit statt der Testseite die Startseite Ihres eigenen Webauftritts erscheint, müssen Sie Ihre HTML-Dateien in das Dokumentverzeichnis von Apache speichern. Auch dieses Verzeichnis ist distributionsabhängig (Schlüsselwort DocumentRoot in den Konfigurationsdateien, siehe Tabelle 32.1). Ihre HTML-Dateien müssen für den Account des Apache-Webserver lesbar sein!

Eigene HTML-Seiten

Wenn Sie unter Fedora oder RHEL arbeiten, müssen Sie außerdem darauf achten, dass alle HTML-Dateien mit dem SELinux-Attribut httpd_sys_content_t ausgestattet sind. Für Dateien innerhalb von /var/www/html erreichen Sie das am einfachsten durch das folgende Kommando:

SELinux

```
root# restorecon -R -v /var/www/html/*
```

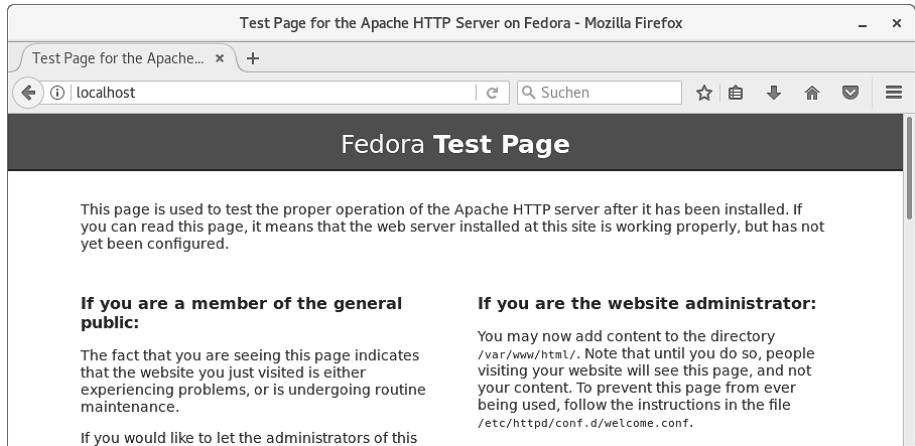



Abbildung 32.1 Apache-Testseite eines Fedora-Rechners

Wenn Sie Ihre HTML-Dateien in einem anderen Verzeichnis ablegen, ist hingegen das folgende Kommando erforderlich:

```
root# chcon -R system_u:object_r:httpd_sys_content_t:s0 /mein-web-verzeichnis
```

Beachten Sie, dass für CGI-, Webalizer- und Konfigurationsdateien andere Attribute vorgesehen sind. Details zum SELinux-Modul für Apache können Sie mit `man httpd_selinux` nachlesen, wenn Sie vorher das Paket `selinux-policy-doc` installieren.

Konfiguration

In diesem Buch fehlt der Platz, um detailliert auf alle Konfigurationsoptionen und -varianten einzugehen. Ich möchte Ihnen an dieser Stelle aber zumindest einen Überblick darüber geben, wo sich die Konfigurationsdateien je nach Distribution befinden und wie ganz elementare Einstellungen durchgeführt werden.

Früher erfolgte die Konfiguration von Apache durch eine einzige Datei `httpd.conf`, wobei deren genauer Ort distributionsabhängig war. Diese Konfigurationsdatei wurde im Laufe der Zeit immer unübersichtlicher.

Aus diesem Grund sind die meisten Distributionen dazu übergegangen, die Einstellungen auf diverse Dateien zu verteilen, die durch `Include`-Anweisungen aus verschiedenen Verzeichnissen gelesen werden (siehe Tabelle 32.2 bis Tabelle 32.4). Das macht jede einzelne Datei übersichtlicher und ermöglicht eine automatisierte Wartung – also beispielsweise das Aktivieren oder Deaktivieren von Plugins durch Kommandos oder Scripts.

Dateien	Inhalt
/etc/apache2/apache2.conf	Startpunkt
/etc/apache2/httpd.conf	benutzerspezifische Konfiguration
/etc/apache2/ports.conf	überwachte Ports, normalerweise Port 80
/etc/apache2/conf.d/*	weitere Konfigurationsdateien
/etc/apache2/mods-available/	verfügbare Erweiterungsmodule
/etc/apache2/mods-enabled/*.conf	Links auf aktive Erweiterungsmodule
/etc/apache2/conf-available/	verfügbare Konfigurationsdateien
/etc/apache2/conf-enabled/*.conf	Links auf aktive Konfigurationsdateien
/etc/apache2/sites-available/	verfügbare Websites (virtuelle Hosts)
/etc/apache2/sites-enabled/*.conf	Links auf aktive Websites
/etc/apache2/envvars	Umgebungsvariablen für das Init-Script

Tabelle 32.2 Apache-Konfiguration bei Debian und Ubuntu

Dateien	Inhalt
/etc/httpd/conf/httpd.conf	Startpunkt
/etc/httpd/conf/magic	MIME-Konfiguration (für <code>mod_mime</code>)
/etc/httpd/conf.d/*.conf	sonstige Konfigurationsdateien

Tabelle 32.3 Apache-Konfiguration bei CentOS, Fedora und Red Hat

Dateien	Inhalt
/etc/apache2/httpd.conf	Startpunkt
/etc/apache2/*.conf	globale Konfigurationsdateien
/etc/apache2/conf.d/*.conf	sonstige Konfigurationsdateien
/etc/apache2/sysconf.d/*.conf	automatisch erzeugte Systemkonfigurationsdateien
/etc/apache2/vhosts.d/*.conf	Websites (virtuelle Hosts)
/etc/sysconfig/apache2	Grundeinstellungen

Tabelle 32.4 Apache-Konfiguration bei SUSE

Wenn Sie ein bestimmtes Schlüsselwort in den Konfigurationsdateien suchen, gehen Sie am besten so vor:

```
user$ cd /etc/httpd (bzw.) cd /etc/apache2
user$ grep -i -r Schlüsselwort
```

Debian/Ubuntu Bei Debian/Ubuntu enthält das Verzeichnis `mods-available` eine Kollektion von `*.load-` und `*.conf`-Dateien für diverse Apache-Module. Um weitere Module zu aktivieren, richten Sie in `mods-enabled` Links auf diese Dateien ein. Bei der Verwaltung der Links helfen die Debian-spezifischen Kommandos `a2enmod` und `a2dismod`.

Mit `a2ensite` und `a2dissite` aktivieren bzw. deaktivieren Sie virtuelle Hosts. Standardmäßig enthält `sites-available` nur die Dateien `000-default.conf` und `default-ssl.conf`: Dort befinden sich diverse Grundeinstellungen für das Verzeichnis `/var/www`. Der Mechanismus funktioniert wie bei den Modulen: Das Verzeichnis `sites-available` enthält die Konfigurationsdateien für alle Hosts, in `sites-enabled` befinden sich die entsprechenden Links.

Derselbe Mechanismus kümmert sich auch um sonstige Konfigurationsdateien. Die Dateien befinden sich im Verzeichnis `conf-available`. Mit den Kommandos `a2enconf` bzw. `a2disconf` werden im Verzeichnis `conf-enabled` entsprechende Links darauf eingerichtet bzw. wieder entfernt. Bei älteren Ubuntu-Versionen sowie bei Debian werden derartige Konfigurationsdateien in `conf.d` gespeichert. Die Verwaltung erfolgt manuell ohne Kommandos.

SUSE Bei SUSE werden sämtliche `*.conf`-Dateien im Verzeichnis `sysconf.d` bei jedem Apache-Start durch das Init-System neu erstellt. Es ist daher zwecklos, Änderungen an diesen Dateien vorzunehmen. Vielmehr müssen Sie die Variablen in `/etc/sysconfig/apache2` ändern. In dieser Datei ist auch festgelegt, welche Module beim Apache-Start geladen werden (Variable `APACHE_MODULES`). Wenn Sie den SUSE-Konfigurationsdateien eine eigene Datei hinzufügen möchten, geben Sie deren Dateinamen in der Variablen `APACHE_CONF_INCLUDE_FILES` an.

Konfiguration testen Nach Änderungen an der Syntax können Sie mit `httpd -t`, `httpd2 -t` bzw. `apache2 -t` testen, ob die Konfiguration frei von Syntaxfehlern ist. Bei Debian und Ubuntu müssen Sie vorher einige Umgebungsvariablen aus `envvars` einlesen:

```
root# . /etc/apache2/envvars
root# apache2 -t
Syntax OK
```

Anschließend fordern Sie Apache dazu auf, die Konfigurationsdateien neu einzulesen:

```
root# systemctl restart apache2|httpd
```

Der Webserver Apache funktioniert zwar im Regelfall auf Anhieb. Je nach Netzwerk-konfiguration müssen Sie aber oft eine Zeile in den Konfigurationsdateien ändern bzw. zu ihnen hinzufügen: `ServerName` sollte den Namen Ihres Rechners enthalten. Falls diese Einstellung nicht wirksam wird, müssen Sie außerdem die Einstellung `UseCanonicalName Off` verwenden.

```
# in /etc/apache2/httpd.conf (Debian/Ubuntu)
# bzw. /etc/httpd/conf/httpd.conf (Fedora/Red Hat)
ServerName mars.sol # geben Sie hier den Namen Ihres Rechners an
```

Bei SUSE stellen Sie den Rechnernamen in `/etc/sysconfig/apache2` mit der Variablen `APACHE_SERVERNAME` ein.

Sofern der Root-Server über eine IPv6-Adresse verfügt, beantwortet Apache auch IPv6-Webanfragen. Dafür verantwortlich ist die Standardeinstellung `Listen 80`, mit der Apache den Port 80 überwacht, unabhängig von der IP-Version. Wenn Sie IPv6 deaktivieren möchten, fügen Sie die Anweisung `Listen 0.0.0.0:80` in die passende Konfigurationsdatei ein. Falls Apache auch HTTPS-Seiten liefern soll, benötigen Sie eine weitere `Listen`-Anweisung für den Port 443:

```
# Datei /etc/apache2/ports.conf (Debian, Ubuntu)
# Dateien /etc/httpd/conf/httpd.conf und conf.d/ssl.conf (CentOS, Fedora, RHEL)
# Datei /etc/apache2/listen.conf (SUSE)
Listen 0.0.0.0:80
Listen 0.0.0.0:443 https
```

Standardzeichensatz

Bei allen gängigen Linux-Distributionen gilt automatisch der Unicode-Zeichensatz UTF-8. Wenn Sie also mit einem Texteditor eine Textdatei erstellen, die die deutschen Buchstaben ä, ö, ü oder ß enthält, werden diese in der UTF-8-Codierung gespeichert.

Apache ist die Codierung der HTML-Dateien grundsätzlich egal. Das Programm überträgt die Dateien einfach Byte für Byte an den Webbrowser, der die Seite angefordert hat. Allerdings sendet Apache zusätzlich einen sogenannten Header mit, der unter anderem Informationen darüber enthält, in welchem Zeichensatz die Seite codiert ist. Der Webbrowser wertet diese Information aus und verwendet den angegebenen Zeichensatz zur Darstellung der Seite.

Der springende Punkt ist nun, dass Apache den richtigen Zeichensatz angibt: Wenn das schiefgeht, sieht der Benutzer in seinem Webbrowser statt ä oder ü irgendwelche merkwürdigen Zeichenkombinationen. Aus diesem Grund bietet Apache diverse Möglichkeiten zur Zeichensatzkonfiguration:

ServerName

IPv6 blockieren

Zeichensatz einstellen

- **AddDefaultCharset off:** Bei dieser Einstellung wertet Apache das `<meta>`-Tag in der zu übertragenden HTML-Datei aus und sendet den dort angegebenen Zeichensatz an den Browser. Wenn die HTML-Datei wie folgt beginnt, kommt der Zeichensatz Unicode UTF-8 zur Anwendung:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    ...
```

- **AddDefaultCharset Zeichensatz:** Apache überträgt den hier angegebenen Zeichensatz für alle Seiten an den Browser. Die Einstellung gilt sowohl für HTML- als auch für PHP-Dateien. Das `<meta>`-Tag im HTML-Code wird ignoriert.
- **AddCharset Zeichensatz kennung:** Damit wird ein Zeichensatz für Dateien mit einer bestimmten Kennung eingestellt. `AddCharset utf-8 .utf8` bewirkt also, dass für alle Dateien, deren Name auf `.utf8` endet, als Zeichensatz Unicode UTF-8 an den Browser gesendet wird. `AddCharset` setzt das Apache-Modul `mod_mime` voraus.

Debian, Ubuntu Natürlich gilt je nach Distribution eine unterschiedliche Standardkonfiguration. Für die globale Voreinstellung des Zeichensatzes ist unter Debian und Ubuntu die Konfigurationsdatei `/etc/apache2/conf-enabled/charset.conf` vorgesehen. Normalerweise ist diese Datei leer, d. h., es gilt `AddDefaultCharset off`.

Sie können `AddDefaultCharset` und `AddCharset` auch in den Konfigurationsdateien für virtuelle Hosts (Verzeichnis `sites-available`) sowie in `.htaccess`-Dateien einsetzen, wenn Sie eine `host-` bzw. `verzeichnispezifische` Konfiguration wünschen. Beachten Sie aber, dass die Zeichensatzeinstellungen in `.htaccess` nur berücksichtigt werden, wenn für das Webverzeichnis `AllowOverride All` oder `FileInfo` gilt.

Fedora, Red Hat Bei Fedora und Red Hat gilt `AddDefaultCharset UTF-8`. Die Einstellung befindet sich in `/etc/httpd/conf/httpd.conf`. In derselben Datei ist auch `AllowOverride None` für das Verzeichnis `/var/www/html` eingestellt.

SUSE Bei SUSE fehlt in den Konfigurationsdateien eine explizite Zeichensatzeinstellung. Damit gilt `AddDefaultCharset off`, d. h., die `<meta>`-Informationen in den HTML-Dateien sind für die richtige Zeichensatzerkennung entscheidend. Ein geeigneter Ort zur Einstellung von `AddDefaultCharset` ist die Datei `/etc/apache2/mod_mime-defaults.conf`. Auch bei SUSE gilt `AllowOverride None` für das Verzeichnis `/srv/www/htdocs`. Sie können die Einstellung in `/etc/apache2/default-server.conf` verändern.

Logrotate

Die Logging-Dateien von Apache zählen bei vielen Servern zu den Dateien, die am schnellsten wachsen. Deswegen müssen Sie sich darum kümmern, dass die Logging-Dateien regelmäßig umbenannt, komprimiert und schließlich gelöscht werden. Genau diese Aufgabe erledigt das Programm Logrotate (siehe Abschnitt 18.9, »Logging (Syslog)«), das auf Linux-Servern in der Regel standardmäßig installiert ist.

Das Programm wird üblicherweise einmal täglich durch `/etc/cron.daily/logrotate` gestartet. In der Standardkonfiguration verarbeitet es die Apache-Logging-Dateien `/var/log/httpd/*.log` (Fedora/RHEL) bzw. `/var/log/apache2/*.log` (Debian/Ubuntu) einmal pro Woche, benennt sie in `name.nn` um und komprimiert sie. Die komprimierten Dateien werden für 52 Wochen archiviert und dann gelöscht.

Falls Sie bei der Konfiguration virtueller Hosts eigene Logging-Verzeichnisse definieren, müssen Sie in der Konfigurationsdatei `/etc/logrotate.d/apache2` die erste Zeile anpassen und dort die Orte der zusätzlichen Logging-Dateien angeben. Dabei sind auch Muster wie `/home/*/www-log/*.log` erlaubt:

```
# Datei /etc/logrotate.d/apache2
/var/log/apache2/*.log /home/meinefirma/www-log/*.log {
    weekly
    missingok
    rotate 52
    ...
}
```

32.2 Webverzeichnisse einrichten und absichern

Nach der Grundkonfiguration von Apache werden Sie in der Regel verschiedene Webverzeichnisse einrichten, die jene HTML- und PHP-Dateien enthalten, aus denen sich Ihre Webseite zusammensetzt. Wenn Sie also beispielsweise WordPress als CMS für Ihre Webseite einrichten möchten, laden Sie die Installationsdateien herunter, richten ein für Apache erreichbares Verzeichnis ein und packen die Dateien dort aus. Dieser Abschnitt beschäftigt sich natürlich nicht mit den Details der WordPress-Installation, erläutert aber, welche Einstellungen Sie in Apache für das Verzeichnis vornehmen müssen, in dem Sie WordPress, phpMyAdmin, ownCloud/Nextcloud oder irgendeine andere Webapplikation einrichten möchten.

Auf den folgenden Seiten gehe ich dabei von der Apache-Standardkonfiguration aus, wie Sie sie unter Ubuntu bzw. Debian vorfinden. Wenn Sie mit einer anderen Distribution arbeiten, gibt es bei der Standardkonfiguration kleine Variationen. Die hier präsentierten Schlüsselwörter und Arbeitstechniken gelten aber auch dort.

Ubuntu-Standard-konfiguration

Unter Ubuntu ist Apache so vorkonfiguriert, dass für die Standard-Website Dateien aus dem Verzeichnis `/var/www` verwendet werden. Die erforderlichen Einstellungen befinden sich in der Datei `/etc/apache2/sites-available/000-default.conf`:

```
# Datei /etc/apache2/sites-available/000-default.conf (Ubuntu)
<VirtualHost *:80>
    ServerAdmin      webmaster@localhost
    DocumentRoot     /var/www/html
    ErrorLog          ${APACHE_LOG_DIR}/error.log
    CustomLog         ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Eine Debian- bzw. Ubuntu-spezifische Besonderheit der Defaultkonfiguration besteht darin, dass alle Einstellungen in einer `<VirtualHost>`-Gruppe gebündelt sind. `<VirtualHost>`-Gruppen dienen dazu, Einstellungen für mehrere eigenständige Hosts (Websites) voneinander zu trennen (siehe Abschnitt 32.3, »Virtuelle Hosts«). Der Host in der Datei `default` ist allerdings weder an eine IP-Adresse noch an einen Hostnamen gekoppelt und gilt aus diesem Grund für alle Webzugriffe, die nicht einem speziellen virtuellen Host zugeordnet werden können.

Host-Konfiguration

Mit den im Folgenden beschriebenen Schlüsselwörtern zur Konfiguration einer `<VirtualHost>`-Gruppe werden die Details des Hosts festgelegt – also die Herkunft der Daten, die E-Mail-Adresse des Administrators, der Ort der Logging-Dateien etc.:

- `DocumentRoot` gibt an, in welchem Verzeichnis sich die HTML-Dateien befinden.
- `ServerAdmin` gibt die E-Mail-Adresse des Administrators des virtuellen Hosts an. Die Adresse wird z. B. bei Fehlermeldungen angezeigt. Sie sollten hier eine E-Mail-Adresse angeben, die tatsächlich aktiv ist. Üblich ist `webmaster@hostname`.
- `ServerSignature` steuert, ob Apache bei selbst generierten Dokumenten (Fehlermeldungen, Verzeichnislisten etc.) am Ende eine Signatur hinzufügen soll. Die Signatur besteht aus der Apache-Version und dem Hostnamen. Mit `ServerSignature=EMail` wird auch die E-Mail-Adresse des Administrators hinzugefügt.
- `LogLevel` bestimmt, in welchem Ausmaß Webserver-Probleme protokolliert werden sollen. Mögliche Werte reichen von `emerg` (nur kritische Fehler protokollieren, die zum Ende von Apache führen) bis `debug` (alles protokollieren, selbst Debugging-Texte). Sinnvolle Einstellungen sind in der Regel `error` oder `warn`. Letztere Einstellung gilt per Default.
- `ErrorLog` gibt den Dateinamen der Protokolldatei für Fehlermeldungen an.

- `CustomLog` gibt den Dateinamen des Zugriffsprotokolls an. In dieser Datei protokolliert Apache jede erfolgreiche Übertragung einer Datei. An den zweiten Parameter übergeben Sie entweder den Namen eines vordefinierten Loggingformats oder eine Zeichenkette mit eigenen Formatanweisungen. Die erlaubten Formatcodes sind hier beschrieben:

```
http://httpd.apache.org/docs/2.4/de/mod/mod_log_config.html
```

Unter Ubuntu sind in `apache2.conf` einige Formate vorkonfiguriert, z. B. `combined` oder `common`.

- `ErrorDocument` gibt an, wie Apache auf Fehler reagieren soll. Als ersten Parameter geben Sie die Fehlernummer an (z. B. 404 für *not found*), im zweiten Parameter den Namen einer lokalen Datei bzw. die Adresse einer externen Seite, die in diesem Fall angezeigt werden soll. Der Dateiname muss relativ zu `DocumentRoot` angegeben werden. Die wichtigsten Fehlercodes sind:

- 400 *Bad Request*
- 401 *Authorization Required*
- 403 *Forbidden*
- 404 *Not Found*
- 500 *Internal Server Error*

Eine Liste aller HTTP-Statuscodes finden Sie hier:

```
https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
```

Standardmäßig ist `ErrorDocument` nicht konfiguriert. Um unschöne Fehlermeldungen zu vermeiden, sollten Sie sich die Mühe machen, eine Fehlerseite einzurichten und deren Ort mit `ErrorDocument` anzugeben.

- `Alias` stellt eine Zuordnung zwischen einem Webverzeichnis und einem Verzeichnis der Festplatte (auch außerhalb von `DocumentRoot`) her. Beispielsweise bewirkt `Alias /mytool /usr/local/mytool`, dass bei Zugriffen auf `http://meinserver.de/mytool` die Dateien aus dem Verzeichnis `/usr/local/mytool` gelesen werden.

In der Regel müssen Sie für jedes `alias`-Verzeichnis in einer `<Directory>`-Gruppe die Zugriffsrechte einstellen (siehe den folgenden Abschnitt). Zu `Alias` gibt es die Variante `ScriptAlias`, die zur Definition von Verzeichnissen mit CGI-Scripts dient.

Verzeichniskonfiguration

Im Anschluss an diese Einstellungen, die für den gesamten virtuellen Host gelten, können Sie in einer oder mehreren `<Directory "/verzeichnis/">`-Gruppen die Eigenschaften für einzelne Verzeichnisse Ihres Hosts einstellen. Die folgende Liste nennt hierfür nur die wichtigen Schlüsselwörter:

- **DirectoryIndex** gibt an, welche Datei Apache senden soll, wenn eine Adresse mit / endet und somit ein ganzes Verzeichnis betrifft (standardmäßig index.html). Es dürfen auch mehrere Dateien angegeben werden. In diesem Fall arbeitet Apache alle Angaben der Reihe nach bis zum ersten Treffer ab (z. B. DirectoryIndex index.php index.html).
- **Options** ermöglicht die Angabe diverser Optionen, die für das Verzeichnis gelten. Dazu zählen:

ExecCGI	CGI-Scripts ausführen
FollowSymLinks	symbolische Links verfolgen
Includes	Include-Dateien hinzufügen (Modul mod_include)
Indexes	Dateiliste anzeigen, wenn index.html fehlt
MultiViews	automatische Sprachauswahl (Modul mod_negotiation)

Standardmäßig gilt in Apache die Einstellung All. Damit sind alle Optionen mit der Ausnahme von MultiViews aktiv. Die Ubuntu-Konfiguration ist etwas restriktiver: Für das gesamte Dateisystem gilt Options FollowSymLinks, für das /var/www-Verzeichnis gilt Options Indexes FollowSymLinks MultiViews.

Um einzelne Optionen gegenüber den Voreinstellungen eines übergeordneten Verzeichnisses wieder zu deaktivieren, muss ein Minuszeichen vorangestellt werden. Ein vorangestelltes Pluszeichen ist ebenfalls erlaubt, hat aber keine Wirkung: Die Option ist genau so aktiviert, als wäre sie ohne Pluszeichen angegeben.

Aus Sicherheitsgründen sollte für Options die Devise »Weniger ist mehr« gelten: Die Option Indexes verrät neugierigen Websurfern die Namen aller Dateien, die sich in einem Verzeichnis befinden, sofern Sie einmal index.html vergessen. Das ist ein potenzielles Sicherheitsrisiko. MultiView brauchen Sie nur für mehrsprachige Websites mit automatischer Sprachauswahl. Bietet Ihre Seite so etwas nicht, können Sie auch auf diese Option verzichten.

- **AllowOverride** gibt an, welche Einstellungen verzeichnisspezifisch durch eine .htaccess-Datei verändert werden dürfen. Zur Auswahl stehen:

AuthConfig	Authentifizierungsverfahren einstellen
FileInfo	Datei- und Dokumenttypen einstellen
Indexes	Verzeichnisindex modifizieren
Limit	Zugriffsrechte ändern (Allow, Deny, Order)
Options	Verzeichnisoptionen ändern

Standardmäßig sind in Apache alle Möglichkeiten aktiv, d. h., jede Option kann verändert werden. Bei den meisten Distributionen ist die Standardkonfiguration aus Sicherheitsgründen aber restriktiver. So ist unter Debian und Ubuntu für alle relevanten Verzeichnisse None voreingestellt (siehe /etc/apache2/apache2.conf).

Verzeichnisse absichern

Auf einen zentralen Punkt bin ich bisher noch nicht eingegangen: auf die Steuerung der Zugriffsrechte für Verzeichnisse. Die Apache-Versionen 2.2 und 2.4 unterscheiden sich hierbei deutlich. Da die Version-2.2-Syntax auf vielen Apache-2.4-Installationen weiterhin funktioniert und immer noch weitverbreitet ist, gehe ich hier auf beide Varianten ein. Für Neuinstallationen sollten Sie aber unbedingt die Schlüsselwörter von Apache 2.4 verwenden!

In Apache 2.2 können Sie innerhalb der <Directory>-Gruppe mit Order, Allow und Deny einstellen, unter welchen Umständen Apache Dateien aus dem jeweiligen Verzeichnis lesen und weitergeben darf.

Zugriffsrechte für Verzeichnisse (Apache 2.2)

Zugriffsregeln gelten auch für alle Unterverzeichnisse, sofern nicht explizit in einer weiteren <Directory>-Gruppe andere Regeln definiert werden. Die Zugriffsregeln für das Verzeichnis / geben daher Standardregeln für das gesamte Dateisystem vor!

- **Order Allow,Deny** bedeutet, dass zuerst alle Allow- und dann alle Deny-Regeln ausgewertet werden. Wenn auf einen Seitenzugriff keine Regel angewendet werden kann, wird der Zugriff blockiert.
- **Order Deny,Allow** dreht die Reihenfolge der Regeln um. Beachten Sie aber: Wenn bei einem Seitenzugriff keine Regel passt, ist der Zugriff erlaubt! Diese Regel gilt in Apache standardmäßig.
- **Allow from** gibt an, von welchen Hostnamen bzw. IP-Adressen Zugriffe erlaubt sind – also beispielsweise Allow from 213.214.215.216 bekannteseite.de. IP-Adressbereiche können Sie in der Form 213.214 oder 213.214.0.0/255.255.0.0 oder 213.214.0.0/16 (für 213.214.*) angeben. Bei Hostnamen gilt site.de auch für www.site.de, sub.site.de etc. Die Regel Allow from all erlaubt jeden Zugriff.
- **Deny from** funktioniert gerade umgekehrt und blockiert den Zugriff für die angegebenen Hosts bzw. Adressen.

Per Default gilt Order Deny,Allow, und mangels anderer Regeln ist somit der gesamte Zugriff auf alle Verzeichnisse blockiert! Wenn Sie Webdateien in anderen Verzeichnissen unterbringen, vergessen Sie nicht, den Zugriff darauf zu erlauben.

Unter Apache 2.4 gelten die drei Schlüsselwörter Order, Allow und Deny als obsolet. Die Schlüsselwörter werden aber weiterhin vom Modul mod_access_compat verarbeitet. Dieses Modul steht bei den meisten Apache-2.4-Installationen zur Verfügung und stellt sicher, dass ein Apache-2.4-Update nicht die gesamte bisherige Konfiguration über den Haufen wirft.

Zugriffsrechte für Verzeichnisse (Apache 2.4)

Bei einer Neukonfiguration wird der Einsatz des neuen Schlüsselworts Require empfohlen. Die folgenden Beispiele zeigen verschiedene Anwendungsformen:

```
# erlaubt den Zugriff vom Rechner mit der IP-Adresse 192.168.0.2
Require ip 192.168.0.2

# erlaubt den Zugriff aus dem Adressbereich 10.0.*.*
Require ip 10.0

# erlaubt den Zugriff aus einem IPv6-Adressbereich
Require ip 2001:1234:789a:0471::/64

# erlaubt den Zugriff für einen bestimmten Hostnamen
Require host intern.meine-firma.de

# erlaubt den Zugriff für *.meine-firma.de
Require host meine-firma.de

# erlaubt den Zugriff von localhost (IPv4 und IPv6)
Require local

# erlaubt den Zugriff für authentifizierte Benutzer
Require valid-user

# erlaubt den Zugriff von überall
Require all granted

# blockiert jeden Zugriff
Require all denied
```

Wenn Sie für ein `<Directory>` mehrere Bedingungen formulieren, dann reicht es, wenn *eine* dieser Bedingungen erfüllt ist:

```
<Directory /var/www/cms>
  Require local
  Require ip 192.168
  Require host meine-firma.de
</Directory>
```

Mit `<RequireAll>` können Sie mehrere Bedingungen durch ein logisches Und kombinieren. Apache liefert die angeforderte Seite nur, wenn *alle* Bedingungen gleichzeitig zutreffen.

```
<Directory /var/www/internal-wiki>
  <RequireAll>
    Require valid-user
    Require ip 192.168.17
  </RequireAll>
</Directory>
```

Wenn Sie Apache zur firmeninternen Kommunikation einrichten, können Sie den Webzugriff auf das lokale Netzwerk beschränken. Wenn das lokale Netzwerk den Adressbereich 192.168.1.* und die lokale Domain .sol nutzt, sieht die richtige Konfiguration für `/var/www` so aus:

```
# für Apache 2.4
<Directory /var/www/>
  Require local
</Directory>
```

Eine alternative Vorgehensweise besteht darin, mit `Listen` die IP-Adresse der lokalen Netzwerkschnittstelle anzugeben. Das setzt voraus, dass die IP-Adresse statisch ist. Nehmen wir an, der Server hat zwei Netzwerkschnittstellen: eine für die Verbindung in das Internet und eine zweite für das LAN mit der IP-Adresse 192.168.1.17. Dann bewirkt `Listen 192.168.1.17`, dass Apache nur noch auf Anfragen aus dem lokalen Netzwerk reagiert. `Listen` gilt allerdings für die gesamte Apache-Konfiguration, nicht nur für einzelne Verzeichnisse oder virtuelle Hosts. `Listen` funktioniert gleichermaßen unter Apache 2.2 und Apache 2.4.

Eine dritte Variante ist die Verwendung einer Firewall: Die Firewall muss den Empfang von Paketen verweigern, die von außen (also aus dem Internet) kommen und an die Ports 80 und 443 (https) gerichtet sind. Die Verwendung einer Firewall ist generell eine gute Idee, weil sie vollkommen unabhängig von Apache funktioniert.

Passwortschutz für Webverzeichnisse

Häufig sollen Webverzeichnisse nur nach einer Authentifizierung durch einen Benutzernamen und das dazugehörige Passwort freigegeben werden. Apache sieht hierfür ein einfaches Verfahren vor, das gleichermaßen in den Versionen 2.2 und 2.4 funktioniert.

Der erste Schritt hin zum Passwortschutz ist eine Passwortdatei. Die Datei sollte aus Sicherheitsgründen außerhalb aller Webverzeichnisse angelegt werden, um einen Zugriff per Webadresse auszuschließen. Das folgende Beispiel geht davon aus, dass die Passwortdatei im Verzeichnis `/var/www-private` gespeichert wird. Wenn Sie ein neues Verzeichnis einrichten, achten Sie darauf, dass Apache hierfür Leserechte hat. Unter CentOS/Fedora/RHEL müssen Sie auch SELinux im Auge haben und das Passwortverzeichnis entweder innerhalb von `/var/www` einrichten oder nach dem Erzeugen des Verzeichnisses den SELinux-Kontext korrekt einstellen.

Um eine neue Passwortdatei anzulegen, verwenden Sie das Kommando `htpasswd` mit der Option `-c` (*create*). Das Passwort wird selbstverständlich verschlüsselt.

Internet-Zugriff blockieren

Zugriffsschutz durch »Listen«

Zugriffsschutz durch eine Firewall

Passwortdatei

```
root# cd /var/www-private
root# htpasswd -c passwords.pwd username
New password: *****
Re-type new password: *****
Adding password for user username
```

Weitere Benutzernamen/Passwort-Paare werden mit `htpasswd` ohne die Option `-c` hinzugefügt:

```
root# cd /var/www-private
root# htpasswd passwords.pwd name2
New password: *****
Re-type new password: *****
Adding password for user username
```

Konfiguration Es gibt nun zwei Varianten, um Apache so zu konfigurieren, dass die Passwortdatei tatsächlich berücksichtigt wird. Die erste Variante setzt voraus, dass Sie die Konfiguration direkt in einer Apache-Konfigurationsdatei durchführen, unter Debian oder Ubuntu also in `/etc/apache2/sites-available/default` für die Standard-Website des Servers bzw. in `.../sitename` für einen virtuellen Host. Bei der zweiten Variante erfolgt die Konfiguration in der Datei `.htaccess`, die sich innerhalb des Webverzeichnisses befindet.

Damit die Passwortdatei von Apache berücksichtigt wird, müssen Sie in die `<Directory>`-Gruppe diverse Authentifizierungsoptionen einfügen. Wenn es für das zu schützende Verzeichnis noch keine eigene `<Directory>`-Gruppe gibt, legen Sie eine neue Gruppe an. Dabei werden automatisch alle Optionen vom übergeordneten Verzeichnis übernommen. Sie müssen also nur die Authentifizierungsoptionen hinzufügen. Die folgenden Zeilen geben hierfür ein Muster:

```
# in /etc/apache2/sites-available/xxx (Debian/Ubuntu)
...
# passwortgeschütztes Verzeichnis
<Directory "/var/www/admin/">
    AuthType      Basic
    AuthUserFile  /var/www-private/passwords.pwd
    AuthName      "admin"
    Require       valid-user
</Directory>
```

Kurz eine Erklärung der Schlüsselwörter:

- **AuthType** gibt den Authentifizierungstyp an. Ich gehe hier nur auf den `Basic`-Typ ein.
- **AuthUserFile** gibt den Ort der Passwortdatei an.
- **AuthName** bezeichnet den Bereich (Realm), für den der Zugriff gültig ist. Der Sinn besteht darin, dass Sie nicht jedes Mal einen Login durchführen müssen, wenn

Sie auf unterschiedliche Verzeichnisse zugreifen möchten, die durch dieselbe Passwortdatei geschützt sind. Sobald Sie sich mit einer bestimmten `AuthName`-Bezeichnung eingeloggt haben, gilt dieser Login auch für alle anderen Verzeichnisse mit diesem `AuthName`.

- **Require valid-user** bedeutet, dass als Login jede gültige Kombination aus Benutzernamen und Passwort erlaubt ist. Alternativ können Sie hier auch angeben, dass ein Login nur für ganz bestimmte Benutzer erlaubt ist:

`Require user name1 name2`

Die oben skizzierte Vorgehensweise ist nur möglich, wenn Sie Zugang zu den Apache-Konfigurationsdateien haben, d. h., wenn Sie selbst der Webadministrator sind. Ist das nicht der Fall, kann eine gleichwertige Absicherung auch durch die Datei `.htaccess` erfolgen, die sich im zu schützenden Verzeichnis befindet. In dieser Datei müssen sich dieselben Anweisungen befinden, die vorhin innerhalb der `<Directory>`-Gruppe angegeben wurden, also `AuthType`, `AuthUserFile`, `AuthName` und `Require`.

.htaccess erfordert AllowOverride AuthConfig

`.htaccess`-Dateien werden nur beachtet, wenn innerhalb des Webverzeichnisses eine Veränderung der Authentifizierungsinformationen zulässig ist. Die (übergeordnete) `<Directory>`-Gruppe muss `AllowOverride AuthConfig` oder `AllowOverride All` enthalten.

32.3 Virtuelle Hosts

Für jede Website (für jeden Host) ein eigener Webserver – das wäre angesichts der Leistungsfähigkeit aktueller Rechner eine Verschwendung von Ressourcen! Mit Apache können Sie dank sogenannter virtueller Hosts viele Websites parallel einrichten. Solange die Gesamtzugriffszahlen nicht an die Limits des Rechners gehen, bemerkt kein Anwender, dass die Websites in Wirklichkeit alle auf demselben Rechner laufen.

Aus technischer Sicht gibt es drei Verfahren, wie Apache entscheidet, an welchen virtuellen Host eine Webanfrage gerichtet ist. Als Ausgangspunkt dient in jedem Fall der vom Browser an den Server übertragene HTTP-Header.

- **Namensbasierte virtuelle Hosts:** Apache erkennt die gewünschte Website anhand des im HTTP-Header enthaltenen Hostnamens. Diese Variante ist am einfachsten zu realisieren und am weitesten verbreitet.
- **IP-basierte virtuelle Hosts:** Apache erkennt die gewünschte Website anhand der IP-Adresse im Header. Diese Vorgehensweise ist mit einem gravierenden Nachteil verbunden: Jeder virtuelle Host erfordert eine eigene IP-Adresse, und IP-Adressen sind für IPv4 Mangelware.

- **Port-basierte virtuelle Hosts:** Apache erkennt aufgrund der Port-Nummer die gewünschte Website. Diese Variante ist in der Praxis unüblich, weil die Port-Nummer als Teil der Webadresse angegeben werden muss. Das sieht unübersichtlich aus und eignet sich bestenfalls für eine technisch versierte Zielgruppe, z. B. für Administratoren.

Ich beziehe mich in diesem Abschnitt wiederum auf die Defaultkonfiguration von Debian bzw. Ubuntu. Dort ist es üblich, für jeden virtuellen Host eine eigene Konfigurationsdatei zu verwenden.

Wenn Sie Ihren Webserver unter Fedora oder RHEL einrichten, kommen naturgemäß dieselben Apache-Schlüsselwörter zum Einsatz. Allerdings erfolgen sämtliche Einstellungen wahlweise in `/etc/httpd/conf/httpd.conf` oder in `/etc/httpd/conf.d/sitename.conf`.

Virtuelle Hosts einrichten

NameVirtualHost (Apache 2.2) Apache 2.4 erkennt bei der Analyse der Konfigurationsdateien automatisch, dass namensbasierte virtuelle Hosts verwendet werden. Das aus Apache 2.2 vertraute Schlüsselwort `NameVirtualHost` ist nicht mehr erforderlich.

Host-Dateien (Debian, Ubuntu) Unter Debian und Ubuntu ist die Standard-Website in `/etc/apache2/sites-available/default` als virtueller Host definiert. Um einen neuen virtuellen Host zu definieren, legen Sie unter Debian oder Ubuntu eine neue Datei im Verzeichnis `/etc/apache2/sites-available/` an. Diese Datei sollte genau eine `<VirtualHost>`-Gruppe enthalten. Die drei folgenden Listings geben je ein Beispiel für einen namens-, IP- und port-basierten Host.

Namensbasierte virtuelle Hosts `ServerName` gibt den Namen des Hosts an. Dieser Hostname muss in den Header-Informationen einer Webanfrage enthalten sein, damit Apache darauf reagiert. Optional können Sie mit `ServerAlias` weitere Namen nennen. Beispielsweise empfiehlt sich zur Einstellung `ServerName www.meinserver.de` die Ergänzung `ServerAlias meinserver.de`, damit ein virtueller Host mit oder ohne die vorangestellten Buchstaben `www.` verwendet werden kann.

```
# /etc/apache2/sites-available/beispiel-named-host (Debian/Ubuntu)
<VirtualHost *:80>
    DocumentRoot /var/www/verzeichnis1/
    ServerName www.firma-1.de
    ServerAlias firma-1.de
    ...
</VirtualHost>
```

Bei IP- und port-basierten Hosts muss die IP-Adresse mit einer der IP-Adressen des Servers übereinstimmen:

```
# /etc/apache2/sites-available/beispiel-IP-host (Debian/Ubuntu)
<VirtualHost 213.214.215.216:80>
    DocumentRoot /var/www/verzeichnis2/
    ServerName www.firma-2.com
    ...
</VirtualHost>

# /etc/apache2/sites-available/beispiel-port-host (Debian/Ubuntu)
<VirtualHost 213.214.215.216:12001>
    DocumentRoot /var/www/verzeichnis3/
    ServerName www.admin-firma3.de
    ...
</VirtualHost>
```

Vergessen Sie nicht, Zusatzports mit »Listen« anzugeben!

Die Adress- und Port-Angaben in `<VirtualHost>` haben keinen Einfluss darauf, welche IP-Adressen und Ports Apache überwacht. Bei den meisten Distributionen sind nur die Ports 80 und 443 (https) vorgesehen (also `Listen 80` und `Listen 443`).

Wenn Apache weitere Ports überwachen soll, müssen Sie die Apache-Konfiguration entsprechend erweitern. Weitere Informationen zur Konfiguration für virtuelle Hosts finden Sie hier:

<https://httpd.apache.org/docs/2.4/de/vhosts>

Um einen virtuellen Host zu aktivieren bzw. später wieder zu deaktivieren, führen Sie nun unter Debian/Ubuntu `a2ensite name` bzw. `a2dissite name` aus und fordern Apache dann zum Neuladen der Konfigurationsdateien auf:

```
root# a2ensite beispiel-named-host (Debian/Ubuntu)
root# systemctl reload apache2
```

Theoretisch ist es möglich, mit `a2dissite` auch die Standard-Website des Servers zu deaktivieren. Das sollten Sie aber nicht tun, weil die Datei `/etc/apache2/sites-available/000-default.conf` diverse Standardeinstellungen für Apache enthält!

Sobald Sie virtuelle Hosts eingerichtet haben, wird die in `sites-available/default` definierte Standard-Website nur noch angezeigt, wenn Webanfragen für keine der virtuellen Hosts zutreffen.

IP- und port-basierte virtuelle Hosts

Virtuelle Hosts aktivieren

Unter CentOS, Fedora, RHEL und SUSE entfallen die Kommandos `a2ensite/a2dissite`, weil sich alle Angaben zu den virtuellen Hosts in der zentralen Konfigurationsdatei `httpd.conf` oder in eigenen Dateien im Verzeichnis `conf.d` befinden. Dort durchgeführte Änderungen aktivieren Sie mit dem folgenden Kommando:

```
root# systemctl reload httpd      (CentOS, Fedora, RHEL)
root# systemctl reload apache2    (SUSE)
```

Beispiel

Dieser Abschnitt beschreibt, wie Sie auf einem Debian- oder Ubuntu-Server den neuen virtuellen Host `firma-123.de` einrichten – zusammen mit einem neuen Login `firma123`, sodass Ihr Kunde, Freund etc. den virtuellen Host selbst administrieren kann. Dabei gehe ich davon aus, dass die Web- und Logdateien des virtuellen Hosts innerhalb des Heimatverzeichnisses des neuen Benutzers `firma123` angeordnet werden. Ebenso gut ist es möglich, zu diesem Zweck ein neues Verzeichnis `/var/www-firma123` einzurichten und dem Benutzer hierfür Schreibrechte zu geben.

Der erste Schritt besteht darin, einen neuen Account einzurichten, ein Passwort zuzuweisen und die erforderlichen Verzeichnisse zu erzeugen. In den folgenden Kommandos müssen Sie natürlich `firma123` durch den tatsächlichen Benutzernamen ersetzen!

```
root# adduser firma123
root# passwd firma123
Enter new UNIX password: *****
Retype new UNIX password: *****
passwd: password updated successfully
root# mkdir ~firma123/www
root# chown firma123:firma123 ~firma123/www
root# mkdir ~firma123/www-log
root# chown root:root ~firma123/www-log
root# chmod go-w ~firma123/www-log
```

Im zweiten Schritt erzeugen Sie eine neue Datei im Verzeichnis `sites-available`, die so ähnlich wie das folgende Muster aufgebaut ist. Abermals müssen Sie `firma123` durch den tatsächlichen Benutzernamen ersetzen und außerdem statt `firma-123.de` den tatsächlichen Hostnamen angeben. Mit `AllowOverride AuthConfig File` geben Sie Ihrem Kunden relativ weitreichende Möglichkeiten, die Konfiguration der Website durch eine `.htaccess`-Datei anzupassen. Wenn Sie das nicht möchten, müssen Sie diverse Konfigurationsdetails absprechen und `fix` einstellen.

```
# /etc/apache2/sites-available/firma-123.de
<VirtualHost * >
    DocumentRoot /home/firma123/www/
    ServerName firma-123.de
```

```
ServerAlias www.firma-123.de
ErrorLog /home/firma123/www-log/error.log
CustomLog /home/firma123/www-log/access.log combined
ServerAdmin webmaster@firma-123.de
ErrorDocument 404 /not-found.html
<Directory "/home/firma123/www/" >
    AllowOverride AuthConfig File
</Directory>
</VirtualHost>
```

Zur Aktivierung der Website führen Sie die folgenden Kommandos aus:

```
root# a2ensite firma-123.de
root# systemctl reload apache2
```

Ihr Kunde muss nun nur noch die DNS-Konfiguration seiner Domain anpassen: Die zugeordnete IP-Adresse muss mit der Ihres Servers übereinstimmen. Sobald das der Fall ist, beantwortet Ihr Webserver alle Anfragen, die an `www.firma-123.de` gerichtet sind.

Wie ich bereits erwähnt habe, läuft Apache aus Sicherheitsgründen nicht mit `root`-Rechten, sondern unter einem Account mit eingeschränkten Rechten (`www-data` bei Debian/Ubuntu, `apache` bei Fedora/RHEL bzw. `wwwrun` bei SUSE). Stellen Sie die Zugriffsrechte der Webdateien so ein, dass Apache sie lesen kann!

Zugriffsrechte

Wenn Apache einzelne Dateien auch verändern soll (z. B. über ein PHP-Script), ordnen Sie den Verzeichnissen und Dateien die Gruppe `www-data/apache/wwwrun` zu und geben den Gruppenmitgliedern Schreibrechte (`chmod g+w`). Unter Fedora und RHEL müssen Sie außerdem den SELinux-Kontext korrekt einstellen.

Im obigen Beispiel werden alle Fehler- und Zugriffsmeldungen in eigenen Dateien für den virtuellen Host gespeichert. Diese Vorgehensweise erleichtert die Auswertung der Logging-Dateien. Allerdings ist die Anzahl der offenen Datei-Handles für Apache (wie für jeden anderen Linux-Prozess) beschränkt. Wenn Sie sehr viele virtuelle Hosts einrichten, müssen Sie alle Zugriffe in einer zentralen Datei protokollieren und diese Datei dann durch ein anderes Programm in kleinere Dateien je nach Host zerlegen. Weitere Informationen zu diesem Thema finden Sie hier:

Logging

<https://httpd.apache.org/docs/2.4/vhosts/fd-limits.html>

Virtuelle Hosts setzen voraus, dass die DNS-Konfiguration stimmt! Um die im vorigen Abschnitt beschriebene Website `firma-123.de` zu testen, muss der DNS-Eintrag der Domain `firma-123.de` auf die IP-Adresse Ihres Webserver zeigen. Änderungen am DNS-Eintrag kann nur der Eigentümer der Domain durchführen. Die meisten Domain-Händler bieten dazu entsprechende Werkzeuge an. Beachten Sie, dass DNS-Änderungen nicht sofort gelten. Die Synchronisation der vielen, weltweit verteilten Nameserver kann etliche Stunden dauern, auch wenn es oft schneller geht.

Test

Bei Serverumbauten oder -umzügen besteht oft der Wunsch, den neuen Server zuerst in Ruhe zu testen, bevor die DNS-Änderung tatsächlich durchgeführt wird. Der einfachste Weg besteht darin, den neuen virtuellen Host anfänglich nicht namensbasiert, sondern port-basiert zu konfigurieren. Dazu entfernen Sie die `ServerName`- und `ServerAlias`-Anweisungen und geben im `<VirtualHost>`-Tag statt des Sterns die IP-Adresse des Servers sowie eine freie Port-Nummer an, beispielsweise so:

```
<VirtualHost 213.214.215.216:12001>
```

Standardmäßig verarbeitet Apache nur Anfragen, die an die Ports 80 und 443 (für https) gerichtet sind. Damit Apache auch den hier eingesetzten Port 12001 berücksichtigt, müssen Sie in `/etc/apache2/ports.conf` eine weitere Zeile mit `Listen 12001` einfügen. Nun ist noch das Kommando `systemctl reload apache2` erforderlich, damit Apache die veränderte Konfiguration berücksichtigt. Jetzt können Sie den neuen Web-auftritt mit Ihrem Webbrowser testen, indem Sie die IP-Adresse des Servers samt der Port-Nummer 12001 angeben, also beispielsweise `http://213.214.215.216:12001`.

`/etc/hosts` zum
Testen ändern

Ganz anders können Sie vorgehen, wenn sich bei einem Server-Umzug inklusive Wechsel auch die IP-Adresse ändert: In diesem Fall können Sie in Ruhe den neuen Server einrichten. Für Ihre Kunden ist der neue Server noch nicht sichtbar, weil Ihr DNS-Eintrag ja noch auf den alten Server verweist. Um den neuen Server aber schon jetzt selbst unter dem richtigen Hostnamen zu testen, können Sie auf Ihrem lokalen Linux-Rechner zu Hause (also nicht auf dem Server mit Apache!) vorübergehend den folgenden Eintrag in `/etc/hosts` vornehmen.

Nehmen wir an, für `firma-123.de` soll ein neuer Webauftritt erstellt werden. Momen-tan zeigt der DNS-Eintrag der Firma noch auf den alten Server, z. B. auf die IP-Adresse `234.234.236.237`. Als Administrator haben Sie mittlerweile den neuen Server eingerich-tet, der die IP-Adresse `123.124.125.126` hat. Jetzt möchten Sie testen, ob alles funktio-niert. Also verändern Sie vorübergehend auf Ihrem lokalen Rechner `/etc/hosts`. Die dort durchgeführte Einstellung hat Vorrang vor allen Nameservern!

```
# /etc/hosts auf einem lokalen Rechner (NICHT auf dem Server)
...
# neue IP-Adr.      # vorhandener Name
123.124.125.126    firma-123.de www.firma-123.de
```

Der entscheidende Vorteil dieser Variante im Vergleich zur vorhin skizzierten Vor-gehensweise mit einem eigenen Port besteht darin, dass beim Test auch alle Links funktionieren. Wenn Sie im Testbetrieb den Link `http://www.firma-123.de/cms/seite-xy.html` anklicken, wird auch die neue Seite wieder korrekt in Ihrem Webbrowser angezeigt.

32.4 Verschlüsselte Verbindungen (HTTPS)

In der Standardkonfiguration verwendet Apache das Protokoll HTTP. Es überträgt alle Daten unverschlüsselt. Für lokale Testinstallationen ist das in Ordnung, aber für im Internet erreichbare Webseiten ist HTTP nicht mehr zeitgemäß. Selbst für Webseiten, die keine Benutzerdaten entgegennehmen, wird heute HTTPS empfohlen – und sei es nur, um das Ranking in den Suchergebnissen zu verbessern. Sobald Ihre Webseite in Formularen oder auf anderen Wegen persönliche Daten (Passwörter, Kreditkarten-nummern etc.) entgegennimmt, sollte eine HTTPS-Konfiguration selbstverständlich sein. Immer mehr Webbrowser zeigen andernfalls unmissverständliche Warnungen an, die darauf hinweisen, dass die unverschlüsselte Übertragung der Daten unsicher ist.

HTTPS vereint die Protokolle *Hypertext Transfer Protocol* (HTTP) mit *Secure Sockets Layer* (SSL) und fügt HTTP so Verschlüsselungsfunktionen hinzu. In diesem Abschnitt erkläre ich Ihnen, wie Sie Apache für HTTPS-Verbindungen konfigurieren. Details zur Verwendung der kostenlosen Zertifikate der Organisation *Let's Encrypt* folgen in Abschnitt 32.5.

In der Vergangenheit war es unmöglich, HTTPS mit virtuellen Hosts zu kombinieren, die sich nur durch den Hostnamen unterscheiden. Das Problem bestand darin, dass auch der Hostname selbst verschlüsselt übermittelt wurde. Für den Webserver war es damit unmöglich, das richtige Zertifikat zu »erraten«.

Namensbasierte
virtuelle Hosts
und HTTPS

Mittlerweile senden Webbrowser den Hostnamen beim Verbindungsaufbau vorweg einmal unverschlüsselt. Damit weiß Apache, welches Zertifikat er für die Verschlüsse-lung verwenden soll. Damit das funktioniert, muss

```
SSLStrictSNIVHostCheck off
```

gelten, was bei aktuellen Apache-Versionen standardmäßig der Fall ist. Weitere Details zur korrekten Konfiguration mehrerer Hosts, die jeweils ihr eigenes HTTPS-Zertifikat verwenden, finden Sie hier:

```
https://wiki.apache.org/httpd/NameBasedSSLVHostsWithSNI
```

Zertifikate

Bevor Sie nun mit den Konfigurationsarbeiten beginnen, brauchen Sie noch ein Server-Zertifikat. Und an dieser Stelle muss ich etwas ausholen ...

Die Verschlüsselung der Daten erfolgt auf der Basis asymmetrischer Verschlüsse-lungsverfahren. Die Grundidee besteht darin, dass es ein Schlüsselpaar gibt, das aus einem privaten (geheimen) und einem öffentlichen Schlüssel besteht. Der öffentliche

Grundlagen

Schlüssel eignet sich nur zum *Verschlüsseln* von Daten. Zum *Entschlüsseln* ist der private Schlüssel erforderlich. Auf die Details dieser Verfahren gehe ich hier nicht ein – sie wurden schon unzählige Male beschrieben und erklärt, unter anderem auch in der Wikipedia.

Beim Verbindungsaufbau zwischen dem Client (also einem Webbrowser) und dem Server (Apache) wird zuerst auf der Basis einer Zufallszahl vom Client und des öffentlichen Schlüssels vom Server ein gemeinsamer Schlüssel ausgehandelt (Handshake-Verfahren). Dieser *Session Key* wird dann zur Verschlüsselung der gesamten weiteren Kommunikation eingesetzt. Der Webbrowser ist damit in der Lage, die zu sendenden Daten so zu verschlüsseln, dass nur der Webserver diese mit seinem privaten Schlüssel auswerten kann.

Der nach heutigem Wissensstand nahezu abhörsichere Datenaustausch ist aber nur *ein* Punkt zur Verbesserung der Sicherheit. Ein zweiter Punkt besteht darin, dass der Anwender Gewissheit haben muss, dass er mit dem richtigen Partner kommuniziert. Was nützt es, wenn die Daten für das Online-Banking zwar abhörsicher übertragen werden, aber statt zur Bank direkt in die Hände eines Betrügers gelangen?

Aus diesem Grund können Zertifikate auch Daten über die Website sowie eine Art Unterschrift einer Zertifizierungseinrichtung enthalten. Deren Aufgabe ist es, die Identität des Zertifikatbewerbers anhand einer Passkopie, eines Gewerbescheins etc. zu überprüfen. Dieser Kontrollprozess macht derartige Zertifikate leider relativ teuer.

Wie vertrauenswürdig ein Zertifikat ist, hängt von der Vertrauenswürdigkeit der Authentifizierungsstelle ab. Bekannte Webbrowser wie Firefox oder Internet Explorer akzeptieren nur Zertifikate, die von etablierten Authentifizierungseinrichtungen ausgestellt wurden (z. B. Verisign oder Thawte). Bei anderen Zertifikaten werden unübersehbare Warnungen angezeigt. Mit etwas Hartnäckigkeit kann man den Webbrowser zwar dennoch dazu bringen, auch unsichere Zertifikate zu akzeptieren, ein florierendes Online-Geschäft ist auf dieser Basis aber unmöglich. Mit anderen Worten: Für ernsthafte Geschäftsanwendungen ist ein autorisiertes Zertifikat unabdingbar.

Dateikennungen

Tabelle 32.5 fasst die üblichen Dateikennungen für Schlüssel- und Zertifikatsdateien zusammen. Dabei steht pem für *Privacy Enhanced Mail*. Das ist ein Verfahren für verschlüsselte E-Mails, das sich leider nicht durchgesetzt hat. Das dort definierte PEM-Format ist aber bis heute üblich. *.crt-Dateien enthalten zumeist ebenfalls pem-Dateien. Dank der abweichenden Dateikennung wird die Datei vom Internet Explorer als Zertifikat erkannt.

Kennung	Bedeutung
*.key	Private-Key-Datei
*.csr	unsigniertes Zertifikat (Certificate Signing Request)
*.pem	Container-Datei für ein signiertes Zertifikat oder eine ganze Zertifikatskette
*.crt	*.pem-Datei mit anderer Kennung (Windows)

Tabelle 32.5 Dateikennungen für Schlüssel- und Zertifikatsdateien

Selbst signierte Zertifikate erstellen

Nachdem ich Sie gerade zu überzeugen versucht habe, ein »richtiges« Zertifikat zu erwerben, erkläre ich Ihnen jetzt, wie Sie ein Zertifikat selbst erstellen können. Es gibt gute Gründe für diesen scheinbaren Sinneswandel: Für erste Experimente reicht ein selbst erstelltes Zertifikat vollkommen aus. Außerdem lässt sich ein eigenes Zertifikat in wenigen Minuten erstellen, während die Erteilung eines autorisierten Zertifikats erfahrungsgemäß tage-, wenn nicht wochenlang dauert. Diese Wartezeit nutzen Sie am besten, indem Sie sich mit den wichtigsten Stolperfallen vertraut machen. Wenn grundsätzlich alles funktioniert, können Sie Ihr eigenes Zertifikat mühelos durch ein »richtiges« ersetzen.

Bei vielen Distributionen sind selbst signierte Zertifikate sogar standardmäßig vorhanden. Unter Debian und Ubuntu werden sie »Snakeoil-Zertifikate« genannt (/etc/ssl/certs/ssl-cert-snakeoil.pem). CentOS, Fedora und RHEL haben stattdessen ein selbst signiertes Zertifikat für den Hostnamen des Rechners (/etc/pki/tls/certs/localhost.crt).

Als Erstes installieren Sie das Paket openssl. Es enthält das gleichnamige Kommando zum Erzeugen, Verwalten und Manipulieren von Schlüsseln und Zertifikaten.

openssl

```
root# apt/dnf/yum/zypper install openssl
```

Das folgende Kommando erzeugt eine Datei mit einem privaten 2048-Bit-RSA-Schlüssel. Diesen Schlüssel werden Sie zweimal benötigen: einmal zum Erzeugen einer Zertifikatsanfrage (*Certificate Signing Request*) und dann nochmals zur Signierung.

Zuerst der Schlüssel

```
root# openssl genpkey -algorithm RSA -pkeyopt rsa_keygen_bits:2048 \
-out server.key
```

Achten Sie darauf, dass nur root diese Datei lesen kann! Wenn diese Datei in fremde Hände gerät, ist Ihr Serverzertifikat wertlos, und Sie müssen es widerrufen!

```
root# chmod 400 server.key
```

Verschlüsselte Schlüssel

Schlüssel sind wertvoll – deswegen werden sie normalerweise selbst mit einer *Passphrase*, also mit einem besonders langen Passwort verschlüsselt. Im Englischen spricht man hier deutlich klarer von einem *encrypted key*. Wenn Sie das möchten, ergänzen Sie das obige openssl-Kommando um die Option `-aes-256-cbc`. Bei den weiteren Kommandos müssen Sie dann jedes Mal, wenn Sie den Schlüssel nutzen, wiederum dieses Passwort angeben.

Das gilt auch für Apache: Der Webserver braucht den Schlüssel, um das Zertifikat auszulesen. Deswegen fragt Apache nun bei jedem (Neu-)Start nach dem Passwort des Schlüssels. Ein automatisierter Neustart, z. B. bei einem Update, wird so unmöglich.

Um diesem Problem zu entgehen, wird für Apache eine unverschlüsselte Kopie des Schlüssels erzeugt (`openssl rsa -in server.key -out server-unencrypted.key`). Apache wird nun so konfiguriert, dass er anstelle des verschlüsselten Schlüssels die unsichere Schlüsseldatei verwendet. Diese muss also im Dateisystem des Servers gespeichert werden. Und spätestens jetzt wird klar, dass ein verschlüsselter Schlüssel in unserem Fall nur ein unnötiger Mehraufwand ist, der die Sicherheit nicht steigert. Das heißt aber natürlich nicht, dass verschlüsselte Schlüssel generell nicht zu empfehlen wären – ganz im Gegenteil!

... dann das Zertifikat

Schon etwas mehr Arbeit macht es, das Zertifikat zu erstellen. Genau genommen erzeugt das folgende Kommando nicht das endgültige Zertifikat, sondern einen *Certificate Signing Request*, also eine Anfrage zur Signierung des Zertifikats. In das Zertifikat fließt auch der Schlüssel `server.key` ein (Option `-key`).

Bei der Ausführung des Kommandos müssen Sie angeben, in welchem Land und in welchem Ort Sie wohnen, wie Sie heißen etc. Entscheidend ist die Frage nach dem *Common Name*: Hier ist nicht Ihr Name gefragt, sondern der exakte Name Ihrer Website in der Form, in der er für verschlüsselte Verbindungen verwendet wird. Manche Websites verwenden für verschlüsselte Verbindungen eine eigene Subdomain (z. B. `banking.ing-diba.de`), andere nicht (z. B. `www.amazon.de`). Wie auch immer, das Zertifikat gilt nur für eine bestimmte Schreibweise. Sie können also beispielsweise ein Zertifikat für `www.firma-abc.de` nicht auch für `firma-abc.de` verwenden (oder umgekehrt)! Achten Sie darauf, dass Sie das Challenge-Passwort leer lassen, die entsprechende Frage also mit der Eingabe eines Punktes beantworten.

```
root# openssl req -new -sha256 -key server.key -out server.csr
Enter pass phrase for server.key: *****
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN. There are quite a few fields but you can leave
some blank. For some fields there will be a default value,
```

```
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:      DE
State or Province Name (full name) [...]:
Locality Name (eg, city) []:            Berlin
Organization Name (eg, company) [Sample Ltd]:  Max Muster
Organizational Unit Name (eg, section) []:
Common Name (eg server FQDN or YOUR name) []: www.firma-abc.de
Email Address []:                        webmaster@firma-abc.de
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Mit dem nächsten Kommando unterschreiben Sie Ihr Zertifikat selbst. Bei einem »richtigen« Zertifikat erfolgt dieser Vorgang – natürlich nach einer Kontrolle der von Ihnen vorgelegten Dokumente – durch die Authentifizierungseinrichtung. Zur Unterschrift wird dann der Schlüssel der Authentifizierungsstelle verwendet.

... und zuletzt die Signatur

Standardmäßig gilt das fertige Zertifikat nur für 30 Tage. Die Option `-days 1900` verlängert den Gültigkeitszeitraum auf circa fünf Jahre:

```
root# openssl x509 -req -days 1900 -in server.csr \
      -signkey server.key -sha256 -out server.pem
Signature ok
subject=/C=DE/L=Berlin/O=Max Muster/CN=www.firma-abc.de/
emailAddress=webmaster@firma-abc.de
Getting Private key
```

unable to write random state

Mitunter tritt beim Ausführen des obigen Kommandos der Fehler *unable to write random state* auf. Der Grund dafür besteht zumeist darin, dass openssl das Heimatverzeichnis nicht findet – oft deswegen, weil Sie zuvor `sudo -s` ausgeführt haben. Abhilfe schafft `export PATH=/root`.

Physikalisch gesehen, handelt es sich bei den erzeugten Schlüsseln und Zertifikaten um relativ kleine Textdateien. Um die in einem Zertifikat enthaltenen Daten im Klartext anzuzeigen, verwenden Sie das Kommando `openssl x509 -text`. Die folgenden Ausgaben sind aus Platzgründen gekürzt:

Kontrolle

```
root# ls
server.key  (unverschlüsselter privater Schlüssel)
server.csr  (Zertifikat ohne Unterschrift)
server.crt  (Zertifikat mit Unterschrift)
```



```
root# cat server.pem
-----BEGIN CERTIFICATE-----
MIICWTCCAcICQCL6ExhrQiELDANBgqhkiG9wOBAQUFADBxMQswCQYDVQQGEwJB ...
-----END CERTIFICATE-----

root# openssl x509 -text -in server.pem
Certificate:
  Data:
    Version: 1 (0x0)
    Serial Number: 12669601459972319941 (0xafd37766c36baac5)
    Signature Algorithm: sha256WithRSAEncryption
    Issuer: C=DE, L=Berlin, O=Max Muster,
             CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de
    Validity
      Not Before: Sep 28 14:48:03 2015 GMT
      Not After : Dec 10 14:48:03 2020 GMT
    Subject: C=DE, L=Berlin, O=Max Muster,
             CN=www.firma-abc.de/emailAddress=webmaster@firma-abc.de
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (2048 bit)
      Modulus:
        00:be:37:21:23:b6:13:e4:92:74:36:9f:de:4e:9a:
      Signature Algorithm: sha256WithRSAEncryption
        43:b8:92:82:0d:f6:e2:ef:ff:07:eb:3a:1f:da:3d:d9:ba:53:
        d7:1f:4a:49:ec:5a:c1:fb:0f:95:e8:94:89:ab:7b:05:95:62:
```

Apache-Konfiguration für den HTTPS-Betrieb

mod_ssl Die für das Protokoll HTTPS erforderlichen Apache-Funktionen befinden sich im Modul `mod_ssl`. Unter Debian oder Ubuntu ist dieses Modul standardmäßig installiert und muss nur aktiviert werden:

```
root# a2enmod ssl
root# systemctl restart apache2
```

Unter Fedora bzw. CentOS/RHEL müssen Sie das SSL-Modul zuerst installieren:

```
root# dnf/yum install mod_ssl
root# systemctl restart httpd
```

SSL-Konfiguration Apache muss die Schlüssel- und Zertifikatsdatei lesen – daher liegt es nahe, die beiden Dateien in das Apache-Konfigurationsverzeichnis zu kopieren:

```
root# cp server.pem server.crt /etc/apache2 (Debian, SUSE, Ubuntu)
root# cp server.pem server.crt /etc/httpd (CentOS, Fedora, RHEL)
```

Als Nächstes müssen Sie `httpd.conf` (Fedora, RHEL) um einen `VirtualHost`-Eintrag erweitern bzw. die entsprechenden Zeilen in eine neue Datei in `/etc/apache2/sites-available` einfügen. Bei Debian und Ubuntu wird eine entsprechende Musterdatei gleich mitgeliefert (`default-ssl`). Sie können diese Datei als Ausgangsbasis für eine eigene Site-Datei verwenden, der Sie zur besseren Unterscheidbarkeit von anderen Site-Dateien `ssl` oder `https` voranstellen, also beispielsweise `ssl.firma-abc.de`.

Die folgenden Zeilen zeigen eine minimale Konfiguration, bei der parallel zur Default-website (HTTP) eine HTTPS-Seite eingerichtet wird. Für beide Seiten kommt dieselbe IP-Adresse zum Einsatz. Die Unterscheidung erfolgt durch die in der `VirtualHost`-Zeile eingestellte Port-Nummer 443.

`SSLEngine on` aktiviert die Verschlüsselungsfunktionen. `SSLxxxFile` gibt an, wo sich die Dateien mit dem Zertifikat und dem privaten Schlüssel befinden. `SSLProtocol` und `SSLCipherSuite` bestimmen, welche Version des SSL-Protokolls bzw. welcher Mechanismus zur Erzeugung des gemeinsamen Session Keys eingesetzt werden soll. In der Regel tauschen Apache und der Webbrowser Informationen darüber aus, welche Protokolle sie jeweils unterstützen, und verwenden dann das sicherste Verfahren, das beide beherrschen. Nur wenn es gute Gründe dafür gibt – etwa, weil Sie bestimmte ältere Protokolle/Verfahren nicht akzeptieren möchten –, sollten Sie hier explizite Vorgaben machen.

```
# z.B. in /etc/httpd/conf.d/ssl.conf (CentOS, Fedora, RHEL)
# oder in /etc/apache2/sites-available/ssl-firma-abc.conf (Debian, Ubuntu)
<VirtualHost _default_:443>
  ServerName      www.firma-abc.de
  DocumentRoot    /var/www/
  SSLEngine       on
  SSLCertificateFile /etc/apache2/server.pem
  SSLCertificateKeyFile /etc/apache2/server.key
  <Directory /var/www/>
    AllowOverride None
    Require all granted
  </Directory>
</VirtualHost>
```

Zur Aktivierung der HTTPS-Site müssen Sie Apache dazu auffordern, die Konfiguration neu einzulesen. Falls Sie die HTTPS-Datei unter Debian/Ubuntu in einer eigenen Konfigurationsdatei in `/etc/apache2/sites-available` durchgeführt haben, müssen Sie diese Datei aktivieren:

```
root# systemctl restart httpd (CentOS/Fedora/RHEL)
root# a2ensite ssl-firma-abc (Debian/Ubuntu, Teil 1)
root# systemctl restart apache2 (Debian/Ubuntu, Teil 2)
```

Wenn Sie erstmals von einem Betrieb ohne SSL auf einen Betrieb mit SSL umstellen, reicht ein Neueinlesen der Konfigurationsdateien nicht aus. Damit das SSL-Modul geladen wird, müssen Sie `restart` angeben, nicht `reload`!

Sicherheits-
warnung

Wenn ein Webbrowser auf ein selbst signiertes Zertifikat oder ein Snakeoil-Zertifikat stößt, zeigt der Browser eine Sicherheitswarnung wie in Abbildung 32.2 an. Die Warnung bezieht sich nicht darauf, dass die Verschlüsselung nicht sicher wäre – das ist sie! Vielmehr warnt der Webbrowser, weil die Identität dessen, der das Zertifikat unterzeichnet hat, unbekannt ist. Für technisch unbedarfte Anwender ist das eine Feinheit; sie werden die Webseite als unsicher betrachten. Wenn es Ihnen aber nur darum geht, dass Sie selbst phpMyAdmin sicher verwenden können, um die MySQL-Installation auf dem Server zu administrieren, dann reicht dieses Zertifikat vollkommen aus!

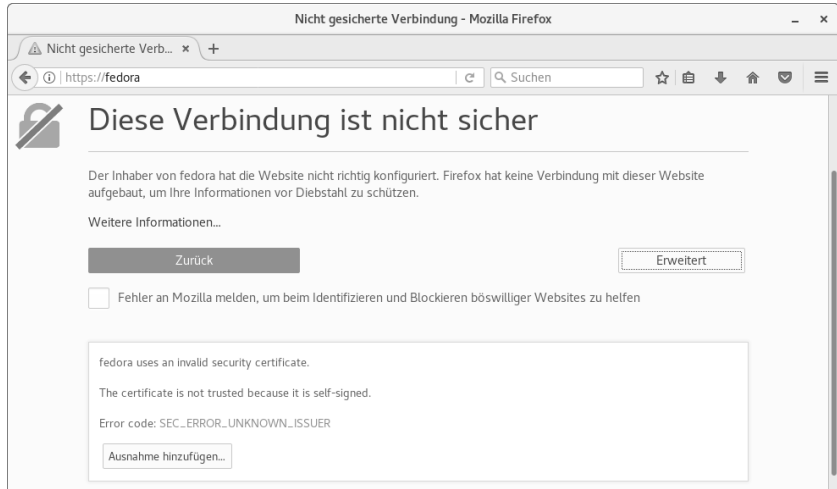


Abbildung 32.2 Warnung vor einem selbst signierten Zertifikat

Snakeoil-Zertifikate

Debian und
Ubuntu

Bei Debian und Ubuntu werden bei der Installation von Apache automatisch ein Snakeoil-Schlüssel und ein zehn Jahre lang gültiges Snakeoil-Zertifikat erzeugt:

```
/etc/ssl/certs/ssl-cert-snakeoil.pem    (Zertifikat)
/etc/ssl/private/ssl-cert-snakeoil.key   (Schlüssel)
```

»Snakeoil« wird im Englischen als Bezeichnung für vorgebliche Wunder- oder Allheilmittel verwendet. Das Zertifikat wird erzeugt, damit Web- und Mail-Server ohne das umständliche Erzeugen eigener Zertifikate sofort verschlüsselt verwendet werden können. Die Apache-Konfigurationsdatei `default-ssl.conf` enthält dementsprechend die folgenden Zeilen:

```
# Datei /etc/apache2/sites-available/default-ssl.conf
...
SSLCertificateFile    /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

Natürlich gelten für das Snakeoil-Zertifikat dieselben Einschränkungen wie bei Zertifikaten, die Sie mit `openssl` selbst erzeugt und signiert haben – daher auch der Name. Das Snakeoil-Zertifikat berücksichtigt den bei der Erstellung gültigen Rechnernamen (`/etc/hostname`). Wenn Sie nach der Änderung eines Hostnames ein neues Zertifikat benötigen, rufen Sie das folgende Kommando auf:

```
root# make-ssl-cert generate-default-snakeoil --force-overwrite
```

`make-ssl-cert` ist ein relativ simples Script, das auf das vorhin beschriebene `openssl`-Kommando zurückgreift.

Bei Distributionen aus der Red-Hat-Familie gibt es ähnliche, automatisch erstellte und selbst signierte Defaultzertifikate:

CentOS, Fedora
und RHEL

```
# Datei /etc/httpd/conf.d/ssl.conf
...
SSLCertificateFile    /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
```

Die Zertifikate sind ein Jahr für den Hostnamen gültig, der während der Installation des `httpd`-Pakets eingestellt war.

Zertifikate etablierter Zertifizierungsanbieter verwenden

Wenn Sie sich entscheiden, ein Zertifikat bei Thwate, Verisign etc. zu erwerben, ersparen Sie sich naturgemäß das Erzeugen der Schlüssel- und Zertifikatsdateien. Sie erhalten diese Dateien vom Zertifikatsanbieter und kopieren Sie in ein Verzeichnis, auf das Apache Zugriff hat, z. B. `/etc/apache2`, `/etc/httpd`, `/etc/pki` oder `/etc/ssl`. Verwenden Sie aber auf keinen Fall ein Verzeichnis mit Webdateien, damit der Webserver den privaten Schlüssel nicht unbeabsichtigt öffentlich macht!

Die Pfade zum Zertifikat und zum Schlüssel geben Sie wiederum durch die Schlüsselwörter `SSLCertificateFile` und `SSLCertificateKeyFile` an. Außerdem müssen Sie in vielen Fällen an den Browser Zusatzinformationen dazu übergeben, wie er Ihre Zertifikate überprüfen kann. Genau genommen geht es hier um Informationen darüber, welche anerkannte Zertifizierungsstelle wiederum Ihrer Zertifizierungsstelle vertraut. Der Browser muss in der Lage sein, eine Vertrauenskette bis zu einer Zertifizierungsstelle herzustellen, die ihm bekannt ist.

SSLCertificate-
ChainFile

Vergessen Sie diese Zusatzfunktionen, beklagt sich der Webbrowser beim Besuch Ihrer Seite, dass der Verbindung nicht vertraut wird, weil keine Zertifikatsaussteller-

kette angegeben wurde. Abhilfe schaffen die Schlüsselwörter `SSLCertificateChainFile` und `SSLCACertificateFile`, mit denen Sie an Apache Zertifikate Ihrer Zertifizierungsstelle übergeben (Certification Authority, daher die Abkürzung CA). Die erforderlichen Zertifikatsdateien stellt Ihnen Ihre Zertifizierungsstelle zum Download zur Verfügung. Nachdem Sie die Dateien so auf Ihrem Rechner eingerichtet haben, dass Apache sie lesen kann, ändern Sie die Konfiguration wie folgt und führen dann `systemctl reload apache2/httpd` aus.

```
...
SSLCertificateFile      /etc/apache2/server.pem
SSLCertificateKeyFile   /etc/apache2/server.pem
SSLCertificateChainFile /etc/apache2/sub.class1.server.ca.pem
SSLCACertificateFile   /etc/apache2/ca.pem
...
```

Diese Zusatzinformationen ermöglichen es dem Webbrowser, die Korrektheit der von Ihnen benutzten Zertifikate zu überprüfen.

SSL-Einstellungen

Mit dem Einrichten von Zertifikaten ist es leider nicht getan. Es gibt unzählige Verschlüsselungsverfahren und -versionen, die alle unter dem Begriff HTTPS zusammengefasst werden. Manche von ihnen sind jedoch nicht mehr sicher. Deswegen müssen Sie mit `SSLCipherSuite` und `SSLProtocol` explizit angeben, welche Verfahren Ihre Webseite unterstützt und welche sie ablehnt. Eine große Hilfe bei der optimalen HTTPS-Konfiguration ist die folgende Webseite (siehe Abbildung 32.3):

`https://www.ssllabs.com/ssltest`

Dort können Sie die Adresse Ihrer Webseite angeben. Ein Script überprüft dann die Sicherheit Ihrer Webseite und gibt Optimierungstipps. Dazu gleich ein konkretes Beispiel: Meine Webseite `https://kofler.info` läuft auf einem Ubuntu-Server mit Apache 2.4. Für HTTPS verwende ich ein Zertifikat von Let's Encrypt (siehe den folgenden Abschnitt). Die relevanten Apache-Konfigurationszeilen sehen wie folgt aus:

```
SSLEngine on

# Zertifikate von Let's Encrypt
SSLCertificateFile      /etc/letsencrypt/live/kofler.info/cert.pem
SSLCertificateKeyFile   /etc/letsencrypt/live/kofler.info/privkey.pem
SSLCertificateChainFile /etc/letsencrypt/live/kofler.info/chain.pem

# »CRIME«-Attacke, siehe
# https://raymii.org/s/tutorials/Strong_SSL_Security_On_Apache2.html
SSLCompression off
SSLCipherSuite AES128+EECDH:AES128+EDH
```

```
# »Poodle«-Problem, siehe https://access.redhat.com/solutions/1232413
SSLProtocol All -SSLv2 -SSLv3
```

```
# HTTP Strict Transport Security (HSTS) aktivieren
# siehe https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security
Header always set Strict-Transport-Security \
    "max-age=63072000; includeSubdomains; preload"
```

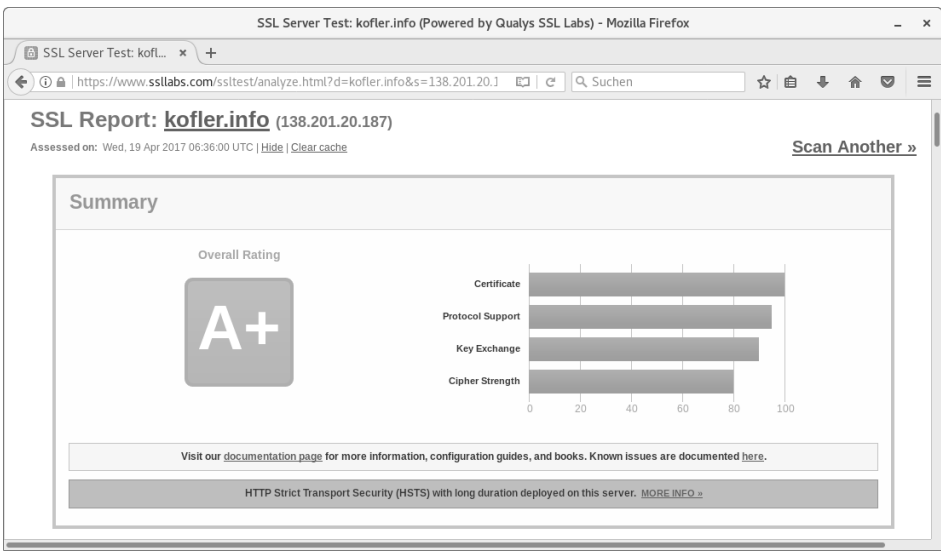


Abbildung 32.3 Online-Überprüfung der HTTPS-Konfiguration

32.5 Let's Encrypt

Seit Anfang 2016 bietet die Organisation *Let's Encrypt* (`https://letsencrypt.org`) kostenlose Zertifikate an. Mittlerweile sind diese Zertifikate sehr weit verbreitet: Anfang 2017 vertrauten bereits mehr als 20 Millionen Websites auf Let's Encrypt. Die naheliegende Frage lautet daher: »Warum noch Geld für Zertifikate anderer Anbieter ausgeben?«

Let's Encrypt ist mit zwei wesentlichen Einschränkungen verbunden: Zum einen sind die Zertifikate nur 90 Tage gültig. Um diesen Nachteil zu umgehen, muss das System so eingerichtet werden, dass die Zertifikate regelmäßig automatisch aktualisiert werden. (Das ist nicht schwierig.)

Zum anderen handelt es sich bei den Zertifikaten von Let's Encrypt um reine Domänenzertifikate: Sier ermöglichen die Kontrolle, dass eine Webseite tatsächlich von der Domäne *firma-abc.de* kommt und nicht von einer anderen Seite. Let's Encrypt führt aber keinerlei Kontrolle durch, wem diese Domäne gehört (*Organization Validation*),

geschweige denn, wer diese Person bzw. Organisation ist (*Extended Validation* durch Überprüfung einer Passkopie, eines Firmenbuchauszugs etc.). Mit anderen Worten: Auch Betrüger können kostenlose Zertifikate von Let's Encrypt verwenden.

Zertifikate von Let's Encrypt sind mit der billigsten und gleichzeitig am weitesten verbreiteten Zertifikatsvariante der etablierten Zertifizierungsstellen vergleichbar. Die Zertifikate reichen aus, um ein eigenes Blog oder ähnliche Seiten so zu verschlüsseln, dass der Webbrowser ein grünes Schloss oder ein anderes Sicherheitssymbol anzeigt. Firmen bzw. Organisationen, die auf ihren Webseiten einen Shop realisieren, Online-Banking anbieten, Gesundheits- oder Versicherungsdaten verwalten etc., sind aber weiterhin auf höherwertige und damit leider auch recht teure Zertifikate von anderen Anbietern angewiesen.

Software In den ersten Monaten hat Let's Encrypt selbst Software für die Zertifikatsverwaltung angeboten, unter anderem das inzwischen veraltete Kommando `letsencrypt`. Mittlerweile kümmert sich die Electronic Frontier Foundation (EFF) um die Entwicklung und Wartung der Software. Der offizielle Let's-Encrypt-Client hat seither den Namen `certbot` (<https://certbot.eff.org>). Darüber hinaus gibt es eine Reihe anderer Clients in allen erdenklichen Programmiersprachen. Eine Referenz finden Sie hier:

<https://letsencrypt.org/docs/client-options>

certbot installieren

Einige Distributionen stellen `certbot` bereits in den eigenen Paketquellen zur Verfügung, z. B. Fedora. Bei anderen Distributionen müssen Sie externe Repositories aktivieren, z. B. EPEL für CentOS/RHEL oder `jessie-backports` für Debian 8.

```
root# yum install python-certbot-apache           (CentOS/RHEL)
root# apt install python-certbot-apache -t jessie-backports (Debian 8)
root# apt install python-certbot-apache           (Debian 9)
root# dnf install python-certbot-apache           (Fedora)
```

Ubuntu-Pakete befinden sich im Private Package Archive `ppa:certbot/certbot`:

```
root# add-apt-repository ppa:certbot/certbot      (Ubuntu)
root# apt update
root# apt install python-certbot-apache
```

Für openSUSE gab es im Frühjahr 2017 noch kein passendes Paket. Zur Installation führen Sie die folgenden Kommandos durch:

```
root# wget https://dl.eff.org/certbot-auto      (openSUSE)
root# mv certbot-auto /usr/local/bin
root# chmod a+x /usr/local/bin/certbot-auto
root# certbot-auto -h
```

`certbot-auto` aktualisiert sich bei der ersten Ausführung selbst (und in der Folge immer wieder, wenn es neue Versionen gibt). Beim ersten Mal werden dabei diverse Pakete installiert. Beachten Sie, dass das Kommando bei einer manuellen Installation `certbot-auto` heißt, bei der Installation aus einem Paket dagegen einfach `certbot`!

Weitere Installationsanleitungen für alle erdenklichen Kombinationen aus verschiedenen Webservern und Distributionen finden Sie hier:

<https://certbot.eff.org>

Zertifikate einrichten

Das Kommando `certbot` erfüllt verschiedene Aufgaben, je nachdem, welche Optionen übergeben werden:

- Es kontaktiert den Server des Let's-Encrypt-Projekts, fordert dort ein Zertifikat an, lädt es herunter und installiert es im Verzeichnis `/etc/letsencrypt/domainname`.
- Es passt die Konfigurationsdateien von Apache oder `nginx` so an, dass das neue Zertifikat verwendet wird.
- Es testet, welche der aktuell installierten Zertifikate in den nächsten 30 Tagen ablaufen, und erneuert diese.

Verwenden Sie certbot zuerst im Testmodus!

Um Missbrauch zu vermeiden, gibt es strikte Limits, wie viele Zertifikate für eine Domain in einer bestimmten Zeit erzeugt werden dürfen:

<https://letsencrypt.org/docs/rate-limits>

Die wichtigste Regel lautet, dass Sie pro Domain maximal 20 Zertifikate pro Woche erzeugen können, wobei ein Zertifikat mehrere Sub-Domänen umfassen darf. Um zu vermeiden, dass Sie diese Limits überschreiten, sollten Sie `certbot` für erste Tests immer mit der Option `--staging` bzw. `--test-cert` (veraltet) aufrufen. Damit erhalten Sie Zertifikate von einem Test-System. Erst wenn Sie sicher sind, dass alles funktioniert, erstellen Sie die richtigen Zertifikate ohne diese Option.

Um Let's-Encrypt-Zertifikate anzufordern und für den Webserver Apache zu installieren, führen Sie das folgende Kommando aus. Dabei ersetzen Sie `meine-domain.de` durch Ihren Domainnamen. Die Zertifikate für `smtp.*`, `mail.*` und `imap.*` können später zur Konfiguration des Mail-Servers verwendet werden (siehe Abschnitt 34.3, »Postfix-Verschlüsselung (TLS/STARTTLS)«). Wenn Sie nicht vorhaben, hierfür Let's-Encrypt-Zertifikate zu verwenden, lassen Sie diese Hostnamen weg.

Zertifikate anfordern und installieren

Auf keinen Fall verzichten sollten Sie hingegen auf die `www`-Variante, selbst dann, wenn Sie `http://domainname` gegenüber `http://www.domainname` vorziehen. Das `www`-Subdomain-Zertifikat ist erforderlich, damit Rewrite-Regeln von `www.domainname` auf `domainname` später auch für HTTPS funktionieren.

```
root# certbot --apache --staging -d meine-domain.de \
      -d www.meine-domain.de -d imap.meine-domain.de \
      -d smtp.meine-domain.de -d mail.meine-domain.de
```

certbot fordert Sie bei der ersten Installation eines Zertifikats dazu auf, eine E-Mail-Adresse anzugeben. An diese E-Mail-Adresse wird vor dem Ablauf eines Zertifikats eine Warnung gesendet (z. B. falls die automatische Aktualisierung aus irgendeinem Grund versagt hat). Oft fragt certbot auch, welche Apache-Konfigurationsdatei es verändern soll. Wenn die automatische Konfiguration der Konfigurationsdateien zu Fehlern führt, rufen Sie das Kommando in der Form `certbot certonly` auf, und führen die Konfiguration selbst durch.

Erst wenn Sie sicher sind, dass alles klappt, entfernen Sie die Option `--staging` und wiederholen das Kommando nochmals zur Installation der endgültigen Zertifikate.

Wo sind die Zertifikate? certbot installiert die Zertifikate und Schlüssel in Verzeichnisse der Form `/etc/letsencrypt/archive/domainname`. In `/etc/letsencrypt/live/domainname` befinden sich Links auf die gerade gültigen Zertifikate. `/etc/letsencrypt` enthält darüber hinaus diverse Metadaten, die für den Zertifikatserneuerungsprozess benötigt werden.

Apache-Konfiguration Damit Let's-Encrypt-Zertifikate für einen bestimmten virtuellen Host verwendet werden, müssen nur drei `SSLxxxFile`-Anweisungen eingefügt bzw. geändert werden:

```
<VirtualHost *:443>
  ServerName  meine-domain.de
  ServerAlias www.meine-domain.de
  ServerAlias imap.meine-domain.de
  ServerAlias smtp.meine-domain.de
  DocumentRoot ...

  SSLCertificateFile      /etc/letsencrypt/live/meine-domain.de/cert.pem
  SSLCertificateKeyFile   /etc/letsencrypt/live/meine-domain.de/privkey.pem
  SSLCertificateChainFile /etc/letsencrypt/live/meine-domain.de/chain.pem
</VirtualHost>
```

Für jede Subdomain einen ServerAlias!

Aus nicht ganz einsichtigen Gründen funktioniert die automatische Zertifikats-erneuerung nur dann ohne Fehler, wenn die Apache-Konfigurationsdateien jede Subdomain mit `ServerName` oder `ServerAlias` enthält!

Wenn Sie Let's Encrypt wie beschrieben auch dazu verwenden, um Zertifikate für den Mail-Server zu erzeugen, dann müssen Sie entsprechende `ServerAlias`-Zeilen hinzufügen, obwohl diese für den Webserver-Betrieb vollkommen sinnlos sind.

Anfang Juli 2017 wurde bekannt, dass Let's Encrypt ab Januar 2018 auch Wildcard-Zertifikate anbieten wird, also Zertifikate, die für alle Subdomains gelten (`*.meine-domain.de`). Das wird die Sache voraussichtlich vereinfachen. Allerdings wird dazu eine neue API-Version zum Einsatz kommen (ACME v2, aktuell ist ACME v1). Es ist aktuell nicht abzusehen, ob sich dadurch der Umgang mit dem `certbot`-Kommando ändert.

Die README-Datei in `/etc/letsencrypt/live/meine-domain.de` empfiehlt eigentlich die Verwendung von `fullchain.pem` anstelle von `cert.pem` und `chain.pem`, aber die Webseite <https://www.ssllabs.com/ssltest> kritisiert dann, dass die Zertifikatskette unvollständig sei.

Unter Debian und Ubuntu richtet certbot außerdem eine Datei mit diversen SSL-Optionen ein:

```
# /etc/letsencrypt/options-ssl-apache.conf
SSLEngine on
SSLProtocol          all -SSLv2 -SSLv3
SSLCipherSuite        ECDHE-RSA-AES128-GCM-SHA256:...:!KRB5-DES-CBC3-SHA
SSLHonorCipherOrder  on
SSLCompression        off
SSLOptions            +StrictRequire
...
```

Auf CentOS/RHEL-Rechnern verzichtet certbot auf eine vergleichbare Konfiguration. Sie sollten sich selbst darum kümmern, weil die Defaulteinstellungen in `ssl.conf` unter CentOS/RHEL mangelhaft sind.

Als weitere Optimierung können Sie noch *HTTP Strict Transport Security* (HTST) aktivieren. Damit teilen Sie dem Browser mit, dass er für eine bestimmte Zeit ausschließlich verschlüsselte Verbindungen zu Ihrer Seite herstellen darf:

https://de.wikipedia.org/wiki/HTTP_Strict_Transport_Security

```
Header always set Strict-Transport-Security \
    "max-age=63072000; includeSubdomains; preload"
```

Häufig ist erwünscht, dass alle HTTP-Seitenzugriffe automatisch auf HTTPS umgeleitet werden. certbot fragt, ob es entsprechende Anweisungen in die Apache-Konfigurationsdateien einbauen soll – aber das können Sie natürlich später auch selbst erledigen. Die folgenden Zeilen geben dafür ein Muster:

Umleitung von
HTTP auf HTTPS

```
<VirtualHost _default_:80>
  ServerName meine-domain.de
  ServerAlias www. meine-domain.de
  ServerSignature Off

  RewriteEngine On
  RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,QA,R=permanent]

  ErrorLog /var/log/httpd/redirect.error.log
  LogLevel warn
</VirtualHost>
```

Falls Sie außerdem eine Umleitung von *www.meine-domain.de* auf *meine-domain.de* wünschen, fügen Sie die folgenden Zeilen hinzu:

```
# Quelle: http://stackoverflow.com/questions/21467329
RewriteCond %{HTTP_HOST} ^(www\.)(.*) [NC]
RewriteRule (.*?) https://%2%{REQUEST_URI} [L,R=301]
```

Zertifikate erneuern

Das Kommando `certbot renew` kontrolliert die Let’s-Encrypt-Zertifikate Ihres Rechners und erneuert alle, die in den nächsten 30 Tagen auslaufen. Es geht also nur noch darum, den Aufruf von `certbot renew` automatisch einmal pro Woche durchzuführen. Dazu erstellen Sie mit einem Editor die Datei `/etc/cron.weekly/letsencrypt` mit dem folgenden Inhalt:

```
#!/bin/bash
# Datei /etc/cron.weekly/letsencrypt
certbot renew
```

Diese Datei machen Sie mit `chmod a+x` ausführbar – fertig! Zu Testzwecken können Sie eine Zertifikatserneuerung mit `certbot --force-renewal renew` ausnahmsweise erzwingen. (Diese Option darf auf keinen Fall in einem Script stehen, das regelmäßig durch Cron ausgeführt wird.)

Um den nach einem Zertifikatswechsel erforderlichen Neustart von Apache kümmert sich `certbot` selbst. Dabei laufen vorhandene Apache-Prozesse mit der alten Konfiguration weiter, neue Prozesse verwenden die neue Konfiguration. Es kommt also zu keinem Verbindungsabbruch oder Session-Verlust. Informationen dazu finden Sie unter:

<https://community.letsencrypt.org/t/is-server-restart-needed-when-obtaining-certs-using-certbot-and-apache-module/17267>

Falls Sie Let’s-Encrypt-Zertifikate auch für Postfix und Dovecot verwenden, müssen Sie diese Programme explizit neu starten. Die folgende Variante des obigen `letsencrypt`-Scripts testet, ob es in `/etc/letsencrypt/live` Zertifikate gibt, die sich innerhalb der letzten 24 Stunden geändert haben:

```
#!/bin/bash
# Datei /etc/cron.weekly/letsencrypt
certbot renew
result=$(find /etc/letsencrypt/live/ -type l -mtime -1 )
if [ -n "$result" ]; then
  systemctl restart postfix
  systemctl restart dovecot
fi
```

Einschränkungen und Sicherheitsvorkehrungen

Bei der Nutzung von Let’s Encrypt sollten Ihnen die folgenden Einschränkungen bewusst sein:

- ▶ Sie können ein Zertifikat später nicht um eine Sub-Domäne erweitern. Wenn Sie das möchten, müssen Sie vielmehr das gesamte Zertifikat neu erzeugen. Das ist an sich kein Problem, es gelten dafür aber die Let’s-Encrypt-Rate-Limits (aktuell maximal 20 Zertifikate je Domäne pro Woche).
- ▶ Sie können mit `certbot` nur Zertifikate für Domänen erstellen, deren DNS-Einträge auf Ihren Server zeigen. Diese naheliegende Einschränkung verhindert die missbräuchliche Erstellung von Zertifikaten für fremde Domänen.
- ▶ Der Zertifikats-Update-Prozess setzt voraus, dass die bisherigen Zertifikate und Schlüssel in `/etc/letsencrypt` zur Verfügung stehen. Das klingt auf den ersten Blick wie eine Selbstverständlichkeit. Pech haben Sie freilich, wenn diese Dateien bei einem Server-Crash oder -Umzug verloren gehen. Eine Aktualisierung der Zertifikate ist unmöglich, und eine Neueinstellung ist erst erlaubt, nachdem die bisherigen Zertifikate ausgelaufen sind – in der Regel also erst nach drei Monaten! Kümmern Sie sich rechtzeitig um Backups des gesamten Verzeichnisses `/etc/letsencrypt`!

Ich habe im vergangenen Jahr mehrere Websites auf Zertifikate von Let’s Encrypt umgestellt. Dabei hatte ich mehrfach Probleme mit der durch das `certbot`-Kommando durchgeführten Veränderung der Apache-Konfiguration. Dieser heikle Prozess funktioniert offensichtlich nur einfachen Fällen fehlerfrei. Rufen Sie `certbot` gegebenenfalls in der Form `certbot certonly ...` auf und führen Sie die Apache-Konfiguration anschließend selbst durch. Davon abgesehen funktionierten die Zertifikatsausstellung, deren Installation in das Verzeichnis `/etc/letsencrypt` sowie deren Updates absolut problemlos.

Postfix und Dovecot neu starten

32.6 Webzugriffsstatistiken

Wer einen eigenen Webserver betreibt oder für jemand anderen administriert, will in der Regel auch wissen, wie viele Personen die Website pro Tag besuchen, welche Webbrowser dabei zum Einsatz kommen etc. Ich stelle Ihnen hier vorweg einige Tools kurz vor und gehe dann etwas ausführlicher auf den Einsatz von GoAccess ein.

Webalizer und AWStats In der Vergangenheit wurden zur Erstellung derartiger Statistiken häufig die Programme Webalizer oder AWStats verwendet. Diese Programme aus den Anfangszeiten des Internets werden zwar noch gewartet, sind aber nicht mehr zeitgemäß. Die Konfiguration ist relativ aufwendig, weil ihr Aufruf mit der Rotation der Logging-Dateien synchronisiert werden muss. Genaueres finden Sie unter:

<http://awstats.org>
<http://webalizer.org>

Google Analytics Gewissermaßen der Star unter den modernen Web-Analyzer-Tools ist *Google Analytics*. Dazu müssen Sie auf den Seiten Ihrer Website JavaScript-Code einbauen, der jeden Seitenzugriff an einen zentralen Server weiterleitet. Diese Vorgehensweise erleichtert die Unterscheidung zwischen »echten« Besuchern und Suchrobotern und führt zu genaueren Ergebnissen, die in Echtzeit beobachtet werden können. Da Google Analytics nicht die Logging-Dateien auswertet, sondern auf Code basiert, der direkt in der Website integriert ist, kommt Google Analytics wesentlich besser mit dynamischen Websites zurecht, bei denen die Startseite nie verlassen wird (Single-Page-Webanwendungen).

Beachten Sie aber, dass der Einsatz von Google Analytics unter Einhaltung der deutschen Datenschutzgesetze problematisch ist! Auf jeden Fall sollten Sie in Ihrer Website einen Bestätigungsdialog einbauen, der die Besucher auf die Verwendung von Google Analytics hinweist. Dieser Dialog wird oft mit der in vielen Ländern ebenfalls vorgeschriebenen (wenngleich vollkommen sinnlosen) Cookie-Einverständniserklärung kombiniert.

<https://www.google.com/analytics>

Piwik Piwik ist eine Open-Source-Alternative zu Google Analytics. Sein Einsatz vermeidet, dass Google mit noch mehr Daten gefüttert wird – die erfassten Daten bleiben unter Ihrer Kontrolle. Aus der Perspektive des Datenschutzes agiert Piwik aber ähnlich wie Google Analytics; daher sollten Sie auch bei der Verwendung von Piwik einen unübersehbaren Hinweis in Ihre Website einbauen.

<https://piwik.org>
<https://www.datenschutzzentrum.de/uploads/projekte/verbraucherdatenschutz/20110315-webanalyse-piwik.pdf>

GoAccess

GoAccess ist aus meiner Sicht ein guter Kompromiss zwischen den veralteten Programmen Webalizer und AWStats auf der einen Seite und datenschutztechnisch problematischen Tools wie Google Analytics oder Piwik auf der anderen Seite. GoAccess wertet die Logging-Dateien des Webserver aus und agiert insofern ähnlich wie Webalizer oder AWStats. Im Gegensatz zu diesen Programmen eignet sich das Programm aber auch zur Echtzeit-Analyse des Web-Traffics. Außerdem kann es bei Bedarf in einem Terminalfenster via SSH verwendet werden, also ohne den sonst üblichen Zwischenschritt der Generierung von Reports in Form von Webseiten.

<https://goaccess.io>

GoAccess steht bei vielen Distributionen als Paket zur Verfügung und kann mit `apt/dnf/yum/zypper` installiert werden. Bei meinen Tests waren die so installierten Versionen aber durchwegs veraltet. Besser ist es daher, wenn Sie eine manuelle Installation gemäß den Anweisungen auf der Download-Seite des Projekts durchführen. Im Frühjahr 2017 waren dazu die folgenden Anweisungen erforderlich:

Installation

```
root# apt install libncursesw5-dev libgeoip-dev libssl-dev      (Ubuntu)
root# yum groupinstall development                             (CentOS)
root# yum install ncurses-devel geoip-devel openssl-devel wget (CentOS Forts.)

user$ wget http://tar.goaccess.io/goaccess-1.2.tar.gz          (alle Distrib.)
user$ tar -xzf goaccess-1.2.tar.gz
user$ cd goaccess-1.2/
user$ ./configure --enable-utf8 --enable-geoip=legacy --with-openssl
user$ make
root# make install

root# ln -s /usr/local/bin/goaccess /usr/bin                    (CentOS)
```

In Zukunft wird sich die Versionsnummer sicher ändern.

Am einfachsten rufen Sie `goaccess` in einer SSH-Session auf und übergeben als Parameter den Dateinamen der Apache-Logging-Datei. Unter Ubuntu gelingt das einem nicht privilegierter Benutzer:

Anwendung im Terminal

```
user$ goaccess /var/log/apache2/access.log
```

Unter CentOS/Fedora/RHEL befinden sich die Logging-Dateien in `/var/log/httpd` und sind nur für `root` zugänglich. Der Aufruf von `goaccess` muss dann mit `root`-Rechten erfolgen.

```
root# goaccess /var/log/httpd/access_.log
```

Beim Start müssen Sie angeben, in welchem Format die Logging-Datei vorliegt. Normalerweise reicht es aus, den vorgegebenen Eintrag `NCSA COMBINED LOG FORMAT`

zu bestätigen. goaccess zeigt dann eine Auswertung der Logging-Datei an und aktualisiert diese regelmäßig, bis Sie das Programm mit **Q** beenden (siehe Abbildung 32.4).

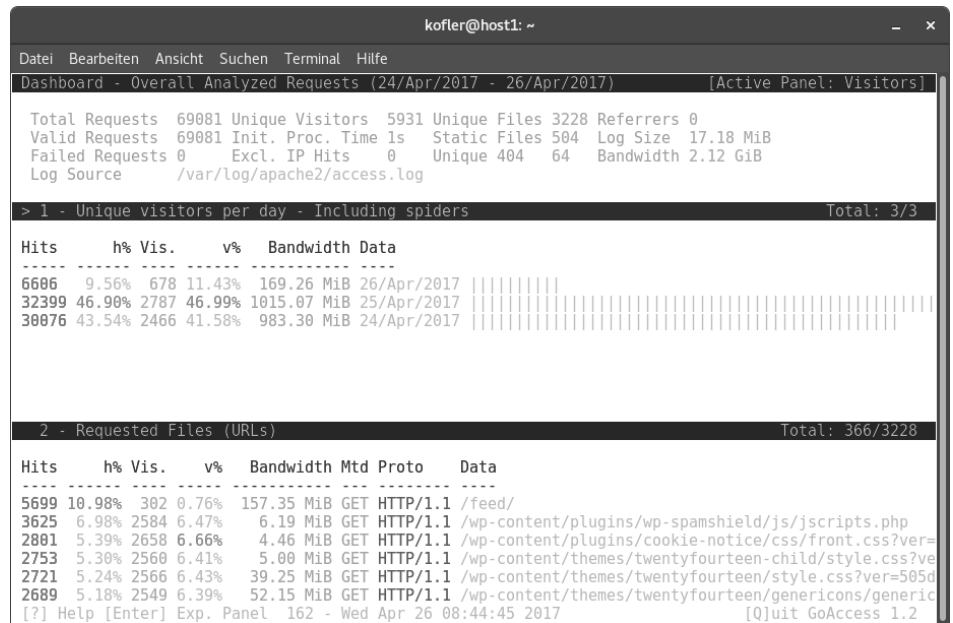


Abbildung 32.4 Zugriffsstatistiken im Terminalfenster

Mit den Cursortasten oder mit **↩** können Sie durch weitere Rubriken scrollen (statische Zugriffe, 404-Fehler, Hostnamen und IP-Adressen etc.). **←** erweitert die gerade aktive Rubrik um weitere Details. **S** sortiert die Zeilen nach einem anderen Kriterium. Eine Zusammenfassung weiterer Tastenkürzel liefert **H**.

Die meisten Distributionen richten täglich oder wöchentlich neue Access-Log-Dateien ein und komprimieren die älteren Dateien (siehe auch die Beschreibung von logrotate in Abschnitt 18.9, »Logging (Syslog)«). Das folgende Kommando verarbeitet alle komprimierten sowie die beiden nicht komprimierten Dateien access.log.1 und access.log.2. Beachten Sie, dass ein derartiger Aufruf von goaccess anfänglich ziemlich lange dauert (alle Logging-Dateien müssen dekomprimiert und eingelesen werden) und dass der Speicherbedarf von goaccess in diesem Fall erheblich ist – je nach Größe der Log-Dateien im GiB-Bereich! Wenn Sie das Zeichen * durch ? ersetzen, werden nur die letzten zehn Logging-Dateien berücksichtigt, was oft auch ausreicht.

```
user$ cd /var/log/apache2
user$ zcat access.log.*.gz | \
      goaccess --log-format=COMBINED access.log access.log.1 -
```

Format-Fehlermeldung

Wenn goaccess eine Fehlermeldung der Art *No time format was found on your conf file* liefert, haben Sie zwei Möglichkeiten: Entweder geben Sie das Logformat Ihres Webserver mit einer Option an (wie im obigen Beispiel mit --log-format) oder Sie schreiben das Format in der Konfigurationsdatei fest, deren Ort im Rahmen der Fehlermeldung angezeigt wird (bei einer manuellen Konfiguration /usr/local/etc/goaccess.conf).

Indem Sie goaccess zusätzlich die Option -o out.html übergeben, erzeugen Sie eine HTML-Datei mit der Zugriffsstatistik. Diese Seite können Sie dann mit einem Webbrowser ansehen (siehe Abbildung 32.5). Die Erzeugung derartiger Zugriffsstatistiken können Sie natürlich in einem Cron-Job automatisieren und einmal täglich oder wöchentlich durchführen.

HTML-Reports erzeugen

```
user$ cd /var/log/apache2
user$ zcat access.log.*.gz | goaccess -o /var/www/html/myreports/out.html \
      --log-format=COMBINED access.log access.log.1
```

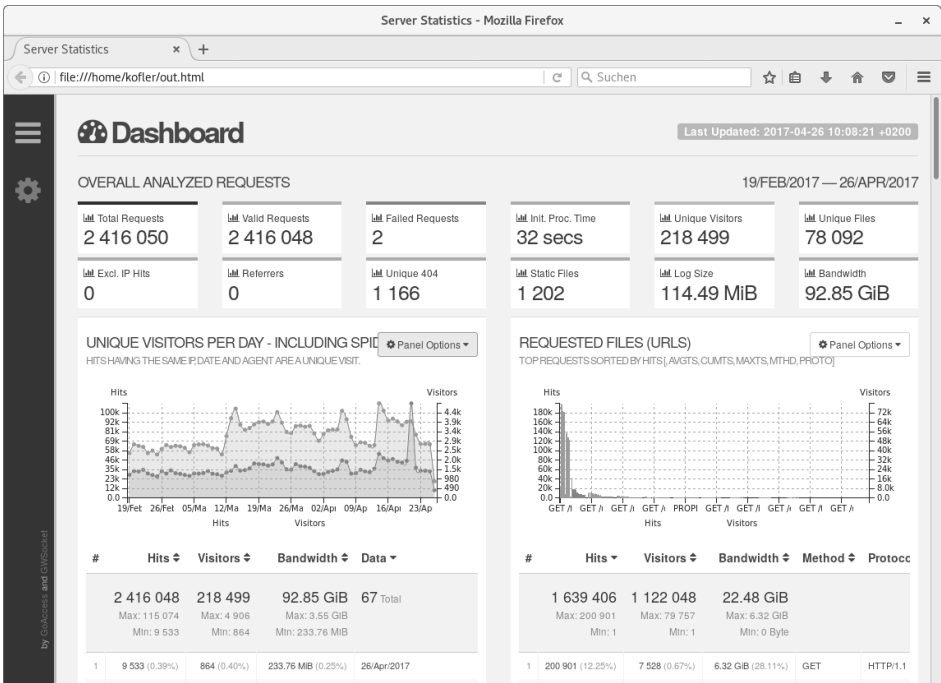


Abbildung 32.5 Zugriffsstatistiken im Webbrowser auswerten

goaccess kann sogar Echtzeit-Updates über die HTML-Seite liefern. Dazu übergeben Sie an goaccess zusätzlich die Option --real-time-html. Wenn die resultierende Seite

über eine HTTPS-Verbindung übertragen werden soll, müssen Sie außerdem das Zertifikat und dessen Schlüssel in Parametern übergeben. Der Zugriff auf diese Dateien erfordert die Ausführung des Kommandos mit root-Rechten.

```
root# cd /var/log/apache2
root# goaccess --log-format=COMBINED access.log access.log.1 \
-o /var/www/html/myreports/uploads/report.html --real-time-html \
--ssl-cert=/etc/letsencrypt/live/mysite/fullchain.pem \
--ssl-key=/etc/letsencrypt/live/mysite/privkey.pem
```

32.7 PHP

Dynamische Webseiten Apache an sich kann nur statische Webseiten übertragen. Alle modernen Websites nutzen aber dynamische Seiten. Jedes Mal, wenn eine derartige Seite angefordert wird, startet Apache ein externes Programm, verarbeitet den Code der Seite und liefert als Ergebnis eine Seite, die individuell angepasst ist. Damit kann die Seite beispielsweise die aktuelle Uhrzeit enthalten oder das Ergebnis einer Datenbankabfrage oder eine ständig wechselnde Werbeeinblendung etc.

PHP Zur Programmierung dynamischer Webseiten eignen sich zahllose Programmiersprachen – z. B. Perl, PHP oder Java. Die Grundidee einer PHP-Webseite besteht darin, dass die Datei mit der Kennung *.php sowohl HTML- als auch PHP-Code enthält. PHP-Code wird mit dem Tag <?php eingeleitet und endet mit ?>.

Wenn ein Webnutzer eine PHP-Seite anfordert, übergibt Apache die Seite an den PHP-Interpreter. Dort wird der PHP-Code ausgeführt. Das Ergebnis des Codes wird direkt in die HTML-Datei eingebettet. Der PHP-Interpreter übergibt die resultierende Seite zurück an Apache, und dieser sendet sie dem Webnutzer. Der Webbrowser des Nutzers sieht also nie den PHP-Code, sondern immer nur die resultierende HTML-Seite.

Hello World! Der Platz reicht hier nicht für eine Einführung in die Programmiersprache PHP. Stattdessen soll das folgende Minibeispiel das Konzept von PHP veranschaulichen. Die folgende Datei liefert nach der Verarbeitung durch den PHP-Interpreter eine HTML-Seite mit der aktuellen Uhrzeit:

```
<!DOCTYPE html>
<html><head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=utf-8" />
  <title>PHP-Beispiel</title>
</head><body>

<p>Die aktuelle Uhrzeit auf diesem Server:
```

```
<?php
  date_default_timezone_set("Europe/Berlin");
  echo strftime("%k:%M:%S") . "</p>";
  echo "<p>Sonderzeichentest: äöü</p>";
?>
</body></html>
```

Sofern PHP nicht bereits mit Apache mitinstalliert wurde, installieren Sie mit Ihrem Paketverwaltungsprogramm die erforderlichen php-Pakete. Was »erforderlich« ist, ist allerdings gar nicht so einfach festzustellen: Ähnlich wie bei Apache ist auch PHP über zahlreiche Pakete verteilt, die die Sprache an sich sowie diverse Erweiterungen enthalten. Für erste Experimente reichen üblicherweise php<n>, php<n>-common sowie libapache2-mod-php<n>. Soweit sich nicht die Paketverwaltung darum kümmert, müssen Sie Apache nach der Installation neu starten, damit der Webserver neu hinzugekommene PHP-Module berücksichtigt.

Zahllose Optionen des PHP-Interpreters werden durch die Datei php.ini gesteuert. Im Regelfall können Sie die Grundeinstellungen einfach beibehalten. Der Ort dieser Datei sowie weiterer PHP-Konfigurationsdateien ist wieder einmal distributionsabhängig:

Debian und Ubuntu mit PHP 5:	/etc/php5/apache2/php.ini, /etc/php5/apache2/conf.d/*.ini
Debian und Ubuntu mit PHP 7:	/etc/php/7.<n>/apache2/php.ini, /etc/php/7.<n>/apache2/conf.d/*.ini
CentOS, Fedora, RHEL:	/etc/php.ini, /etc/php.d/*.ini
SUSE:	/etc/php5/apache2/php.ini

Um zu testen, ob die PHP-Installation funktioniert, erstellen Sie die Datei phptest.php, die aus nur einer einzigen Zeile Code besteht:

```
<?php phpinfo(); ?>
```

Kopieren Sie diese Datei in das DocumentRoot-Verzeichnis (siehe Tabelle 32.1), und stellen Sie sicher, dass Apache die Datei lesen darf. Obwohl es sich bei PHP-Dateien eigentlich um Script-Dateien handelt, reichen Leserechte. Zugriffsrechte zum Ausführen (x-Zugriffsbits) sind nicht erforderlich.

Mit einem Webbrowser sehen Sie sich nun die Seite http://localhost/php-test.php an. Das Ergebnis ist eine sehr umfangreiche Seite, die alle möglichen Optionen und Einstellungen von Apache und PHP enthält (siehe Abbildung 32.6). Aus Sicherheitsgründen ist es nicht empfehlenswert, eine derartige Seite frei zugänglich ins Internet zu stellen. Sie enthält eine Menge Informationen über die Konfiguration Ihres Webservers.

Installation

Konfiguration

Test

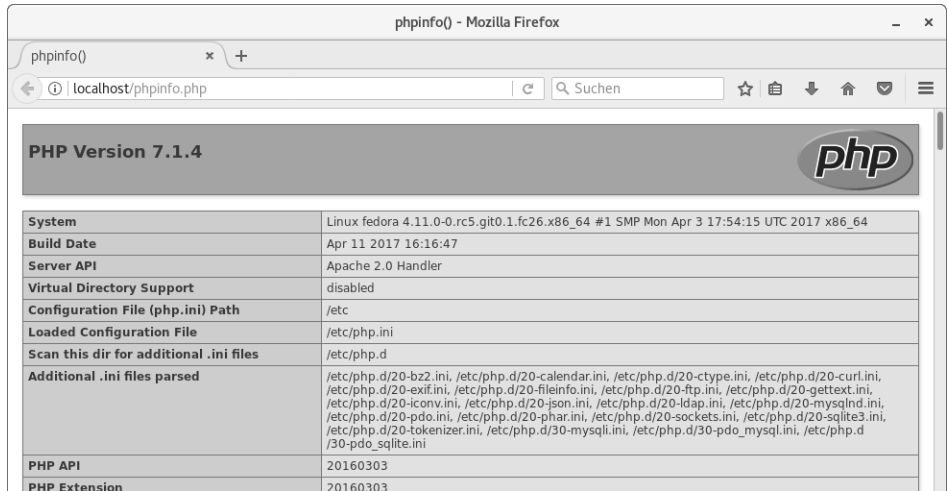


Abbildung 32.6 PHP-Testseite

Wenn es nicht funktioniert

Wenn Sie statt der Testseite den PHP-Code sehen oder die PHP-Datei zum Download angeboten bekommen, ist die wahrscheinlichste Fehlerursache die, dass Sie als Webadresse den Dateinamen (z. B. `/srv/www/htdocs/phpinfo.php`) angegeben haben. In diesem Fall wird die Datei direkt aus dem lokalen Dateisystem gelesen, anstatt von Apache und PHP verarbeitet zu werden. Die Webadresse muss mit `http://` beginnen!

Eine weitere Fehlerursache ist die Apache-Konfiguration: Haben Sie Apache nach der Installation von PHP bzw. nach der Veränderung von Konfigurationsdateien neu gestartet?

Wenn es einmal nicht geklappt hat, kann Ihnen in der Folge der Cache Ihres Webbrowsers einen Strich durch die Rechnung machen. Anstatt die Seite neu von Apache anzufordern, was nun vielleicht funktionieren würde, liest der Browser die Seite aus dem internen Cache. Starten Sie den Browser sicherheitshalber neu bzw. löschen Sie den Cache!

32.8 FTP-Server (vsftpd)

Vielen Webservern gesellt sich ein FTP-Server hinzu, der je nach Website zwei Aufgaben erfüllt: Einerseits ermöglicht er den Download großer Dateien, die auf der Website zur Verfügung gestellt werden, andererseits hilft er bei der Wartung bzw. Aktualisierung der Website, indem er eine einfache Möglichkeit zum Upload von Dateien zulässt.

Sicherheit

FTP ist ein sehr altes Programm. Sein Protokoll führt in Kombination mit Firewalls bzw. mit Masquerading oft zu Problemen. Noch problematischer ist der Umstand, dass beim Verbindungsaufbau zwischen einem FTP-Client und dem -Server der Benutzername und das Passwort unverschlüsselt übertragen werden. Da stehen jedem sicherheitsbewussten Anwender die Haare zu Berge!

Natürlich gibt es schon längst sichere Alternativen zu FTP. Unter anderem stellt der in Kapitel 31 beschriebene SSH-Server mit SFTP (*Secure FTP*) auch Dienste zur Dateiübertragung zur Verfügung. Das Problem liegt hier mehr auf der Client-Seite: Es gibt nur relativ wenige benutzerfreundliche Programme, die SFTP beherrschen. Aus diesem Grund wird FTP trotz aller Sicherheitsmängel noch immer recht häufig eingesetzt.

Eine andere Alternative ist der WebDAV-Standard, der das HTTP-Protokoll erweitert und die Datenübertragung in beide Richtungen erleichtert. Beispielsweise unterstützt Apache in Kombination mit dem Modul `mod_dav` WebDAV:

http://httpd.apache.org/docs/2.4/mod/mod_dav.html
<https://wiki.ubuntuusers.de/Webdav>

Wenn Sie auf einen traditionellen FTP-Server nicht verzichten möchten, können Sie diesen auch als reinen Anonymous-FTP-Server konfigurieren. Dabei werden beim Login keine kritischen Daten übertragen. Allerdings schränkt das auch die Anwendung von FTP stark ein. Zur einfachen Wartung einer Website lässt sich FTP dann nicht mehr verwenden.

Es gibt unzählige verschiedene FTP-Server. Das populärste Programm ist momentan `vsftpd`. Alle gängigen Distributionen stellen hierfür ein Paket zur Verfügung. `vsftpd` steht für *Very Secure FTP Daemon*. Das Attribut *Very Secure* ist aber unter dem Vorbehalt zu sehen, dass auch der beste FTP-Server die Sicherheitsmängel des FTP-Protokolls aufweist.

vsftpd

`vsftpd` kann auf zwei Arten gestartet werden: entweder als eigenständiger Dämon durch das Init-System oder über `xinetd`. Bei den meisten Distributionen ist die Dämon-Variante vorkonfiguriert. Die Konfigurationsdatei `vsftpd.conf` muss dazu die Anweisung `listen=YES` enthalten.

Start als Dämon

Um den FTP-Server zu starten bzw. zu stoppen, verwenden Sie je nach Distribution die üblichen Kommandos (siehe Abschnitt 12.5, »Systemprozesse (Dämonen)«). Unter CentOS, Fedora und RHEL gehen Sie z. B. so vor:

```
root# systemctl start vsftpd
root# systemctl enable vsftpd
```

Wie üblich müssen Sie auch sicherstellen, dass die Firewall die FTP-Ports 20 und 21 nicht blockiert. Unter SUSE verwenden Sie zur Firewall-Konfiguration am besten YaST.

Unter CentOS/Fedora/RHEL stellen Sie zuerst fest, welche Firewall-Zone für die Netzwerkschnittstelle zum Internet gilt (hier FedoraWorkstation), und aktivieren dann für diese Zone Ausnahmeregeln:

```
root# firewall-cmd --get-zone-of-interface=enp0s3 (aktive Zone herausfinden)
FedoraWorkstation
root# firewall-cmd --permanent --zone=FedoraWorkstation --add-service=ftp
root# firewall-cmd --reload
```

Konfiguration Die Konfiguration von vsftpd erfolgt durch die Datei /etc/vsftpd.conf bzw. /etc/vsftpd/vsftpd.conf. Standardmäßig ist oft nur ein Read-Only-Zugang per Anonymous FTP zugelassen. FTP-Clients können also nur einen Download, aber keinen Upload durchführen. Wenn Sie neben Anonymous FTP auch Benutzer-Logins benötigen, müssen Sie locale_enable auf YES stellen. Wenn Sie bei dieser FTP-Form auch einen Daten-Upload zulassen möchten, müssen Sie zusätzlich write_enable auf YES stellen. Wenn vsftpd.conf die Zeile tcp_wrappers=Yes enthält, wertet vsftpd wie xinetd die Dateien /etc/hosts.allow und /etc/hosts.deny aus (siehe Abschnitt 37.2, »Basisabsicherung von Netzwerkdiensten«). Die folgenden Zeilen fassen die wichtigsten Einstellungen in vsftpd.conf zusammen:

```
# /etc/vsftpd.conf bzw. /etc/vsftpd/vsftpd.conf
...
local_enable=YES / NO      # FTP-Login zulassen
write_enable=YES / NO      # Daten-Upload grundsätzlich zulassen
...
anonymous_enable=YES / NO  # Anonymous FTP zulassen
anon_upload_enable=YES / NO # Daten-Upload auch bei Anonymous FTP
...
listen=YES / NO            # Start als Init-Dämon (YES) oder durch xinetd (NO)
tcp_wrapper=YES / NO       # hosts.allow und hosts.deny auswerten
```

FTP ausprobieren FTP müsste jetzt eigentlich auf Anhieb funktionieren. Führen Sie auf dem Server-Rechner ftp localhost aus, um zu testen, ob der FTP-Server ordnungsgemäß gestartet wird. Beachten Sie dabei, dass root grundsätzlich keinen FTP-Login durchführen darf.

Anonymous FTP Wenn Anonymous FTP in vsftpd.conf zugelassen ist, akzeptiert vsftpd als Login die Namen anonymous und ftp in Kombination mit einem beliebigen Passwort. Es ist üblich, als Passwort die E-Mail-Adresse anzugeben. vsftpd kontrolliert das aber nicht.

Nach dem Login kann der FTP-Client auf die Dateien des Home-Verzeichnisses des Linux-Benutzers ftp zugreifen. Der Ort dieses Verzeichnisses wird in /etc/passwd angegeben:

```
Debian, Ubuntu:  /srv/ftp
Fedora, Red Hat:  /var/ftp/
SUSE:             /srv/ftp/
```

Upload per Anonymous FTP

Wenn Sie den Upload von Dateien per Anonymous FTP zulassen, sollten Sie darauf achten, dass es nur ein einziges Verzeichnis innerhalb des FTP-Datenverzeichnisses gibt, das Schreibrechte hat – z.B. /var/ftp/upload bei Fedora oder Red Hat. Dieses Verzeichnis sollte dem Benutzer ftp gehören und aus Sicherheitsgründen keine Lese-rechte haben:

```
root# mkdir /var/ftp/upload
root# chown ftp upload
root# chmod 730 upload
```

Somit kann jeder einen Upload durchführen und dem FTP-Administrator anschließend eine E-Mail mit Instruktionen senden, wofür die Datei dient. Andere FTP-Nutzer können die Datei aber im upload-Verzeichnis weder sehen noch herunterladen. Wenn Sie auf derartige Sicherheitsmaßnahmen verzichten, kann es passieren, dass das FTP-Upload-Verzeichnis zum Austausch illegaler Dateien missbraucht wird.

Aus Sicherheitsgründen sind root und einige andere Spezialbenutzer (wie daemon, lp oder nobody) von der FTP-Benutzung ausgeschlossen. Die dazu erforderliche Konfiguration variiert von Distribution zu Distribution.

FTP für root und andere Spezialbenutzer

Bei Fedora und Red Hat erfolgt der Login-Schutz doppelgleisig. Einerseits greift vsftpd für die Login-Kontrolle auf PAM zurück (*Pluggable Authentication Modules*). PAM wertet die Datei /etc/pam.d/vsftpd aus, die auf die Datei etc/vsftpd/ftpusers verweist. Diese Datei enthält eine Liste aller Login-Namen, die FTP nicht benutzen dürfen.

Andererseits wendet vsftpd auch eine interne Login-Kontrolle an und sperrt alle Benutzer, die in /etc/vsftpd.user_list genannt sind. Diese Login-Kontrolle wird in vsftpd.conf durch userlist_enable=YES und userlist_deny=YES (gilt standardmäßig) aktiviert.

Bei Debian, SUSE und Ubuntu greift vsftpd für den Login ebenfalls auf PAM zurück. /etc/pam.d/vsftpd verweist hier allerdings auf /etc/ftpusers. Diese Datei enthält eine Liste aller Login-Namen, die FTP nicht benutzen dürfen.

Kapitel 39

VirtualBox und Vagrant

Virtualisierung macht es möglich, auf einem Rechner mehrere Betriebssysteme parallel auszuführen. Daraus ergeben sich unzählige Anwendungen: Sie können Linux unter Windows ausprobieren oder Windows unter Linux ausführen, eine neue Alpha-Version der Distribution xyz gefahrlos testen, ohne die vorhandene Linux-Installation zu gefährden, Server-Funktionen sicher voneinander trennen etc.

Das für die Plattformen Windows, Linux und macOS verfügbare Programm VirtualBox eignet sich am besten zur Desktop-Virtualisierung, also zur Ausführung von virtuellen Maschinen, die im Grafiksystem bedient werden sollen. Hinter VirtualBox stand ursprünglich die deutsche Firma InnoTek. 2008 übernahm Sun InnoTek, und 2010 kaufte Oracle Sun. Damit ist nun Oracle der Eigentümer von VirtualBox. Umfassende Dokumentation zu VirtualBox finden Sie unter:

<https://www.virtualbox.org>

Große Teile von VirtualBox bestehen aus Open-Source-Code. Die einzige Ausnahme sind einige Zusatzfunktionen, die extra installiert werden müssen. Ihre Nutzung ist für Privatanwender ebenfalls kostenlos, für kommerzielle Anwender hingegen kostenpflichtig.

Dieses Kapitel beschreibt, wie Sie VirtualBox unter Linux installieren und darin virtuelle Maschinen ausführen. Mit Einschränkungen ist VirtualBox auch zur Server-Virtualisierung geeignet. Für diesen Zweck ist das Programm KVM, das ich Ihnen in Kapitel 40 näher vorstelle, wesentlich besser geeignet.

Das Einrichten neuer virtueller Maschinen ist mit Arbeit verbunden. Mit Vagrant können Sie diesen Prozess automatisieren. Das ist vor allem dann praktisch, wenn Sie reproduzierbar Testumgebungen aufsetzen möchten, z. B. für eine bestimmte Server-Konfiguration. Vagrant kann derartige Aufgaben auch für andere Virtualisierungssysteme erledigen. VirtualBox eignet sich aber besonders gut, um Vagrant kennenzulernen.

Vagrant

39.1 VirtualBox installieren

Zur Installation von VirtualBox gibt es grundsätzlich zwei Möglichkeiten. Die bequeme Variante besteht darin, einfach die VirtualBox-Pakete zu verwenden, die sich in den Paketquellen Ihrer Distribution befinden. Sollten diese Pakete fehlen oder nicht ausreichend aktuell sein, können Sie VirtualBox selbst von der Webseite <https://www.virtualbox.org> herunterladen und manuell installieren. Das ist ein wenig umständlicher.

In diesem Abschnitt gehe ich auf beide Varianten ein. Losgelöst davon sind nach Abschluss der Installation noch einige Vorbereitungsarbeiten zu erledigen, die ich am Ende dieses Abschnitts erläutere.

Host und Gast

Bei der Beschreibung von Virtualisierungssystemen hat es sich eingebürgert, das Grundsystem als Wirt (*Host*) und die darauf laufenden virtuellen Maschinen als Gäste (*Guests*) zu bezeichnen. In diesem Kapitel gehe ich davon aus, dass der Host ein bereits funktionierendes Linux-System ist.

VirtualBox-Pakete Ihrer Distribution

VirtualBox unter Linux installieren

Die meisten Distributionen bieten fertige VirtualBox-Pakete an. Bei Fedora müssen Sie vorher die `rpmfusion`-Paketquelle aktivieren. Bei openSUSE befinden sich die Kernfunktionen und die Benutzeroberfläche in getrennten Paketen; dort müssen Sie auch das Paket `virtualbox-qt` installieren.

Nicht erforderlich sind hingegen die diversen `virtualbox-guest`-Pakete! Diese Pakete enthalten Treiber, die in virtuellen Maschinen auszuführen sind, also wenn eine Linux-Distribution selbst innerhalb von VirtualBox ausgeführt werden soll.

VirtualBox-Kernelmodule

VirtualBox greift auf dem Wirtssystem auf die vier Kernelmodule `vboxdrv`, `vboxpci`, `vboxnetadp` und `vboxnetflt` zurück. Manche Distributionen stellen diese Module in binärer Form durch ein eigenes Paket zur Verfügung, das bei jedem Kernel-Update aktualisiert wird. Bei openSUSE lautet der Paketname `virtualbox-host-kmp-default`.

DKMS

Bei anderen Distributionen wird der Quellcode der VirtualBox-Pakete installiert. Bei jedem Kernel-Update müssen die entsprechenden VirtualBox-Module neu kompiliert werden. Darum kümmert sich bei einigen Distributionen DKMS (*Dynamic Kernel Module Support*). Dies ist z. B. bei Ubuntu der Fall, wo Sie das Paket `virtualbox-dkms` installieren müssen.

Die RPMFusion-Paketquelle für Fedora sieht anstelle von DKMS das Kommando `akmods` vor. Zur erstmaligen Installation der VirtualBox-Kernelmodule aktivieren Sie zuerst die RPMFusion-Paketquellen und führen dann die folgenden Kommandos aus:

akmods (Fedora)

```
root# dnf install akmod-VirtualBox kernel-devel-$(uname -r)
root# akmods
root# systemctl restart systemd-modules-load
```

Das erste Kommando installiert die erforderlichen Pakete, das zweite kompiliert die Kernelmodule, das dritte lädt sie. In der Zukunft sollte sich `akmods` nach jedem Kernel-Update selbstständig um die Aktualisierung der VirtualBox-Treiber kümmern. Bei meinen Tests hat das häufig nicht funktioniert. Sie müssen dann die obigen drei Kommandos neuerlich ausführen.

Fedora bereitet auch sonst als Gast in VirtualBox oft Probleme. Das liegt daran, dass Fedora häufig die allerneuesten xorg-Grafiktreiber verwendet. Diese sind nicht immer mit VirtualBox kompatibel. Das kann dazu führen, dass Sie trotz installierter Gasttreiber nur mit einer Bildschirmauflösung von 1024 × 768 Pixel arbeiten können.

Steht weder DKMS noch `akmods` zur Verfügung, können Sie die Module durch ein Script manuell kompilieren:

Manuell kompilieren

```
root# /usr/lib/virtualbox/vboxdrv.sh setup
```

Zum Kompilieren sind aber auch der C-Compiler `gcc` sowie die Kernel-Header-Dateien erforderlich. Bei vielen Distributionen müssen Sie die entsprechenden Pakete vorher installieren (siehe Abschnitt 25.3, »Kernelmodule selbst kompilieren«).

Ob das Kompilieren und Laden der VirtualBox-Kernelmodule funktioniert hat, prüfen Sie mit dem folgenden Kommando:

Test

```
root# lsmod | grep vbox
vboxpci                24576  0
vboxnetadp             28672  0
vboxnetflt             28672  0
vboxdrv                434176 3  vboxnetadp,vboxnetflt,vboxpci
```

VirtualBox-Pakete von Oracle

Statt der mit Ihrer Distribution mitgelieferten VirtualBox-Pakete können Sie auch die von Oracle zum Download angebotene Version installieren. Das ist vor allem dann zweckmäßig, wenn Oracle eine neuere VirtualBox-Version anbietet als Ihre Distribution.

https://www.virtualbox.org/wiki/Linux_Downloads

	<p>Auf der obigen Website finden Sie VirtualBox in verschiedenen Formaten: als RPM- und Debian-Paket für diverse Distributionen sowie als Universal-Installer, den Sie wie folgt starten:</p> <pre>root# chmod u+x VirtualBox_nnn.run install root# ./VirtualBox_nnn.run install</pre>
Kernelmodule	<p>Nach Möglichkeit sollten Sie vor VirtualBox das dkms-Paket Ihrer Distribution installieren. In diesem Fall verwaltet DKMS die VirtualBox-Kernelmodule und kümmert sich bei Kernel-Updates automatisch um eine Neukompilierung. Bei meinen VirtualBox-Installationen hat das allerdings nicht immer zuverlässig funktioniert.</p> <p>Wenn DKMS nicht zur Verfügung steht bzw. versagt, kompilieren Sie die Kernelmodule selbst. Wie vorhin schon erwähnt, müssen Sie gegebenenfalls vorher den C-Compiler und die Kernel-Header-Dateien oder den Kernel-Quellcode installieren.</p> <pre>root# /usr/lib/virtualbox/vboxdrv.sh setup</pre>
APT-Paketquelle	<p>Für Debian- und Ubuntu-Anwender gibt es eine eigene APT-Paketquelle. Gegenüber der manuellen Installation eines einzelnen Pakets hat die Paketquelle den Vorteil, dass Sie innerhalb der gewählten Major-Version automatisch Updates erhalten. Dazu fügen Sie zu /etc/apt/sources.list eine der folgenden Zeilen hinzu:</p> <pre>deb https://download.virtualbox.org/virtualbox/debian stretch contrib deb https://download.virtualbox.org/virtualbox/debian xenial contrib</pre> <p>Anstelle von stretch bzw. xenial müssen Sie den Codenamen der von Ihnen eingesetzten Debian- bzw. Ubuntu-Distribution verwenden. Werfen Sie gegebenenfalls einen Blick in die Datei /etc/os-release.</p> <p>Außerdem führen Sie diese beiden Kommandos aus, um den Schlüssel der Paketquelle zu installieren:</p> <pre>root# wget -q https://www.virtualbox.org/download/oracle_vbox_2016.asc root# apt-key add oracle_vbox_2016.asc</pre> <p>Anschließend installieren Sie VirtualBox mit apt oder apt-get:</p> <pre>root# apt update root# apt install virtualbox-5.1</pre>
Yum-Paketquelle	<p>Für Anwender von Yum-kompatiblen Distributionen (CentOS, Fedora, openSUSE, Red Hat etc.) gibt es analog eine Yum-Paketquelle. Auch in diesem Fall müssen Sie zuerst den Schlüssel importieren:</p> <pre>root# wget -q https://www.virtualbox.org/download/oracle_vbox.asc root# rpm --import oracle_vbox.asc</pre>

<p>Anschließend laden Sie die für Ihre Distribution passende *.repo-Datei von der VirtualBox-Download-Seite herunter und kopieren sie in das Verzeichnis /etc/yum.repos.d. Die folgenden Zeilen zeigen die Fedora-Variante der *.repo-Datei:</p> <pre># Datei /etc/yum.repos.d/virtualbox.repo [virtualbox] name=Fedora \$releasever - \$basearch - VirtualBox baseurl=http://download.virtualbox.org/virtualbox/rpm/fedora/\$releasever/\$basearch enabled=1 gpgcheck=1 repo_gpgcheck=1 gpgkey=https://www.virtualbox.org/download/oracle_vbox.asc</pre> <p>Die VirtualBox-Installation führen Sie nun mit dnf install oder yum install oder zypper install durch.</p>

Vorbereitungsarbeiten

<p>VirtualBox richtet für jede virtuelle Maschine ein Unterverzeichnis innerhalb von VirtualBox VMs ein. In mehreren Dateien werden dort die Einstellungen der virtuellen Maschine sowie die virtuelle Festplatte gespeichert. Mit DATEI • GLOBALE EINSTELLUNGEN können Sie gegebenenfalls einen anderen Speicherort einstellen.</p> <p>Oracle bietet auf seiner Website ein sogenanntes Extension Pack zum Download an. Beim Download des Extension Packs schlägt der Webbrowser vor, die Datei direkt mit VirtualBox zu öffnen. Diesem Vorschlag folgen Sie einfach.</p> <p>Das Extension Pack ergänzt VirtualBox um einige Zusatzfunktionen: Unter anderem können Sie dann in den virtuellen Maschinen auf USB-Geräte (USB-2 und USB-3), PCI-Karten und Webcams zugreifen und die virtuellen Maschinen via RDP (Remote Display Protocol) auf einem anderen Rechner im Netzwerk steuern. Diese Erweiterungen werden nur in Binärform vertrieben, es handelt sich also nicht um Open-Source-Code. Die kommerzielle Nutzung dieser Erweiterungen erfordert eine Lizenz von Oracle!</p>
--

Speicherort für virtuelle Maschinen

Extension Pack

<p>Unabhängig davon, aus welcher Quelle Ihre VirtualBox-Installation stammt, wurde die Gruppe vboxusers eingerichtet. Nur Benutzer, die dieser Gruppe angehören, können in virtuellen Maschinen auf USB-Geräte zugreifen. Deswegen müssen Sie vor dem ersten Start von VirtualBox Ihren Account der Gruppe vboxusers hinzufügen. Ersetzen Sie beim folgenden Kommando kofler durch Ihren Login-Namen:</p>

vboxusers-Gruppe

```
root# usermod -a -G vboxusers kofler
```

Damit die geänderte Gruppenzuordnung wirksam wird, müssen Sie sich aus- und neu einloggen. Anschließend starten Sie die Benutzeroberfläche von VirtualBox über das KDE- oder Gnome-Menü bzw. mit dem Kommando VirtualBox.

VirtualBox unter Windows oder macOS installieren

Die Installation von VirtualBox unter Windows oder macOS ist grundsätzlich ein Kinderspiel: Sie laden das passende Setup-Programm von der VirtualBox-Seite herunter und führen es aus. Das Extension Pack muss auch in diesem Fall extra heruntergeladen und eingerichtet werden.

Unter Windows kann es allerdings passieren, dass VirtualBox nicht richtig funktioniert: Virtuelle Maschinen lassen sich dann nur im 32-Bit-Modus einrichten oder können gar nicht gestartet werden. Schuld daran ist in der Regel, dass VirtualBox die Virtualisierungstechnik VT nicht nutzen kann, die in viele Intel-CPU's integriert ist.

Dafür kann es mehrere Ursachen geben: Am wahrscheinlichsten ist es, dass Windows die Funktion durch Hyper-V blockiert. Abhilfe: Starten Sie das Programm WINDOWS-FEATURES, suchen Sie nach HYPER-V und deaktivieren Sie die Option. Danach muss der Rechner neu gestartet werden. Sollte Hyper-V nicht schuld sein, ist VT möglicherweise im BIOS/EFI deaktiviert. Und natürlich kann es auch sein, dass Ihre CPU die Funktion wirklich nicht enthält.

39.2 VirtualBox-Maschinen einrichten

Ist VirtualBox einmal installiert, können Sie mit dem Einrichten virtueller Maschinen beginnen. Dieser Abschnitt berücksichtigt sowohl Linux- als auch Windows-Gäste.

Eine virtuelle Maschine mit Linux einrichten

Dieser Abschnitt beschreibt, wie Sie innerhalb von VirtualBox eine virtuelle Maschine mit Linux einrichten. Dabei spielt es keine Rolle, ob VirtualBox selbst unter Linux, Windows oder macOS läuft.

Beim Einrichten einer neuen virtuellen Maschine unterstützt Sie ein Assistent. Als Betriebssystemtyp stehen neben Windows diverse Linux-Distributionen zur Auswahl. Wenn Ihre Distribution nicht vertreten ist, wählen Sie LINUX MIT KERNEL 2.6 / 3.x / 4.x; diese Einstellung gilt für alle aktuellen Kernelversionen. Achten Sie darauf, dass es für jedes Betriebssystem zwei Versionen gibt: eine für 32- und eine für 64-Bit-Installationen. Wählen Sie den passenden Eintrag!

VirtualBox sieht standardmäßig 1 GiB RAM für virtuelle Linux-Maschinen vor. Viele Desktop-Distributionen laufen flüssiger, wenn Sie etwas mehr RAM spendieren.

Als Nächstes müssen Sie eine virtuelle Festplatte einrichten. Der Datenträger wird als Image-Datei im Host-Dateisystem gespeichert. Dazu stehen verschiedene Forma-

te zur Auswahl. Im Regelfall sollten Sie beim VirtualBox-eigenen Format VDI bleiben und auch die Option DYNAMISCH ALLOZIERT beibehalten. Damit wird der Speicherplatz für die Festplatte erst nach und nach angefordert. Die Alternative FESTE GRÖSSE bedeutet, dass der gesamte Speicherplatz sofort vorreserviert wird.

Die vorgeschlagenen 8 GiB sind allerdings arg knapp bemessen. Bei vielen Distributionen reicht das nicht einmal für eine Minimalinstallation aus. Stellen Sie zumindest 16 GiB ein.

Schließlich zeigt VirtualBox eine Zusammenfassung aller Hardware-Komponenten an. Mit ÄNDERN können Sie nun bei Bedarf weitere Einstellungen durchführen, z. B. den Netzwerkzugang verändern oder im Dialogblatt MASSENSPEICHER eine ISO-Datei als Datenquelle für das DVD-Laufwerk auswählen.

Wenn Sie mit der Konfiguration fertig sind, starten Sie die virtuelle Maschine. Das von der ISO-Datei geladene Linux-Installationsprogramm erscheint in einem eigenen Fenster. Dort installieren Sie Linux wie auf einem realen Rechner.

Mögliche Fehlermeldungen beim ersten Start einer virtuellen Maschine

VirtualBox testet erst mit dem Start einer virtuellen Maschine, ob die VirtualBox-Kernelmodule geladen sind und ob Hardware-Virtualisierungsfunktionen zur Verfügung stehen. Ist eine dieser Voraussetzungen nicht erfüllt, wird eine Fehlermeldung oder Warnung angezeigt. Bei den Kernelmodulen müssen Sie sicherstellen, dass diese installiert sind. Wenn Sie VirtualBox frisch installiert haben, hilft es oft, das Script `/usr/lib/virtualbox/vboxdrv.sh` setup zum Neukompilieren der Module auszuführen. Denken Sie auch daran, dass die Hardware-Virtualisierungsfunktionen im BIOS oder EFI aktiviert sein müssen.

Die virtuelle Maschine erhält automatisch den Tastatur- und Mausfokus, sobald Sie eine Taste drücken. Standardmäßig lösen Sie den Fokus mit der rechten `[Strg]`-Taste. Im VirtualBox-Hauptfenster können Sie mit DATEI • EINSTELLUNGEN • EINGABE • VIRTUELLE MASCHINE eine andere »Host«-Taste einstellen. Die gerade gültige Kombination wird rechts in der Statusleiste des VirtualBox-Fensters angezeigt. Die wichtigsten Host-Tastenkombinationen sind in Tabelle 39.1 zusammengefasst.

Host-Tasten-kombination

Nachdem die eigentliche Installation abgeschlossen ist, sollten Sie in der virtuellen Maschine noch die sogenannten Guest Additions installieren. Sie stellen dem Gast-system zusätzliche Treiber zur Verfügung und verbessern das Zusammenspiel mit dem Wirt: Die Maus kann nun aus der virtuellen Maschine herausbewegt werden, die virtuelle Bildschirmauflösung des Gasts passt sich automatisch an die Fenstergröße an, der Datenaustausch mit dem Wirtssystem kann über Shared Folders erfolgen, Text kann über die Zwischenablage kopiert werden etc.

Gasterweiterungen installieren

Tastenkürzel	Bedeutung
Host	Tastatur- und Mausfokus lösen
Host + F	Vollbildmodus (de)aktivieren
Host + Entf	Strg + Alt + Entf an das Gastsystem senden
Host + ←	Strg + Alt + ← an das Gastsystem senden
Host + Fn	Strg + Alt + Fn an das Gastsystem senden
Host + S	Snapshot der virtuellen Maschine erstellen
Host + H	virtuelle Maschine per ACPI ausschalten
Host + R	virtuelle Maschine sofort ausschalten (Reset, Vorsicht!)

Tabelle 39.1 VirtualBox-Tastenkürzel

Manche Distributionen liefern fertige Pakete mit den VirtualBox-Gasterweiterungen mit. Bei openSUSE werden sie sogar gleich automatisch installiert. Allerdings sind diese Pakete selten auf dem aktuellen Stand. Sie bezahlen die Bequemlichkeit der Installation also möglicherweise mit Inkompatibilitäten zu der von Ihnen eingesetzten aktuelleren VirtualBox-Version.

- Debian, Ubuntu: virtualbox-guest-dkms, virtualbox-guest-utils, virtualbox-guest-x11
- Fedora mit RPMFusion: VirtualBox-guest-additions
- openSUSE: virtualbox-guest-kmp-default, virtualbox-guest-tools, virtualbox-guest-x11

Bei anderen Distributionen bzw. dann, wenn Sie die neueste Version der Gasterweiterungen benötigen, müssen Sie eine manuelle Installation durchführen. Dazu werfen Sie eine eventuell eingebundene CD/DVD aus und führen dann im VirtualBox-Fenster GERÄTE • GASTERWEITERUNGEN EINLEGEN aus. Im Regelfall erscheint nach einigen Sekunden in der virtuellen Maschine ein Dateimanagerfenster, in dem Sie autorun.sh starten. Sollte das nicht funktionieren, helfen die folgenden Kommandos weiter:

```
root# mkdir /media/cdrom
root# mount /dev/sr0 /media/cdrom
root# sh /media/cdrom/autorun.sh
```

Das Installationsprogramm richtet nun die drei neuen Kernelmodule vboxadd, vboxvideo und vboxvfs sowie einen neuen X-Treiber ein und fügt einige Init-Scripts hinzu, damit diese Gasterweiterungen beim nächsten Start der virtuellen Maschine auch verwendet werden.

Unter Ubuntu funktioniert die Installation der Gasterweiterungen auf Anhieb. Bei den meisten anderen Linux-Distributionen müssen Sie vor der Installation der Gasterweiterungen diverse Pakete installieren, die den C-Compiler und die Kernel-Header-Dateien enthalten. Führen Sie vorher ein Update aus, um sicherzustellen, dass die installierte Kernelversion und die Version der Kernel-Header-Dateien zusammenpassen!

```
root# yum install gcc make kernel-headers kernel-devel (CentOS)
root# apt install gcc make linux-headers-platform (Debian)
root# dnf install gcc make kernel-headers kernel-devel (Fedora)
root# zypper install gcc make kernel-source kernel-syms (openSUSE)
```

Fedora verwendet mitunter ganz aktuelle Versionen des Xorg-Servers, zu denen der VirtualBox-Grafiktreiber noch nicht kompatibel ist. Abhilfe schafft dann unter Umständen ein Downgrade auf eine ältere Xorg-Version vor der Installation der Gasttreiber. Das folgende Kommando zeigt, wie Sie unter Fedora 26 die Grafiktreiber von Fedora 25 installieren:

```
root# dnf --showduplicates --allowerase --releasever=25 \
downgrade xorg-x11-server-Xorg
```

Im Idealfall stehen innerhalb der virtuellen Maschine sogar 3D-Funktionen zur Verfügung. Dazu müssen auf jeden Fall die Gasterweiterungen aktiv sein, außerdem müssen die 3D-Funktionen in den Eigenschaften der virtuellen Maschine aktiviert sein (Dialogblatt ANZEIGE, Option 3D-BESCHLEUNIGUNG). Gleichzeitig sollten Sie den Grafikspeicher auf zumindest 64 MiB stellen (siehe Abbildung 39.1).

3D-Grafik

Das allein ist aber nicht in jedem Fall ausreichend – ob 3D-Funktionen an den Gast weitergereicht werden können, hängt auch davon ab, in welchem Host-Betriebssystem VirtualBox an sich läuft und welchen Grafiktreiber Sie im Host-System verwenden. Recht gute Erfahrungen habe ich mit Linux-Hosts in Kombination mit dem Intel-Grafiktreiber gemacht. In vielen anderen Fällen, insbesondere auch, wenn VirtualBox unter macOS läuft, funktionierte die 3D-Unterstützung gar nicht. Und selbst wenn die 3D-Funktionen prinzipiell durchgereicht werden, können fallweise Fehldarstellungen auftreten, z. B. nach der Veränderung der Fenstergröße.

Wenn Sie sich vergewissern möchten, ob alles funktioniert, installieren Sie in der virtuellen Maschine je nach Distribution das Paket mesa-utils, glx-utils oder Mesa-demo-x und führen dann glxinfo aus. Das Ergebnis sollte so wie im folgenden Listing aussehen:

```
user$ glxinfo | grep render
...
OpenGL renderer string: Chromium
```

Wenn der OpenGL renderer string hingegen llvmpipe enthält, dann werden die 3D-Funktionen durch die CPU emuliert, was spürbar langsamer ist.

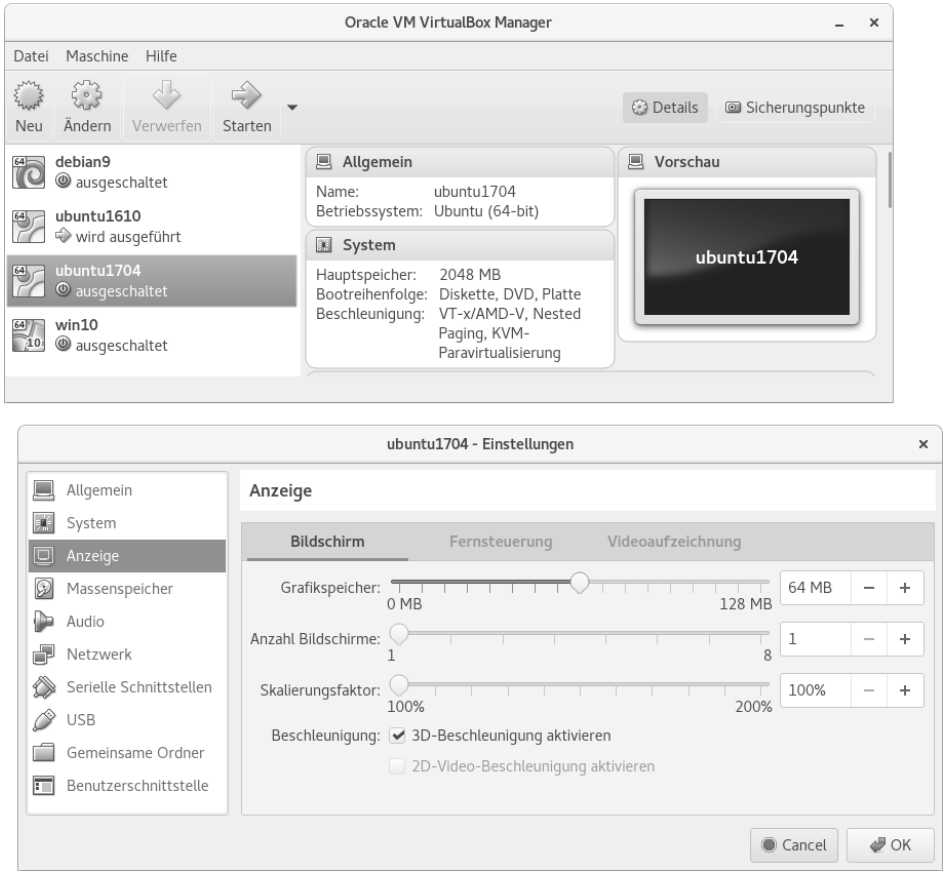


Abbildung 39.1 Überblick über alle virtuellen Maschinen (oben) und deren Einstellungen

Eine virtuelle Maschine mit Windows einrichten

Sofern Sie über eine Installations-CD/DVD bzw. die entsprechende ISO-Datei sowie eine gültige Lizenz und den dazugehörenden Schlüssel verfügen, können Sie in VirtualBox auch Windows installieren (siehe Abbildung 39.2). Die Installation von Windows und der VirtualBox-Gasterweiterungen verlief bei meinen Tests stets problemlos.

Warten Sie mit der Online-Registrierung so lange ab, bis Sie mit der Leistung zufrieden sind. Wenn Sie später in den Einstellungen der virtuellen Maschine das RAM vergrößern oder andere virtuelle Hardware-Parameter ändern, müssen Sie unter Umständen die Registrierung wiederholen!

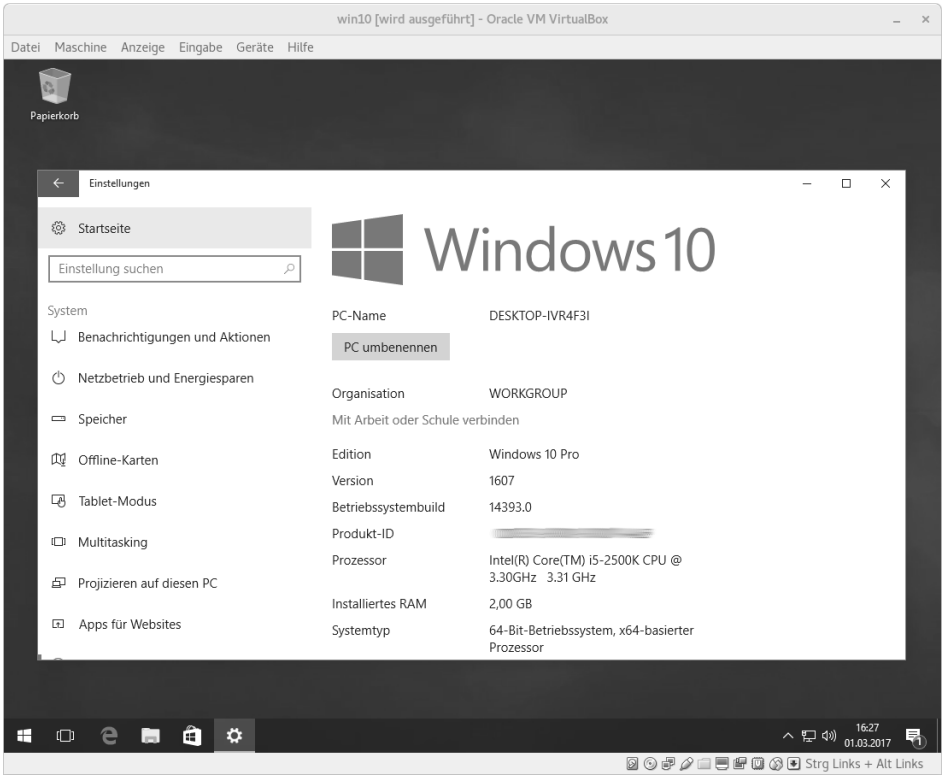


Abbildung 39.2 Windows 10 in einer virtuellen Maschine unter Linux ausführen

39.3 Arbeitstechniken und Konfigurationstipps

Dieser Abschnitt gibt Tipps zur Optimierung virtueller Maschinen sowie zum Datenaustausch zwischen dem Host-System bzw. dem »realen« lokalen Netzwerk und den virtuellen Maschinen.

Netzwerkconfiguration

VirtualBox stellt seinen Gästen die Netzwerkinfrastruktur des Wirts in Form einer virtuellen Netzwerkkarte zur Verfügung. Dabei existieren unterschiedliche Verfahren, wie der Netzwerkverkehr von der virtuellen Netzwerkkarte in das reale Netzwerk geleitet wird. Die entsprechenden Parameter finden Sie im Einstellungsdialog im Dialogblatt NETZWERK (siehe Abbildung 39.3). Entscheidend ist die Einstellung des Listenfelds ANGESCHLOSSEN AN, wobei der Vorgabewert NAT lautet.

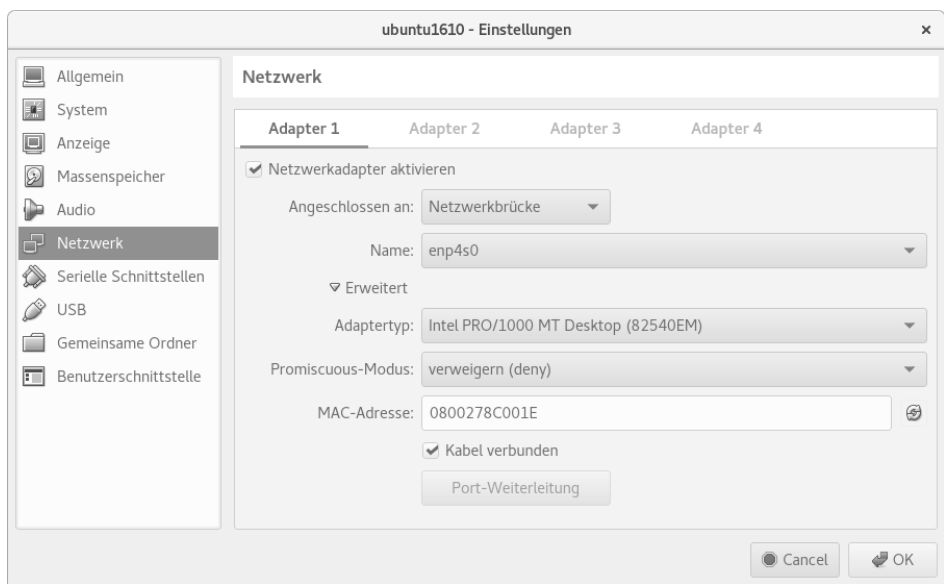


Abbildung 39.3 Netzwerkeinstellungen für die virtuelle Maschine

- **NAT:** Bei der NAT-Variante stellt VirtualBox seinen Gästen einen eigenen DHCP-Server zur Verfügung und realisiert Masquerading (NAT) (siehe auch Kapitel 27, »Internet-Gateway«). Auf diese Weise können die Gäste den Internetzugang des Wirtssystems nutzen. Ein Zugang zum lokalen Netzwerk ist wegen der unterschiedlichen Adressbereiche für das lokale Netz und das virtuelle NAT-Netz des Virtualisierungssystems unmöglich. Ebenso wenig können Sie vom Host eine SSH-Verbindung zum Gast herstellen. Die virtuellen Maschinen sind vom Host wie durch eine einfache Firewall getrennt.

Bei der NAT-Variante verwendet VirtualBox auf dem Host die IP-Adresse 10.0.2.2. Die virtuellen Maschinen erhalten andere 10.0.2.*-Adressen.

- **Netzwerkbrücke:** Bei dieser Variante erscheint der Gast als zusätzlicher Client im lokalen Netz. Diese Variante ist optimal, wenn es im lokalen Netzwerk einen DHCP-Server gibt bzw. wenn der Host-Rechner mit einem ADSL- oder WLAN-Router verbunden ist. Die virtuellen Gäste beziehen ihre Netzwerkkonfiguration dann über diesen Server/Router und können sowohl auf das lokale Netzwerk als auch auf das Internet zugreifen. Wenn Ihr Host-Rechner mehrere Netzwerkschnittstellen besitzt, müssen Sie angeben, welche Schnittstelle die Verbindung zum lokalen Netzwerk herstellt.

Im Büro ist diese Variante meine bevorzugte Konfiguration: Reale und virtuelle Maschinen sind damit im lokalen Netzwerk gleichwertige Partner, und der Datenaustausch via SSH, Samba etc. funktioniert unkompliziert. Beachten Sie aber, dass

die Netzwerkbrücke in manchen (Unternehmens-)WLANs nicht funktioniert. Die besten Erfahrungen habe ich mit dieser Konfigurationsvariante gemacht, wenn der Host-Rechner über ein Ethernet-Kabel (also nicht über WLAN) mit dem lokalen Netzwerk verbunden ist.

- **Host-only Adapter:** Bei dieser Variante kann der Gast über die Netzwerkfunktionen nur mit dem Wirt kommunizieren, nicht aber mit anderen Rechnern im lokalen Netzwerk oder mit dem Internet. Diese Variante ist dann zweckmäßig, wenn Sie ein von außen nicht zugängliches Testsystem aus mehreren virtuellen Maschinen aufbauen möchten.
- **Internes Netzwerk:** Hier bildet VirtualBox ein virtuelles Netzwerk, in dem ausschließlich virtuelle Maschinen kommunizieren können. Sie haben bei dieser Variante weder Zugriff auf das lokale Netzwerk noch auf das Internet.

Sie können virtuelle Maschinen mit bis zu vier Netzwerkadaptern ausstatten. Das gibt Ihnen die Möglichkeit, mehrere Konfigurationsvarianten parallel zu verwenden – z. B. einen NAT-Adapter, damit die virtuellen Maschinen Internetzugang erhalten, und einen Host-only-Adapter, damit Sie eine SSH-Verbindung zwischen den virtuellen Maschinen und dem Host-Rechner herstellen können.

Die Netzwerkkonfiguration kann im laufenden Betrieb geändert werden! Es ist also nicht erforderlich, die virtuelle Maschine bei jeder Änderung neu zu starten. Der schnellste Weg in den Konfigurationsdialog führt über das Icon AKTIVITÄT DER NETZWERKADAPTER in der Statusleiste des VirtualBox-Fensters.

Datenaustausch über die Zwischenablage

In den Einstellungen der virtuellen Maschine können Sie im Dialogblatt ALLGEMEIN • ERWEITERT für die gemeinsame Zwischenablage und für die Funktion Drag & Drop den Modus BIDIREKTIONAL aktivieren. Beide Optionen setzen auf jeden Fall voraus, dass in der virtuellen Maschine die Gasterweiterungen installiert sind.

Diese Konfiguration gibt Ihnen die Möglichkeit, über die Zwischenablage Text zwischen dem Host und dem Gast zu kopieren. Außerdem können Sie nun per Drag & Drop Dateien zwischen einem Dateimanager im Host und einem Dateimanager im Gast hin- und herkopieren. Fallweise hat dies bei meinen Tests gut funktioniert, aber ganz ausgereift wirkt diese Funktion noch nicht.

Datenaustausch mit einem Shared Folder

Ein zuverlässigerer Weg zum Datenaustausch zwischen Wirt und Gast sind sogenannte Shared Folder. Zur Konfiguration öffnen Sie mit ÄNDERN den Einstellungsdialog, wechseln in das Dialogblatt GEMEINSAME ORDNER, wählen dann ein lokales Verzeichnis auf dem Wirtssystem aus und geben dem Ordner einen Namen (z. B. myshare). Das

Host-Konfiguration

Verzeichnis gilt spezifisch für eine bestimmte virtuelle Maschine. Für Windows-Gäste aktivieren Sie auch gleich die Option AUTOMATISCH EINBINDEN.

Linux-Gäste Nach einem Neustart eines Linux-Gastsystems ist nun ein manuelles mount-Kommando erforderlich, um auf das gemeinsame Verzeichnis zugreifen zu können. Dabei müssen Sie myshare durch den Namen ersetzen, den Sie bei der Konfiguration verwendet haben.

```
root@gast# mkdir /media/vbox-share
root@gast# mount -t vboxsf myshare /media/vbox-share
```

Wenn Linux die Fehlermeldung *unknown filesystem vboxsf* liefert, sind die VirtualBox-Gasterweiterungen nicht richtig installiert. Abhilfe schafft bei den meisten Distributionen die Installation des Pakets *virtualbox-guest-utils*.

Windows-Gäste In Windows-Gästen finden Sie das gemeinsame Verzeichnis im Explorer als Netzwerkverzeichnis des virtuellen Rechners *vboxsrv*. Wenn Sie bei der Konfiguration die Option AUTOMATISCH EINBINDEN verwendet haben, dann wird dem Verzeichnis unter Windows auch gleich ein eigener Laufwerksbuchstabe zugeordnet.

USB-Geräte in virtuellen Maschinen

Sofern Sie auf dem Host das VirtualBox Extension Pack installiert haben, können Sie USB-Geräte auch in virtuellen Maschinen nutzen. Das funktioniert nur, wenn das USB-Gerät im Wirtssystem *nicht* verwendet wird. USB-Datenträger werden im Wirtssystem normalerweise automatisch in das Dateisystem eingebunden; Sie müssen sie wieder aus ihm lösen, um sie im Gast verwenden zu können.

Eine weitere Voraussetzung besteht darin, dass der Benutzer, der VirtualBox ausführt, Mitglied der Gruppe *vboxusers* ist. Schließlich müssen Sie darauf achten, dass der USB-CONTROLLER bei den Einstellungen der virtuellen Maschine im Dialogblatt USB aktiviert ist. In diesem Dialogblatt können Sie auch einen Filter definieren, um ein USB-Gerät direkt einer virtuellen Maschine zuzuordnen. Das ist aber keine zwingende Voraussetzung. Sie können das USB-Gerät nach dem Einschalten auch dynamisch in der VirtualBox-Statusleiste beim USB-Icon der virtuellen Maschine zuordnen (siehe Abbildung 39.4).

Generell funktionierten die von mir getesteten USB-Geräte (ein Scanner und eine Digitalkamera) in den virtuellen Maschinen anstandslos, wenn auch langsamer als im Wirtssystem.

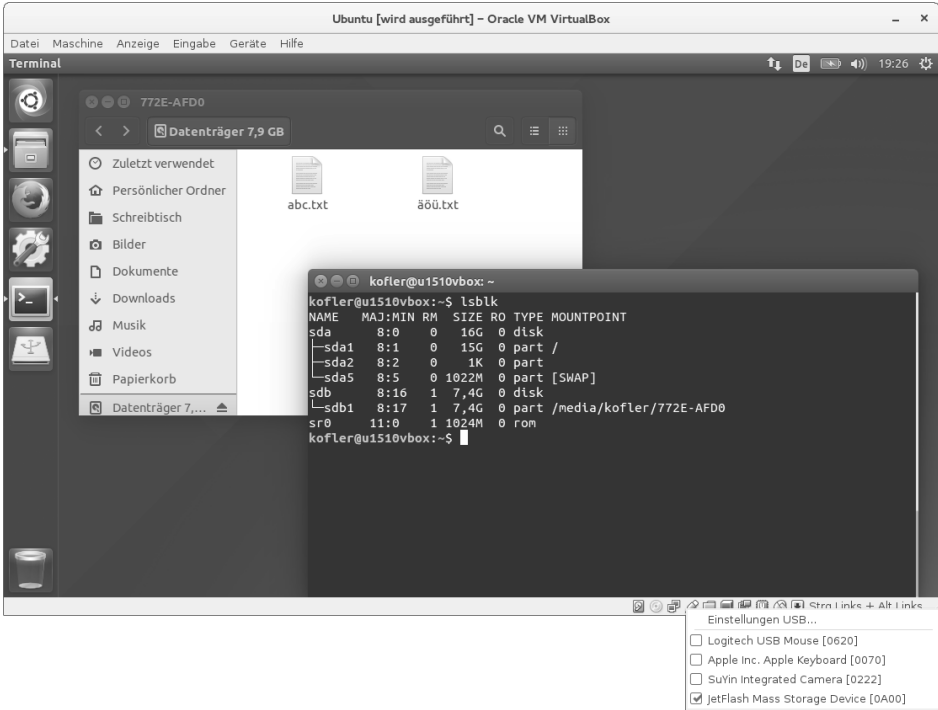


Abbildung 39.4 Die virtuelle Ubuntu-Maschine kann auf einen USB-Stick zugreifen, der an das Notebook angesteckt wurde, auf dem VirtualBox ausgeführt wird.

Export/Import virtueller Maschinen

Um eine virtuelle Maschine weiterzugeben, erzeugen Sie mit DATEI • APPLIANCE EXPORTIEREN eine sogenannte Virtual Appliance, also eine zur Weitergabe bestimmte virtuelle Maschine, die üblicherweise aus zwei Dateien besteht: *.ovf enthält eine Beschreibung der virtuellen Maschine, *.vmdk das Festplatten-Image in komprimierter Form. Diese virtuelle Maschine können Sie nun bei einer anderen VirtualBox-Installation mit DATEI • APPLIANCE IMPORTIEREN wieder einrichten.

Eine virtuelle Maschine auf einen anderen Host übertragen

Wenn es Ihnen nur darum geht, eine oder mehrere virtuelle Maschinen von einem Rechner auf einen anderen zu übertragen, können Sie sich die Umwandlung in eine Virtual Appliance sparen. In diesem Fall reicht es aus, das betreffende Verzeichnis *VirtualBox VMs/vm-name* zu kopieren. Anschließend führen Sie in VirtualBox das Kommando *MASCHINE • HINZUFÜGEN* aus und wählen die *.vbox-Datei aus.

Geschwindigkeitsoptimierung

Mit zwei Optionen bei der Einstellung der virtuellen Hardware können Sie ein klein wenig mehr Geschwindigkeit aus Ihren virtuellen Maschinen herauskitzeln:

- **Host-Caching für die virtuelle Festplatte:** Im Dialogblatt MASSENSPEICHER der virtuellen Maschine können Sie für den SATA-Controller die Option HOST-I/O-CACHE VERWENDEN aktivieren. Sie erreichen damit, dass Schreibzugriffe zwischengespeichert werden, was die Geschwindigkeit I/O-lastiger Vorgänge stark vergrößern kann. Der Nachteil: Sollte der Host-Rechner abstürzen, riskieren Sie ein beschädigtes Dateisystem in der virtuellen Maschine.
- **Paravirtualisierte Netzwerktreiber:** Sofern es sich bei der virtuellen Maschine um eine Linux-Distribution handelt, können Sie im Dialogblatt NETZWERK bei den erweiterten Einstellungen die Option PARAVIRTUALISIERTES NETZWERK (VIRTIO-NET) aktivieren. VirtualBox spielt der virtuellen Maschine nun nicht mehr die Logik eines Netzwerkadapters vor, sondern spricht direkt mit dem virtio-net-Treiber des Linux-Kernels. Das ist deutlich effizienter.

Virtuelle Festplatten vergrößern

Linux-Gast Die Benutzeroberfläche von VirtualBox gibt Ihnen leider keine Möglichkeit, eine virtuelle Festplatte nachträglich zu vergrößern. Wo die Benutzeroberfläche versagt, hilft oft auch ein Kommando weiter – so auch in diesem Fall. Bevor Sie loslegen, müssen Sie Ihre virtuelle Maschine herunterfahren. Ein vollständiges Backup ist sehr zu empfehlen!

Anschließend suchen Sie die *.vdi-Datei der virtuellen Festplatte und wenden darauf das Kommando `vboxmanage` an. Mit der Option `--resize` geben Sie die gewünschte neue Größe in MiB an. Im Regelfall wird das Kommando blitzschnell ausgeführt.

```
root# vboxmanage modifyhd debian.vdi --resize 60000
```

Das ist aber erst die halbe Miete. Die virtuelle Maschine weiß nämlich noch nichts davon, dass ihre Festplatte größer geworden ist. Bei einer Gast-Installation ohne LVM und mit ext4- oder xfs-Dateisystemen binden Sie nun ein ISO-Image einer Linux-Live-CD in das virtuelle CD/DVD-Laufwerk ein und starten innerhalb der virtuellen Maschine ein Live-System. Dort führen Sie `parted /dev/sda` aus und können nun die Größe der letzten Partition erhöhen. Anschließend müssen Sie auch das darin enthaltene Dateisystem mit `resize2fs` oder `xfs_growfs` vergrößern.

Wenn Sie im Linux-Gast hingegen LVM oder btrfs-Dateisysteme verwenden, können Sie das Dateisystem im laufenden Betrieb vergrößern. Diese Eingriffe sind natürlich nicht ganz ungefährlich. Lesen Sie vorher die relevanten Abschnitte aus Kapitel 22, »Administration des Dateisystems«!

Analog kann auch ein Windows-Dateisystem vergrößert werden. Auch in diesem Fall fahren Sie die virtuelle Maschine zuerst herunter und vergrößern die *.vdi-Datei mit dem Kommando `vboxmanage`. Dann starten Sie Windows, öffnen darin ein Eingabeaufforderungsfenster mit Administratorrechten und führen die folgenden Kommandos aus:

Windows-Gast

```
> Diskpart
list disk
select disk 0
list partition
select partition 2
extend
```

`list disk` liefert eine Liste aller virtuellen Festplatten. Normalerweise muss die erste Platte mit dem Index 0 ausgewählt werden. Nun ermittelt `list partition` die Partitionen. Abermals muss mit `select` eine Partition zur weiteren Bearbeitung ausgewählt werden – im Regelfall die letzte. Mit `extend` wird diese nun auf die maximale Größe erweitert.

Virtuelle Maschinen unsichtbar ausführen

Normalerweise wird jede laufende virtuelle Maschine in einem eigenen Fenster angezeigt. Beim Schließen des Fensters haben Sie die Wahl, den Status der virtuellen Maschine zu speichern (die virtuelle Maschine also gewissermaßen zu pausieren), sie per ACPI herunterzufahren oder sie gewaltsam zu stoppen (wie durch das Lösen eines Netzkabels).



Abbildung 39.5 Das Menü des Start-Buttons enthält zwei versteckte Einträge zum Start der virtuellen Maschine ohne bzw. mit abkoppelbarer Oberfläche.

Mitunter wäre es aber praktisch, virtuelle Maschinen unsichtbar, also *ohne* eigenes Fenster auszuführen. Das gilt besonders für Server-Installationen, die ohnedies im Textmodus laufen. Für derartige virtuelle Maschinen können Sie beim Start-Button den Menüeintrag OHNE GUI STARTEN wählen.

Noch mehr Flexibilität gibt der Eintrag ABKOPPELBARER START (siehe Abbildung 39.5). Damit wird die virtuelle Maschine beim Start wie üblich in einem Fenster angezeigt. Mit MASCHINE • GUI ABKOPPELN können Sie das Fenster dann aber bei Bedarf schließen, ohne die virtuelle Maschine zu stoppen. Mit einem Doppelklick auf das Symbol der virtuellen Maschine im VirtualBox-Hauptfenster können Sie die Benutzeroberfläche der virtuellen Maschine sogar wiederbeleben.

39.4 Vagrant

Das Einrichten einer neuen virtuellen Maschine ist mit relativ viel Handarbeit verbunden. Solange es nur um eine Installation geht, ist das kein großes Problem. Wenn Sie aber regelmäßig virtuelle Maschinen einrichten müssen und dabei womöglich Wert darauf legen, dass die virtuellen Maschinen reproduzierbar exakt gleich konfiguriert sind, sollten Sie sich mit dem Programm *Vagrant* anfreunden. Vagrant ist ein Werkzeug, das beim Einrichten, Ausführen, Steuern und Stoppen von virtuellen Umgebungen hilft.

Vagrant wird zusammen mit einigen weiteren Programmen (Atlas, Packer, Vault, Nomad, Consul) von der Firma Hashicorp entwickelt. Alle Produkte verwenden Open-Source-Lizenzen und stehen kostenlos zur Verfügung. Zum Teil gibt es darüber hinaus Enterprise-Varianten mit Zusatzfunktionen für zahlende Kunden.

<https://www.hashicorp.com/#open-source-tools>

Vagrant ist unabhängig von der Betriebssystem- und Virtualisierungsplattform!

Auch wenn ich Ihnen Vagrant hier im VirtualBox-Kapitel vorstelle, kommt das Programm auch mit anderen Virtualisierungssystemen zurecht, z.B. mit VMware und Hyper-V. Vagrant unterstützt auch Cloud-Systeme wie AWS sowie Docker. Für die libvirt-Werkzeuge, die ich in Kapitel 40, »KVM«, vorgestellt habe, gibt es auf GitHub einen Provider, der allerdings Mitte 2017 noch nicht vollständig ausgereift war. Vagrant lässt sich problemlos auch unter Windows und macOS installieren.

Nomenklatur Die Dokumentation zu Vagrant ist leichter zu verstehen, wenn Sie sich zuerst mit einigen Begriffen vertraut machen:

- **Vagrant-Datei:** Vagrant richtet virtuelle Maschinen auf der Basis einer Vagrant-Datei und einer Box ein. Die Textdatei *Vagrantfile* gibt die Quelle der Box-Datei an und beschreibt, welche Operationen auf die Box angewendet werden müssen, um die virtuelle Maschine fertigzustellen. Dieser einmalig durchzuführende Vorgang wird »Provisioning« genannt. Die Anweisungen in der Vagrant-Datei werden in der Syntax der Programmiersprache Ruby angegeben. Die Vagrant-Datei kann beispielsweise Kommandos zum Einrichten der Netzwerkverbindung und des SSH-Servers enthalten. Sie können aber auch externe Scripts aufrufen, die zur Installation von Zusatz-Software oder für Konfigurationsarbeiten in der virtuellen Maschine auf Werkzeuge wie *Puppet* oder *Chef* zurückgreifen.
- **Boxes:** Eine Box ist eine komprimierte Datei, die eine virtuelle Maschine enthält. Box-Dateien sind wegen des inkludierten Festplatten-Images zumeist recht groß (mehrere Hundert MiB). Auf der Webseite <https://atlas.hashicorp.com/boxes> finden Sie einen Katalog kostenlos verfügbarer Vagrant-Boxes. Vagrant kommt aber auch mit Boxes zurecht, die lokal gespeichert sind oder auf anderen Webservern zugänglich sind.

Beim ersten Start wird die virtuelle Maschine zuerst aus der Box geklont; anschließend führt Vagrant die in *Vagrantfile* aufgezählten Konfigurationsarbeiten durch, führt am Klon also noch Änderungen durch. Die Box selbst bleibt dabei unverändert und kann später neuerlich als Basis verwendet werden, wenn weitere Instanzen erzeugt werden sollen oder die virtuelle Maschine neu eingerichtet werden soll.
- **Vagrant-Kommando:** Die gesamte Administration von Vagrant erfolgt durch das Kommando *vagrant*. Damit starten und stoppen Sie virtuelle Maschinen, stellen SSH-Verbindungen zu ihnen her etc.
- **Provider:** Vagrant verwendet standardmäßig VirtualBox als Virtualisierungssystem. Sogenannte Provider stellen optionale Schnittstellen zu anderen Virtualisierungssystemen her. Einige Provider sind standardmäßig in Vagrant enthalten, andere können extra installiert werden.
- **Plugins:** Vagrant hat einen modularen Aufbau. Selbst etliche Grundfunktionen sind als Plugins realisiert. Zur Realisierung von Zusatzfunktionen können Sie Vagrant durch externe Plugins erweitern (*vagrant plugin install name*).

Bei vielen Distributionen installieren Sie Vagrant am einfachsten mit den Paketverwaltungskommandos. Allerdings erhalten Sie damit selten die aktuellste Version. Auf der Vagrant-Webseite <https://www.vagrantup.com> finden Sie aktuelle Pakete im Debian- und RPM-Format, deren Installation in der Regel auf Anhieb aus dem Webbrowser heraus gelingt. Nach der Installation können Sie mit *vagrant version* die Versionsnummer feststellen:

Installation

```
user$ vagrant version
Installed Version: 1.9.2
Latest Version: 1.9.2
```

Base Boxes Ich gehe in diesem Buch nur auf die Nutzung und Modifizierung vorgefertigter Boxes ein. Fortgeschrittene Vagrant-Anwender können aber auch vollkommen neue Boxes erzeugen. In der Regel ist es zweckmäßig, dabei eine sogenannte »Base Box« einzurichten, also eine virtuelle Maschine, die auf einer minimalen Installation der jeweiligen Distribution basiert und die speziell für Vagrant vorkonfiguriert ist. Eine typische Vagrant-Konfiguration besteht aus einem SSH-Server, einem vagrant-Benutzer mit sudo-Rechten ohne Passwort und eventuell der Installation von Gasterweiterungen für das gewünschte Virtualisierungssystem. Eine ausführliche Anleitung, wie Sie Vagrant-kompatible Base Boxes einrichten, finden Sie hier:

<https://www.vagrantup.com/docs/boxes/base.html>

Hello World!

Um Vagrant auszuprobieren, greifen Sie am besten auf eine der vielen vorgefertigten Vagrant-Boxes zurück. Der Boxes-Katalog auf <https://atlas.hashicorp.com/boxes> enthält leider nicht viel mehr als den Namen der jeweiligen Box und eine Liste der unterstützten Provider. Unbegreiflicherweise fehlt eine Beschreibung, welche Zielsetzung die jeweilige Box hat. Nicht einmal die Größe der Box ist dokumentiert.

Für erste Experimente können Sie z. B. die Box ubuntu/xenial64 verwenden. Sie enthält einen tagesaktuellen Build einer Minimalinstallation von Ubuntu 16.04 für den Server-Einsatz (also ohne grafische Benutzeroberfläche):

```
user$ mkdir u1604
user$ cd u1604
user$ vagrant init ubuntu/xenial64
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment!
user$ vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Adding box 'ubuntu/xenial64' (v20170303.1.0)
    default: Downloading: https://atlas.hashicorp.com/ubuntu/ \
        boxes/xenial64/versions/20170303.1.0/providers/virtualbox.box
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Mounting shared folders...
    default: /vagrant => /home/kofler/u1604
...
user$ vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)
```

Kurz einige Erklärungen zu den obigen Kommandos, deren Ausgaben aus Platzgründen stark gekürzt abgedruckt sind. vagrant init lädt vom Hashicorp-Server die Datei Vagrantfile für die gewünschte virtuelle Maschine. Das geht schnell, da die Datei nur wenige Kilobyte groß ist. Sie wird im gerade aktuellen Verzeichnis gespeichert. vagrant init verwendet standardmäßig VirtualBox als Virtualisierungsplattform. Wenn Sie ein anderes System verwenden möchten, wählen Sie dieses mit --provider name aus.

vagrant up startet die virtuelle Maschine. Ab dem zweiten Mal wird auch dieses Kommando recht schnell ausgeführt, beim ersten Mal dauert es aber geraume Zeit: Zuerst muss nämlich die Box für die virtuelle Maschine heruntergeladen werden. Diese Box sowie diverse Zusatzdateien werden im Verzeichnis /.vagrant.d/boxes gespeichert, also getrennt von dem Verzeichnis, in dem sich Vagrantfile befindet. Das hat den Vorteil, dass später bei Bedarf weitere virtuelle Maschinen auf Basis der bereits vorhandenen Box eingerichtet werden können. Für ubuntu/xenial64 beträgt der Platzbedarf der Box ca. 275 MiB.

Sobald die Box heruntergeladen ist, wird die entsprechende virtuelle Maschine eingerichtet. Im obigen Beispiel verwendet Vagrant den Default-Provider für VirtualBox. Die Dateien der virtuellen Maschine landen daher in dem von VirtualBox vorgesehenen Verzeichnis. Wenn Sie die VirtualBox-Defaulteinstellungen nicht verändert haben, ist das VirtualBox VMs in ihrem Heimatverzeichnis. Damit gibt es nun Dateien an drei verschiedenen Orten:

- ▶ in Ihrem eigenen Vagrant-Verzeichnis: Es enthält neben Vagrantfile einige weitere Konfigurationsdateien und beansprucht nur wenig Speicherplatz. Alle vagrant-Kommandos müssen in diesem Verzeichnis oder in einem seiner Unterverzeichnisse ausgeführt werden.
- ▶ in /.vagrant.d/boxes: Das Verzeichnis enthält je eine Box für alle irgendwann mit Vagrant eingerichteten Maschinen. Der Platzbedarf beträgt typischerweise einige Hundert MiB pro Box.
- ▶ in VirtualBox VMs: Dieses Verzeichnis enthält die virtuellen Maschinen inklusive der Disk-Images für jede mit Vagrant eingerichtete Maschine. Der Platzbedarf ist hoch und beträgt oft mehrere GiB pro virtueller Maschine.

Die von Vagrant eingerichtete VirtualBox-Maschine wird im VirtualBox-Hauptfenster zwischen selbst erzeugten virtuellen Maschien aufgelistet. Ihr Name endet immer mit einer zufällig generierten Zahl (siehe Abbildung 39.6). vagrant up startet die virtuelle Maschine unsichtbar, also ohne ein VirtualBox-Fenster zu öffnen. Zwar ist es möglich, per Doppelklick auf die Liste der virtuellen Maschinen ein entsprechendes Fenster zu öffnen; im Normalfall ist es aber üblich, Vagrant-Maschinen per SSH zu administrieren.

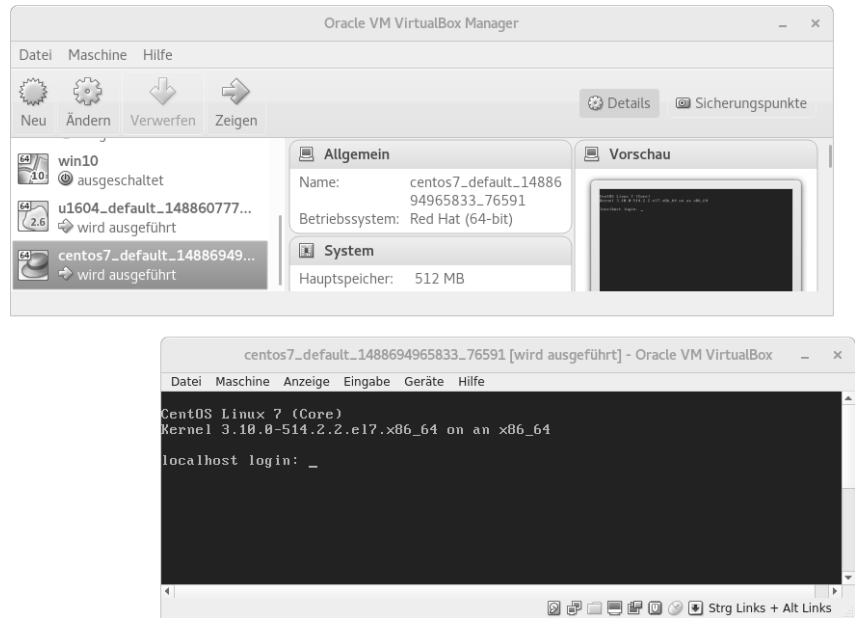


Abbildung 39.6 Die von Vagrant eingerichteten virtuellen Maschinen sind in der Liste der VirtualBox-Maschinen an der Namenserverweiterung »default_nnn« zu erkennen.

Bento-Boxes

Das Boxes-Angebot auf <https://atlas.hashicorp.com/boxes> ist leider ziemlich unübersichtlich. Wenn Sie auf der Suche nach kleinen, vernünftig vorkonfigurierten Boxes für die wichtigsten Linux-Distributionen sind, lohnt sich ein Blick auf die Webseite <http://chef.github.io/bento/>! Zur Verwendung einer derartigen Box führen Sie einfach `vagrant init bento/<name>` aus.

Netzwerkconfiguration

Vagrant-Maschinen verwenden in VirtualBox einen NAT-Netzwerkadapter. Das ist aus Sicherheitsgründen zweckmäßig, weil in Vagrant-Maschinen üblicherweise der Account `vagrant` mit einem gleichnamigen Passwort eingerichtet ist. Wäre die virtuelle Maschine im lokalen Netz oder gar im Internet öffentlich erreichbar, würde sie unweigerlich das Ziel von Hacker-Angriffen.

Damit zwischen dem Host-Rechner und der virtuellen Maschine eine SSH-Verbindung möglich ist, richtet Vagrant standardmäßig eine Port-Umleitung zwischen dem Port 22 der virtuellen Maschine und dem Port 2222 des Hosts her. Ist dieser Port schon von einer anderen Box belegt, sucht `vagrant` selbstständig einen anderen freien Port mit der Nummer `22nn`.

Um eine SSH-Verbindung zur virtuellen Maschine herzustellen, führen Sie einfach das Kommando `vagrant ssh` aus. Vagrant startet den SSH-Client dann mit den richtigen Optionen. Sie brauchen kein Passwort anzugeben. In der virtuellen Maschine werden Sie in der Regel als Benutzer `vagrant` bzw. beim hier vorgestellten Beispiel als Benutzer `ubuntu` angemeldet. Dank einer in `/etc/sudoers.d` vorgesehenen Konfigurationsdatei erlangen Sie mit `sudo -s` ohne Passwort root-Rechte:

```
user@hostsystem$ vagrant ssh
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)
ubuntu@ubuntu-xenial64:~$ sudo -s
root@ubuntu-xenial64:~# cat /etc/sudoers.d/90-cloud-init-users
# User rules for ubuntu
ubuntu ALL=(ALL) NOPASSWD:ALL
```

Bei vielen Boxes richtet Vagrant darüber hinaus ein gemeinsames Verzeichnis zwischen Host und virtueller Maschine ein. Auf dem Host wird dazu das Verzeichnis verwendet, in dem sich `Vagrantfile` befindet. Auf dem Client ist der Shared Folder in Linux-Gästen üblicherweise unter `/vagrant` zugänglich.

Beim vorgestellten Ubuntu-Beispiel ist das zum Datenaustausch vorgesehene Verzeichnis als VirtualBox Shared Folder realisiert. In der Box sind dazu standardmäßig die VirtualBox-Gasterweiterungen installiert. Andere von Vagrant unterstützte Verfahren zur Realisierung des gemeinsamen Verzeichnisses basieren auf Rsync (Synchronisierung nur beim Start), NFS oder SSHFS (`vagrant-sshfs`-Plugin).

Default-Login

Es ist üblich, dass Vagrant-Maschinen einen Default-Account mit dem Login-Namen `vagrant` und einem gleichnamigen Passwort haben. Dieser Benutzer wird auch für SSH-Verbindungen verwendet, wobei die Authentifizierung über eine Schlüsseldatei erfolgt.

Die in diesem Beispiel vorgestellte Ubuntu-Box widerspricht leider den Vagrant-Empfehlungen: In diesem Fall lautet der Default-Login `ubuntu`. Als Passwort wird ein zufälliger hexadezimaler Code verwendet. Um ein eigenes Passwort einzustellen, stellen Sie mit `vagrant ssh` eine Verbindung zur virtuellen Maschine her und führen dann `passwd` aus.

Administration

Die gesamte Administration von Vagrant erfolgt mit dem gleichnamigen Kommando (siehe Tabelle 39.2). Soweit sich die gewünschte Operation auf eine Box bezieht, sucht `vagrant` zuerst im aktuellen Verzeichnis nach `Vagrantfile`, danach in allen übergeordneten Verzeichnissen.

vagrant wird in der Regel ohne root-Rechte ausgeführt. vagrant -h liefert eine Liste aller Kommandos. vagrant kommando -h zeigt weiterführende Informationen zum betreffenden Kommando an.

Kommando	Bedeutung
vagrant box list	heruntergeladene Vagrant-Boxes auflisten
vagrant box update	Vagrant-Box aktualisieren
vagrant destroy	Vagrant-Maschine löschen
vagrant halt	Vagrant-Maschine herunterfahren
vagrant init name	vorgefertige Vagrant-Datei herunterladen
vagrant login	Login zu eigenem Atlas-Account durchführen
vagrant plugin install name	Plugin installieren
vagrant provision	Provisioning wiederholen
vagrant resume	pausierte Vagrant-Maschine wieder aktivieren
vagrant share	Vagrant-Maschine öffentlich zugänglich machen
vagrant ssh	SSH-Verbindung zur Vagrant-Maschine herstellen
vagrant status	Status der Vagrant-Maschine anzeigen
vagrant suspend	Vagrant-Maschine pausieren
vagrant up	Vagrant-Maschine starten

Tabelle 39.2 Wichtige vagrant-Kommandos

VagrantFile

Die Datei VagrantFile beschreibt die Konfiguration der virtuellen Maschine, die Vagrant einrichten soll. Im einfachsten Fall sind drei Zeilen ausreichend, die einfach den Ort der zugrunde liegenden Vagrant-Box auf dem Hashicorp-Server angeben. "2" bedeutet, dass die Vagrant-Datei die Syntax von Version 2 verwendet. config.vm.box gibt den Namen der Box an. Vagrant sucht üblicherweise im Hashicorp-Katalog nach der Box und lädt sie von dort herunter. Wenn sich die Box auf einem anderen Server oder in einem lokalen Verzeichnis befindet, geben Sie diesen Ort zusätzlich mit config.vm.box_url an. Für lokale Dateien verwenden Sie dabei die Syntax "file:///pfad/name.box".

```
# VagrantFile für ubuntu/xenial64
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
end
```

Nicht explizit in VagrantFile aufgeführt sind die Operationen zum Einrichten der Port-Umleitung für den SSH-Server sowie für die Synchronisation des gemeinsamen Verzeichnisses. Darum kümmert sich Vagrant automatisch, sofern dies nicht durch anderslautende Optionen verhindert wird.

Im Folgenden stelle ich Ihnen exemplarisch einige Optionen für VagrantFile vor. Eine vollständige Referenz finden Sie in der Vagrant-Dokumentation:

<https://www.vagrantup.com/docs/vagrantfile>

Einfache Änderungen an VagrantFile werden wirksam, wenn Sie die virtuelle Maschine einfach nur neu starten:

VagrantFile-
Änderungen
aktivieren

```
root# vagrant reload
```

Alle mit config.vm.provision definierten Konfigurationsarbeiten erfordern aber die Option --provision. Damit erzwingen Sie eine neuerliche Konfiguration der virtuellen Maschine. (Normalerweise wird das sogenannte Provisioning ja nur beim ersten Start durchgeführt.)

```
user$ vagrant reload --provision
```

Alternativ können Sie das Provisioning auch im laufenden Betrieb durchführen bzw. wiederholen. Dabei können Sie mit der Option --provision-with einschränken, welchen Typ von Provisioning-Maßnahmen (z. B. shell oder file) bzw. welche benannte Provisioning-Anweisung Sie ausführen möchten:

```
user$ vagrant provision --provision-with shell
```

Bei komplexen Änderungen, für die Vagrant in der richtigen Reihenfolge mehrere Scripts ausführen muss, kann es sogar erforderlich sein, dass Sie die virtuelle Maschine mit vagrant destroy löschen und dann vollständig neu einrichten müssen:

```
user$ vagrant destroy
user$ vagrant up
```

config.vm.hostname legt den Hostnamen der virtuellen Maschine fest.

Hostname

```
config.vm.hostname = "vagrant-u1604"
```

Mit config.vm.network können Sie diverse Parameter der Netzwerkkonfiguration verändern. Die folgende Zeile bewirkt eine Port-Umleitung vom Port 80 der virtuellen Maschine auf den Port 8080 des Hosts:

Portumleitung

```
config.vm.network "forwarded_port", guest: 80, host: 8080
```

Standardmäßig teilt Vagrant das Projektverzeichnis, also das Verzeichnis des Hosts, in dem sich Vagrantfile befindet, im Gast als /vagrant. Bei Bedarf können Sie das verhindern:

Gemeinsame
Verzeichnisse

VirtualBox-
spezifische
Konfiguration

```
config.vm.synced_folder ".", "/vagrant", disabled: true
```

Umgekehrt können Sie mit `config.vm.synced_folder` weitere gemeinsame Verzeichnisse einrichten. Dabei bezieht sich der erste Parameter auf den Host-Rechner (relativ zum Projektverzeichnis), der zweite Parameter auf den Gast.

```
config.vm.synced_folder "html/", "/var/www/html"
```

Wenn Sie Parameter der VirtualBox-Konfiguration verändern möchten, müssen Sie dazu einen eigenen `config.vm.provider`-Block definieren. Das folgende Listing gibt dafür drei Beispiele:

```
Vagrant.configure("2") do |config|
  ...
  config.vm.provider "virtualbox" do |vb|
    # RAM in MiB für die virtuelle Maschine (Default: laut Box)
    vb.memory = 1024
    # CPU-Cores (Default: laut Box)
    vb.cpus = 2
    # beim Start VirtualBox-Fenster anzeigen (Default: false)
    vb.gui = true
  end
end
```

Scripts Mit `vagrant.vm.provision "shell"` ... erreichen Sie, dass Vagrant im Zuge des Provisionings das angegebene Script mit `root`-Rechten in der virtuellen Maschine ausführt. Kleinere Scripts können Sie direkt als Zeichenkette mit dem Schlüsselwort `inline` angeben:

```
config.vm.provision "shell", inline: "echo $(date)"
```

Auch mehrzeilige Scripts können Sie direkt in die Vagrant-Datei einbetten:

```
$myscript = <<END
apt-get update
apt-get install -y joe
END
```

```
Vagrant.configure("2") do |config|
  ...
  config.vm.provision "shell", inline: $myscript
end
```

Längere Scripts sind besser in eigenen Dateien untergebracht. Diese können Sie z. B. direkt im Vagrant-Projektverzeichnis speichern. Mit `path` geben Sie einfach den relativen Ort der Datei an. Anders als bei lokal auszuführenden Scripts ist es übrigens nicht erforderlich, die Datei mit `chmod a+x` ausführbar zu machen.

```
config.vm.provision "shell", path: "my-long-script.sh"
```

Wenn Sie in die Vagrant-Datei mehrere Scripts einbauen, ist es zweckmäßig, diese zu benennen. Dabei gilt die folgende Syntax:

```
config.vm.provision "script1", type: "shell", inline: "echo $(date)"
config.vm.provision "script2", type: "shell", inline: $myscript
```

Das hat zwei Vorteile: Zum einen können Sie damit die Ausgaben des Vagrant-Kommandos klarer einzelnen Scripts zuordnen, zum anderen ist es so möglich, nur ein bestimmtes Script auszuführen:

```
user$ vagrant provision --provision-with script2
```

Scripts für virtuelle Windows-Maschinen müssen übrigens in der Syntax von PowerShell formuliert werden – aber die PowerShell ist in diesem Buch ohnedies kein Thema.

Beispiel: CentOS-Webserver

Der Ausgangspunkt für das folgende Beispiel ist die Box `centos/7` aus dem Hashicorp-Katalog. Ähnlich wie mit `ubuntu/xenial64` erhalten Sie damit eine minimale Server-Installation: Die Box ist ca. 400 MiB groß und kompatibel mit vier Providern: VirtualBox, VMWare Workstation, VMWare Fusion und libvirt. Unter VirtualBox beansprucht `centos/7` anfänglich ca. 1 GiB Platz im Verzeichnis der VirtualBox-Maschinen. Die Eckdaten und einige Konfigurationsdetails sind hier dokumentiert:

<https://atlas.hashicorp.com/centos/boxes/7>

Um die Maschine im originalen Zustand einzurichten, führen Sie die folgenden Kommandos aus:

```
user$ mkdir centos7
user$ cd centos7
user$ vagrant init centos/7
user$ vagrant up
```

Im Gegensatz zu `ubuntu/xenial64` sind in der CentOS-Maschine die VirtualBox-Gasterweiterungen nicht standardmäßig installiert. Die Synchronisation des gemeinsamen Vagrant-Verzeichnisses erfolgt daher mit `rsync` beim Start der virtuellen Maschine. Beachten Sie, dass die Synchronisation einseitig ist: Es werden Dateien vom Host zum Gast übertragen, aber keine Änderungen vom Gast zurück zum Host synchronisiert.

Minimal ist auch die Vagrant-Datei von `centos/7`. Ohne Kommentare verbleiben nur drei Zeilen:

```
Vagrant.configure("2") do |config|
  config.vm.box = "centos/7"
end
```

Webserver installieren und starten Das Ziel dieses Beispiels ist es, in der virtuellen Maschine automatisiert einen Webserver einzurichten. Dazu erstellen Sie im Vagrant-Projektverzeichnis die folgende Script-Datei:

```
#!/bin/bash
# Datei /home/kofler/centos7/install-webserver.sh
yum install -y httpd
systemctl enable httpd
systemctl start httpd
```

Anschließend ergänzen Sie die Vagrant-Datei um die folgenden beiden Zeilen:

```
config.vm.provision "shell", path: "install-webserver.sh"
config.vm.network "forwarded_port", guest: 80, host: 8080
```

Die erste Zeile gibt an, dass das Script install-webserver.sh im Rahmen des Provisioning in der virtuellen Maschine mit root-Rechten ausgeführt werden soll. Die zweite Zeile leitet den Port 80 der virtuellen Maschine auf den lokalen Port 8080 um, sodass der neu installierte Webserver direkt auf dem Hostrechner ausprobiert werden kann.

Um die Installation durchzuführen, starten Sie die virtuelle Maschine mit der Option --provision neu. Dabei werden sämtliche Ausgaben des yum-Kommandos angezeigt. Im folgenden Listing habe ich die Ausgaben aus Platzgründen stark gekürzt:

```
user$ vagrant reload --provision
==> default: Running provisioner: shell...
      default: Running: script
...
==> default: Install 1 Package (+4 Dependent packages)
...
==> default: Created symlink
      from /etc/systemd/system/multi-user.target.wants/httpd.service
      to /usr/lib/systemd/system/httpd.service.
```

Um den Webserver auszuprobieren, öffnen Sie auf dem Hostrechner in einem Webbrowser die Seite `http://localhost:8080`. Sie sollten darin die Testseite des Webservers sehen.

Eigene HTML-Dateien Wenn Sie anstelle der Testseite eigene Webseiten anzeigen möchten, können Sie im Vagrant-Projektverzeichnis ein Unterverzeichnis mit den gewünschten HTML-Dateien einrichten:

```
user$ mkdir html
user$ cat > html/index.html << END
> <html>
> <body>
> <h1>Hello World!</h1>
> </body>
> </html>
> END
```

Damit alle Dateien aus dem lokalen html-Verzeichnis mit dem Verzeichnis `/var/www/html` in der virtuellen Maschine synchronisiert werden, ist die folgende Ergänzung in VagrantFile erforderlich:

```
config.vm.synced_folder "html/", "/var/www/html", type: "rsync"
```

Beachten Sie, dass dieses Beispiel nur statische Webseiten berücksichtigt. Wenn Sie eine dynamische Webseite einrichten möchten, müssen Sie in der virtuellen Maschine auch einen Datenbank-Server sowie geeignete Apache-Erweiterungen installieren, also z. B. MySQL und PHP. Dazu ist ein komplexeres Provisioning-Script erforderlich.

Sharing

Der Begriff »Sharing« meint in der Vagrant-Nomenklatur nicht den Datenaustausch zwischen Host und Gast über ein gemeinsames Verzeichnis (das ist ein »Shared Folder«), sondern bedeutet, dass Sie eine Vagrant-Maschine öffentlich im Internet zugänglich machen. Das setzt voraus, dass Sie zuerst einen kostenlosen Account auf `https://atlas.hashicorp.com` einrichten und dann mit `vagrant login` einmal einen entsprechenden Login durchführen:

```
root# vagrant login
In a moment we will ask for your username and password to HashiCorp's
Atlas. After authenticating, we will store an access token locally on
disk. Your login details will be transmitted over a secure connection, and
are never stored on disk locally.
```

```
Atlas Username: accountname
Password (will be hidden): *****
```

Sind diese Vorbereitungsarbeiten einmal erledigt, aktivieren Sie das Sharing mit dem Kommando `vagrant share`. Mit jeder Ausführung des Kommandos wird dem Share ein zufälliger Name zugewiesen – im folgenden Beispiel `arctic-gazelle-1751.vagrantshare.com`:

```
user$ vagrant share
==> default: Detecting network information for machine...
      default: Local machine address: 127.0.0.1
      default:
      default: Note: With the local address (127.0.0.1), Vagrant Share can only
      default: share any ports you have forwarded. Assign an IP or address to your
      default: machine to expose all TCP ports. Consult the documentation
      default: for your provider ('virtualbox') for more information.
      default:
      default: Local HTTP port: 8080
      default: Local HTTPS port: disabled
      default: Port: 2222
```

```

    default: Port: 8080
==> default: Checking authentication and authorization...
==> default: Creating Vagrant Share session...
    default: Share will be at: arctic-gazelle-1751
==> default: Your Vagrant Share is running! Name: arctic-gazelle-1751
==> default: URL: http://arctic-gazelle-1751.vagrantshare.com
==> default:
==> default: You're sharing your Vagrant machine in "restricted" mode. This
==> default: means that only the ports listed above will be accessible by
==> default: other users (either via the web URL or using `vagrant connect`).

```

Solange `vagrant share` läuft, kann nun jede Person mit Internetzugang über die oben abgedruckte Adresse auf den in Webserver zugreifen, der in einer virtuellen Maschine läuft. Diese Funktion bietet eine großartige Möglichkeit, eine in Entwicklung befindliche Web-Applikation unkompliziert auszuprobieren. Dazu müssen Sie nur `vagrant share` ausführen und dem Tester die zufällig erzeugte URL senden.

Im Gegensatz zu anderen `vagrant`-Kommandos läuft `vagrant share` unlimitiert. Sie müssen explizit `Strg+C` drücken, wenn Sie das Sharing beenden möchten.

SSH-Sharing Standardmäßig kümmert sich `vagrant share` nur um HTTP-Sharing. Jedes Mal, wenn Sie an das Kommando zusätzlich die Option `--ssh` übergeben, erzeugt Vagrant einen neuen SSH-Schlüssel. Der Zugriff auf diesen Schlüssel wird durch ein Passwort abgesichert, das Sie zweimal angeben müssen:

```

user$ vagrant share --ssh
...
==> default: Generating new SSH key...
    default: Please enter a password to encrypt the key: *******
    default: Repeat the password to confirm: *******
...
==> default: You're sharing with SSH access. This means that another user
==> default: simply has to run `vagrant connect --ssh arctic-gazelle-1751`
==> default: to SSH to your Vagrant machine.

```

Andere Vagrant-Benutzer können nun eine SSH-Verbindung zu Ihrem Server herstellen, indem sie das Kommando `vagrant connect --ssh ...` ausführen. Beim Login müssen sie das Passwort des SSH-Schlüssels angeben. Aus Sicherheitsgründen ist die Ausführung von `vagrant connect --ssh ...` nur Vagrant-Benutzern erlaubt, die ebenfalls einen Atlas-Account haben und sich mit `vagrant login` eingeloggt haben.

Auf einen Blick

Teil I	
Installation	23
Teil II	
Linux anwenden.....	137
Teil III	
Linux-Grundlagen.....	327
Teil IV	
Text- und Code-Editoren.....	499
Teil V	
Systemkonfiguration und Administration	561
Teil VI	
LAN-Server.....	909
Teil VII	
Root-Server.....	1055
Teil VIII	
Sicherheit.....	1223
Teil IX	
Virtualisierung & Co.	1297

Inhalt

Vorwort	19
TEIL I Installation	
1 Was ist Linux?	25
1.1 Einführung	25
1.2 Hardware-Unterstützung	26
1.3 Distributionen	28
1.4 Open-Source-Lizenzen (GPL & Co.)	32
1.5 Die Geschichte von Linux	35
1.6 Software-Patente und andere Ärgernisse	36
2 Installationsgrundlagen	39
2.1 Voraussetzungen	39
2.2 BIOS und EFI	41
2.3 Installationsvarianten	44
2.4 Überblick über den Installationsprozess	47
2.5 Start der Linux-Installation	49
2.6 Grundlagen der Festplattenpartitionierung	50
2.7 RAID, LVM und Verschlüsselung	57
2.8 Partitionierung der Festplatte	64
2.9 Installationsumfang festlegen (Paketauswahl)	70
2.10 Grundkonfiguration	72
2.11 Probleme beheben	75
2.12 Systemveränderungen, Erweiterungen, Updates	78
2.13 Linux wieder entfernen	81
3 Installationsanleitungen	83
3.1 CentOS	84
3.2 Debian	91

3.3	Fedora	99
3.4	Linux Mint	106
3.5	openSUSE	110
3.6	Ubuntu	120
3.7	Ubuntu Server	131

TEIL II Linux anwenden

4	Linux-Schnelleinstieg	139
4.1	Linux starten und beenden	139
4.2	Tastatur, Maus und Zwischenablage	141
4.3	Umgang mit Dateien, Zugriff auf externe Datenträger	144
4.4	Dokumentation zu Linux	145
5	Gnome	147
5.1	Erste Schritte	148
5.2	Dateimanager	153
5.3	Systemkonfiguration	162
5.4	Schriften (Fonts)	173
5.5	Gnome Tweak Tool	174
5.6	Gnome-Shell-Erweiterungen	176
5.7	Gnome Shell Themes	179
5.8	Gnome-Interna	181
5.9	Der Gnome-Klassikmodus	184
5.10	MATE	185
5.11	Cinnamon	186
6	KDE und Unity	189
6.1	KDE	190
6.2	KDE-Dateimanager	196
6.3	KDE-Konfiguration	199
6.4	Unity	204
7	Desktop-Apps	213
7.1	Firefox	214
7.2	Google Chrome	220
7.3	Thunderbird	222
7.4	Evolution, KMail und Geary	229

7.5	Dropbox	235
7.6	FileZilla und BitTorrent	237
7.7	Shotwell	238
7.8	digiKam	240
7.9	GIMP	242
7.10	RawTherapee, Darktable und Luminance (RAW- und HDR-Bilder)	246
7.11	Multimedia-Grundlagen	248
7.12	Rhythmbox, Amarok & Co	251
7.13	Spotify	254
7.14	VLC	255
7.15	Audio- und Video-Tools	256
7.16	Screenshots und Screencasts	264
8	Raspberry Pi	267
8.1	Grundlagen	268
8.2	Raspbian installieren und konfigurieren	272
8.3	Kodi und LibreELEC	283
8.4	Hardware-Basteleien	298
8.5	Interna und Backups	317
8.6	Wenn es Probleme gibt	324

TEIL III Linux-Grundlagen

9	Terminalfenster und Konsolen	329
9.1	Textkonsolen und Terminalfenster	330
9.2	Textdateien anzeigen und editieren	334
9.3	man und info	338
10	bash (Shell)	341
10.1	Was ist eine Shell?	341
10.2	Basiskonfiguration	343
10.3	Kommandoeingabe	344
10.4	Ein- und Ausgabeumleitung	349
10.5	Kommandos ausführen	352
10.6	Substitutionsmechanismen	354
10.7	Shell-Variablen	359
10.8	bash-Script-Beispiele	363
10.9	bash-Script-Grundregeln	370

10.10	Variablen in bash-Scripts	371
10.11	Codestrukturierung in bash-Scripts	378
10.12	Referenz wichtiger bash-Sonderzeichen	386
11	Dateien und Verzeichnisse	389
11.1	Umgang mit Dateien und Verzeichnissen	389
11.2	Links	400
11.3	Dateitypen (MIME)	402
11.4	Dateien suchen (find, grep, locate)	404
11.5	Zugriffsrechte, Benutzer und Gruppenzugehörigkeit	409
11.6	Spezialbits und die umask-Einstellung	415
11.7	Access Control Lists und Extended Attributes	420
11.8	Die Linux-Verzeichnisstruktur	425
11.9	Device-Dateien	429
12	Prozessverwaltung	433
12.1	Prozesse starten, verwalten und stoppen	433
12.2	Prozesse unter einer anderen Identität ausführen (su)	441
12.3	Prozesse unter einer anderen Identität ausführen (sudo)	443
12.4	Prozesse unter einer anderen Identität ausführen (PolicyKit)	447
12.5	Systemprozesse (Dämonen)	450
12.6	Prozesse automatisch starten (Cron)	454
12.7	Prozesse automatisch starten (systemd-Timer)	459
13	Konverter für Grafik, Text und Multimedia	463
13.1	Grafik-Konverter	463
13.2	Audio- und Video-Konverter	465
13.3	Textkonverter (Zeichensatz und Zeilentrennung)	468
13.4	Dateinamenkonverter (Zeichensatz)	469
13.5	Dokumentkonverter (PostScript, PDF, HTML, LaTeX)	469
13.6	Markdown und Pandoc	477
14	Netzwerk-Tools	481
14.1	Netzwerkstatus ermitteln	481
14.2	Auf anderen Rechnern arbeiten (SSH)	485
14.3	Dateien übertragen (FTP)	491
14.4	Lynx	496
14.5	Mutt	497

TEIL IV Text- und Code-Editoren

15	Vim	501
15.1	Schnelleinstieg	503
15.2	Cursorbewegung	505
15.3	Text bearbeiten	506
15.4	Suchen und Ersetzen	510
15.5	Mehrere Dateien gleichzeitig bearbeiten	511
15.6	Interna	513
15.7	Tipps und Tricks	516
16	Emacs	519
16.1	Schnelleinstieg	519
16.2	Grundlagen	523
16.3	Cursorbewegung	525
16.4	Text markieren, löschen und einfügen	527
16.5	Text bearbeiten	528
16.6	Fließtext	531
16.7	Suchen und Ersetzen	534
16.8	Puffer und Fenster	537
16.9	Besondere Bearbeitungsmodi	539
16.10	Konfiguration	541
16.11	MELPA	544
16.12	Unicode	545
17	Atom und VSCode	547
17.1	Atom	548
17.2	VSCode	555

TEIL V Systemkonfiguration und Administration

18	Basiskonfiguration	563
18.1	Einführung	563
18.2	Konfiguration der Textkonsolen	567
18.3	Datum und Uhrzeit	570
18.4	Datum und Uhrzeit via NTP synchronisieren	572
18.5	Benutzer und Gruppen, Passwörter	576

18.6	PAM, NSS und nscd	587
18.7	Spracheinstellung, Internationalisierung, Unicode	592
18.8	Hardware-Referenz	598
18.9	Logging (Syslog)	611
18.10	Logging (Journal)	619
19	Software- und Paketverwaltung	623
19.1	Einführung	623
19.2	RPM-Paketverwaltung	627
19.3	Yum	631
19.4	DNF	636
19.5	ZYpp	638
19.6	Debian-Paketverwaltung (dpkg)	640
19.7	APT	643
19.8	PackageKit	655
19.9	tar	656
19.10	Umwandlung zwischen Paketformaten (alien)	656
19.11	Verwaltung von Parallelinstallationen (alternatives)	657
19.12	Flatpak und Snap	659
19.13	Distributionsspezifische Eigenheiten	665
20	Bibliotheken und Java	677
20.1	Bibliotheken	677
20.2	Programme selbst kompilieren	682
20.3	Java	687
21	Grafiksystem	689
21.1	Grundlagen	690
21.2	Grafiktreiber	694
21.3	NVIDIA-Treiberinstallation	700
21.4	Status des Grafiksystems feststellen	702
21.5	Start des Grafiksystems	705
21.6	Konfiguration von X (xorg.conf)	710
21.7	Dynamische Konfigurationsänderungen mit RandR	716
22	Administration des Dateisystems	721
22.1	Wie alles zusammenhängt	723
22.2	USB-Datenträger formatieren und nutzen	725

22.3	Device-Namen für Festplatten und andere Datenträger	728
22.4	Partitionierung der Festplatte oder SSD	733
22.5	parted-Kommando	737
22.6	Partitionierungswerkzeuge mit grafischer Benutzeroberfläche	742
22.7	Dateisystemtypen	744
22.8	Verwaltung des Dateisystems (mount und /etc/fstab)	749
22.9	Dateisystemgrundlagen	755
22.10	Das ext-Dateisystem (ext2, ext3, ext4)	758
22.11	Das btrfs-Dateisystem	764
22.12	Das xfs-Dateisystem	778
22.13	Windows-Dateisysteme (vfat, ntfs)	780
22.14	CDs und DVDs	784
22.15	Externe Datenträger	786
22.16	Swap-Partitionen und -Dateien	787
22.17	RAID	790
22.18	Logical Volume Manager (LVM)	798
22.19	SMART	803
22.20	SSD-TRIM	807
22.21	Verschlüsselung	808
23	GRUB	817
23.1	GRUB-Grundlagen	817
23.2	GRUB-Bedienung (Anwendersicht)	826
23.3	GRUB-Konfiguration	827
23.4	Manuelle GRUB-Installation und Erste Hilfe	841
24	Das Init-System	847
24.1	systemd	848
24.2	Das Init-V-System	857
24.3	Eigene Init-Scripts bzw. Init-Konfigurationsdateien	861
24.4	Systemstart bei CentOS, Fedora und RHEL	865
24.5	Systemstart bei Debian, Raspbian und Ubuntu	867
24.6	Systemstart bei SUSE/openSUSE	869
24.7	Internet Service Daemon	870
25	Kernel und Module	875
25.1	Kernelmodule	876
25.2	Device Trees	882
25.3	Kernelmodule selbst kompilieren	885

25.4	Kernel selbst konfigurieren und kompilieren	889
25.5	Kernel-Neustart mit kexec	899
25.6	Kernel-Live-Patches	900
25.7	Die Verzeichnisse /proc und /sys	902
25.8	Kernel-Boot-Optionen	904
25.9	Kernelparameter verändern	908

TEIL VI LAN-Server

26	Netzwerkkonfiguration	911
26.1	Der NetworkManager	911
26.2	Proxy-Konfiguration	919
26.3	Netzwerkgrundlagen und Glossar	920
26.4	Manuelle LAN- und WLAN-Konfiguration	933
26.5	LAN-Konfigurationsdateien	942
26.6	Distributionsspezifische Konfigurationsdateien	948
26.7	Zeroconf und Avahi	958
27	Internet-Gateway	961
27.1	Einführung	961
27.2	Netzwerkkonfiguration	967
27.3	Masquerading (NAT)	970
27.4	Der WLAN-Authenticator hostapd	973
27.5	DHCP- und Nameserver-Grundlagen	976
27.6	Dnsmasq (DHCP- und Nameserver)	978
28	Samba	987
28.1	Grundlagen und Glossar	988
28.2	Basiskonfiguration und Inbetriebnahme	992
28.3	Passwortverwaltung	999
28.4	Netzwerkverzeichnisse	1006
28.5	Beispiel – Home- und Medien-Server	1013
28.6	Beispiel – Firmen-Server	1016
28.7	Client-Zugriff	1019
29	NFS und AFP	1025
29.1	NFS	1025
29.2	Apple Filing Protocol	1032

30	CUPS	1037
30.1	Grundlagen	1037
30.2	CUPS-Intern	1040
30.3	Druckerkonfiguration	1046
30.4	Drucken in lokalen Netzwerken	1049
30.5	AirPrint	1052

TEIL VII Root-Server

31	Secure Shell (SSH)	1057
31.1	Installation	1058
31.2	Konfiguration und Absicherung	1058
31.3	DenyHosts und Fail2Ban	1061
31.4	Authentifizierung mit Schlüsseln	1064
31.5	Zusatzwerkzeuge	1067
32	Apache	1073
32.1	Apache	1073
32.2	Webverzeichnisse einrichten und absichern	1081
32.3	Virtuelle Hosts	1089
32.4	Verschlüsselte Verbindungen (HTTPS)	1095
32.5	Let's Encrypt	1105
32.6	Webzugriffsstatistiken	1112
32.7	PHP	1116
32.8	FTP-Server (vsftpd)	1118
33	MySQL und MariaDB	1123
33.1	Installation und Inbetriebnahme	1124
33.2	Administrationswerkzeuge	1133
33.3	Backups	1138
34	Postfix und Dovecot	1143
34.1	Einführung und Grundlagen	1143
34.2	Postfix (MTA)	1154
34.3	Postfix-Verschlüsselung (TLS/STARTTLS)	1162
34.4	Postfix-Konten	1169
34.5	Dovecot (POP- und IMAP-Server)	1179

34.6	Client-Konfiguration	1186
34.7	Spam-Abwehr	1187
34.8	ClamAV (Virenabwehr)	1193
34.9	SPF, DKIM und DMARC	1195
34.10	Konfigurationstest und Fehlersuche	1205
35	Nextcloud	1207
35.1	Installation	1208
35.2	Wartung	1215
35.3	Betrieb	1217
35.4	Kontakte und Termine	1219

TEIL VIII Sicherheit

36	Backups	1225
36.1	Backup-Benutzeroberflächen	1225
36.2	Backups auf NAS-Geräten	1231
36.3	Dateien komprimieren und archivieren	1232
36.4	Verzeichnisse synchronisieren (rsync)	1235
36.5	Inkrementelle Backups (rdiff-backup)	1238
36.6	Inkrementelle Backups (rsnapshot)	1240
36.7	Backup-Scripts	1243
36.8	Backups auf S3-Speicher	1246
37	Firewalls	1251
37.1	Netzwerkgrundlagen und -analyse	1251
37.2	Basisabsicherung von Netzwerkdiensten	1257
37.3	Firewall-Grundlagen	1261
37.4	Firewall-Konfigurationshilfen	1267
37.5	Firewall mit iptables selbst gebaut	1274
38	SELinux und AppArmor	1283
38.1	SELinux	1283
38.2	AppArmor	1291

TEIL IX Virtualisierung & Co.

39	VirtualBox und Vagrant	1299
39.1	VirtualBox installieren	1300
39.2	VirtualBox-Maschinen einrichten	1304
39.3	Arbeitstechniken und Konfigurationstipps	1309
39.4	Vagrant	1316
40	KVM	1329
40.1	Grundlagen	1330
40.2	KVM ohne libvirt	1337
40.3	Der Virtual Machine Manager	1339
40.4	libvirt-Kommandos	1348
40.5	Integration der virtuellen Maschinen in das LAN (Netzwerkbrücke)	1354
40.6	Direkter Zugriff auf den Inhalt einer Image-Datei	1357
41	Docker	1363
41.1	Grundlagen, Nomenklatur und Installation	1364
41.2	Docker kennenlernen	1367
41.3	Docker administrieren	1380
41.4	Docker Images erzeugen und weitergeben	1390
41.5	Interna	1398
42	Linux on Windows	1405
42.1	WSL ausprobieren	1406
42.2	Serverbetrieb	1411
42.3	Interna	1417
Index	1419

Index

1-Wire-Thermometer	313
389-Directory-Server	564
4-KiB-Sektoren	55
4k-Bildschirme	168
64-Bit-Bibliotheken	680
64-Bit-Distributionen	40
7zr	1233
802.11x-Standards	929
\$ (Variablen in der bash)	359
\$() (Kommandosubstitution)	357
& (Hintergrundprozesse)	434
< (Ausgabeumleitung)	349
> (Eingabeumleitung)	350
[] (arithmetische Ausdrücke)	357
* (Jokerzeichen)	355, 396
? (Jokerzeichen)	355, 396
# (Kommandointerpreter)	364
~ (Heimatverzeichnis)	144, 390
" " (Zeichenketten)	358
' ' (Zeichenketten)	358
` (Kommandosubstitution)	357
A	
A-Eintrag (DNS)	1150
a2disconf	1078
a2dismod	1078
a2enconf	1078
a2enmod	1078
a2ensite	1078
a2ps	470
aa-complain	1293
aa-enforce	1293
aa-status	1292
AAAA-Eintrag (DNS)	1150
Abkürzungen	348
Access Control Lists	420
Access Point (WLAN)	915, 929, 966
ACL	420
ACPI	601
<i>Kernel-Boot-Optionen</i>	908
Active Directories	991
Ad-hoc-Modus (WLAN)	930
addgroup	577
adduser	577
Administration	563
Administrator-Account	72
Adobe Flash	218
AFP	1032
afp.conf	1033
AirPrint	1052
airprint-generate	1053
akmods	1301
akms	888
Aktion (Syslog)	614
Aktivitäten	
<i>Gnome</i>	149
<i>KDE</i>	195
Alias (httpd.conf)	1083
Alias (E-Mail)	1171
alias	348
alias (in modprobe.conf)	881
alias_database	1171
alias_maps	1172
alien	656
Allow (Apache)	1085
allow-hotplug (/etc/network/interfaces)	952
Allowed-Origins	651
AllowOverride	1084
alsactl	610
alsamixer	610
alternatives	658
Amarok	252
Amazon Web Services (AWS)	1246
amdgpu	696
anacron	458
Android	26, 29
Antergos	29
Apache	1073
<i>Authentifizierung</i>	1087
<i>HTTPS</i>	1095
<i>IPv6</i>	1079
<i>Passwort</i>	1087
<i>SELinux</i>	1075
<i>Sicherheit</i>	1087
<i>Unicode</i>	1079
<i>Verzeichnis absichern</i>	1087
<i>virtuelle Hosts</i>	1089
<i>WSL</i>	1415
<i>Zugriff sperren</i>	1087
apfs-Dateisystem (Apple)	746
APIC	907
aplay	610

apm (Atom-Paketverwaltung) 552
AppArmor 1283, 1291
apparmor-utils 1293
Apple
 AirPrint 1052
 Dateisystem 746
 Filing Protocol (AFP) 1032
 Samba 1024
Applets
 Gnome 149
 KDE 192
applydeltarpm 628
Apps 221
APT 643
 automatische Updates 650
apt 644
apt-cache 649
apt-get 644
apt-key 645
aptitude 644, 647
apturl 650
arandr 719
Arbeitsfläche
 Gnome 152
 KDE 195
Arch Linux 29
Archivieren von Dateien 1232
 Gnome 161
arecord 610
Arithmetische Ausdrücke (bash) 357
Artifex Ghostscript 473
ASCII 593
Asymmetrische Verschlüsselung 1095
async (NFS) 1027
Atlas 1316
Atom 548
Audacious 253
Audio
 ALSA 609
 Dateien recodieren 259
 Konverter 465
audiofile 466, 467
aufs-Dateisystem 748, 1400
Ausgabeumleitung 349
 sudo 444
Auslagerungsdatei 68
authconfig (PAM) 590
AuthConfig (Apache) 1084
Authentifizierung
 Apache 1087
 POP/IMAP 1184
 SMTP 1185
AuthName 1088

AuthType 1088
AuthUserFile 1088
auto-Dateisystem 747
Auto-Login 708
 KDE 201
 SUSE 709
autofs 747
autojump 392
automatic.conf 637
Automatische Ausführung von Jobs 454, 459
automount 747
Autostart
 Gnome 181
 KDE 201
 Unity/Ubuntu 211
Avahi 958
avahi-browse 960
avahi-daemon 959
avahi-discover 960
avahi-dnscfnd 959
avconv 248, 467
aws 1247
awscli 1247
AWStats 1112

B

Babe 253
Background-Prozesse 434
backintime 1229
Backports (Debian) 667
Backup Domain Controller 991
Backups 1225
 Emacs 521
 inkrementelle 1238
 KVM 1245
 LVM-Snapshots 1244
 MySQL 1138
 Scripts 367
bad-interpreter-Fehlermeldung 370
baobab 161
Base Boxes (Vagrant) 1318
Base Images (Docker) 1368
bash 341
 bash_history 344
 bashrc 105
 completion 346
 Programmierung 363
 Script-Beispiele 363, 461
 Tastatureinstellung 343
 Tastenkürzel 346

unter Windows (WSL) 1405
 Variablen 373
Batterie (Notebooks) 601
BCM2835/-36/-37 299
BDC 991
Bedingungen (bash) 379
Benutzer 578
 einrichten 576
 Gruppen 410
 verwalten 576
Besitzer
 neue Dateien 418
 von Dateien 409
Betriebssystem 25
bg 435
Bibliotheken 677
 32/64 Bit 680
 glibc 678
 libc 678
 Prelinking 681
Bilder-Verzeichnis 183
Bildschirmfreigabe 171
Bildschirmschoner (Raspberry Pi) 281
bin-Verzeichnis 371, 426
Binärpaket 628
bind 978
bind interfaces only 997
bind-address (MySQL) 1126
BIOS 41
 GRUB-Partition 841
 GRUB-Reparatur 841
 RAID 57
 Systemstart 822
BitTorrent 237
blacklist (modprobe.conf) 882
Blacklist (E-Mail) 1150
blkid 752, 762
bluedevil 606
Bluetooth 606
 Raspberry Pi 283
bluetoothd 606
BMP-PS-Konverter 464
bmp2eps 464
bmp2tiff 465
Bonjour 958
Bookmarks (Firefox Sync) 215
/boot
 /efi 43, 818
 /grub 827
 /initrd 820
 /initrd selbst erzeugen 823
 /vmlinuz 819, 898
Boot-Optionen 904

Boot-Partition 66, 67
 voll wegen Kernel-Updates 652
Boot-Probleme 76
Boot-Prozess 817, 847
 Bootloader 817, 823
 boot.local 870
 System-V-Init 857
BOOTREC 82
Boxes (Vagrant) 1317
Bridge 1354
bridge-utils 1354
Bridged Networking 1310
browseable (Samba) 1007
Browsing (Samba) 988
BSD-Lizenz 33
btrfs-Dateisystem 745, 764
 RAID 771
 Snapper/opensUSE 775
 Swap-Dateien 790
Buckets (S3) 1246
Budgie 31, 122
build-Kommando (Docker) 1393
bunzip2 1232
Buster 667
bzip2 1232, 1233

C

C/C+ (Programmiersprache) 686
.cache-Verzeichnis 183
CalDAV 1220
Canonical (Ubuntu) 120
canonical-livepatch 901
canonical_maps 1174
Capabilities 424
CapsLock-Taste deaktivieren 162
CardDAV 1220
Carriage Return 468
case 380
cat 334
CD
 auswerfen 785
 Devices 784
 ins Dateisystem einbinden 784
 Ripper 261
 umount-Problem 785
 wechseln 785
cdda2wav 465
cdparanoia 465
CentOS 30, 84
 statische Netzwerkkonfiguration 950
 sudo 446

Systemstart 865
Tastatur 568
certbot (Let's Encrypt) 1106
Certification Authority (CA) 1103
cgroups 747
 Docker 1402
chage 583
chainloader 837
Channel (WLAN) 931
character set 593
character-set-server (MySQL/MariaDB) . 1126
chattr 767
chcon 1286
checkarray 792, 798
checkrestart 652
chkconfig 860, 867, 873
Chrome 220
Chrome OS 30
chrome-gnome-shell 176
Chromium 220
Chrony 573
chroot 842, 843, 1260
chsh 342
cifs-Dateisystem 1020, 1022
cifs-utils 1020
Cinnamon Desktop 186
ClamAV 1193
classes.conf 1040
Client-Konfiguration 911
cloneconfig 895
Cluster SSH 1067
Cluster-Dateisystem 723, 748
CMD (Dockerfile) 1391
cmus 254
Codec 248
collation-server (MySQL/MariaDB) 1126
Compiler 682
Compiz (Unity) 205
complete 346
.config-Verzeichnis 183
config.txt (Device Trees) 883
configure 684
conky 437
console-setup 567, 569
Container (Docker) 1363
Container Layer (Docker) 1399
Contrib-Pakete 666
Control Groups 747
 Docker 1402
Converseen 464
convert 463
convmv 469
Copr (DNF) 637

Copy on Write (COW) 764
 deaktivieren 766
cp 391, 392
 Namen beim Kopieren ändern 398
cPanel 564
cpio 825
CPU-Frequenz limitieren 600
CPU-Temperatur 600
cpu-checker 1330
cpufreq 600
cpufreq-set 600
cpufrequtils 600
cracklib 584
cramfs-Dateisystem 748
create mask 1008
cron 454
 durch systemd ersetzen 459
crontab 454
crossmnt (NFS) 1028
cryptsetup 809
Crypto-Dateisystem 722
csh 342
CSS 249
CSSH 1067
ctrl-alt-del.target 852
CUPS 1037
 Browsing-Funktion 1051
 cupsd 1040
 cupsd.conf 1040
 cupsenable 1041
 Firewall 1049
 Interna 1040
 Netzwerkdrucker nutzen 1049
 SUSE-Besonderheiten 1043
curl 495, 1245
CustomLog 1083
Cut&Paste 143

D

Dämonen 450
dash (Shell) 364
Dash (Gnome) 150
 Dash to Dock 178
Dash (Unity) 205
data-Option (Journaling-Modus) 758
Dateien
 Dateinamen 389
 drucken 1045
 Grundlagen 389
 Jokerzeichen 355
 komprimieren 1232

kopieren mit sed 398
Nautilus 153
suchen 404
Dateisystem
 ext-Dateisystem 758
 Fragmentierung 763
 Konfiguration 752
 Loopback-Device 748
 maximale Dateigröße 757
 reparieren 756
 Schnelleinstieg 144
 Typen 744, 754
 überprüfen 756
 vergrößern (ext) 762
 vergrößern (xfs) 780
 verschlüsseln 722
 verwalten 721
 virtuelles 747
 WSL 1409
Dateityp
 im ls-Kommando 393
 Magic-Datei 404
 MIME 402
Datenbank-Server 1123
Datenpartition 66
Datenträger formatieren 725
dbus-daemon 609
dcfldd 277, 323
dconf 181
dconf-editor 181
dcraw 247, 465
dctrl-Format 641
dd 44
Dead Keys 49, 567
Debian 30
 debian-goodies 652
 DKMS 887
 Firmware 94
 initrd-Datei 824
 Paketverwaltung 640
 Systemstart 867
 Tastatur 567
 VirtualBox 1306
declare 361
decode_MPG2 und _WVC1 289
Decoder 248
Decodierer (MPEG-2, VC-1) 288
Defragmentierung 763
Deja Dup 1225
Delayed Allocation (ext4-Dateisystem) 759
delgroup 577
Delta-Updates 628
deltarpm 628

deluser 577
Deny (Apache) 1085
DenyHosts 1062
Desklets 187
.desktop-Dateien 708
Desktop
 Gnome 147
 KDE 190
deutsche Sonderzeichen
 bash 343
 Emacs, US-Tastatur 546
/dev 426
 /disk 732
 Interna 429
 Liste 431
 /mapper 800
 /md 791
 /pts 747
 /sd 729
 /vd 730
Device-Abschnitt (X) 713
device is busy (Fehlermeldung) 785
Device Trees 314, 882
device-tree-Parameter 883
DeviceKit 608
Devices 412, 429, 728
 CD/DVD-Laufwerke 784
 Interna 429
 Kernelmodule 881
 udev-Dateisystem 430
Devil Linux 32
devtmpfs-Dateisystem 747
df 750
dhclient 937, 956
dhclient.conf 956
DHCP 923, 976
 Client-Konfiguration 923, 985
 Hostname 985
 Server 976
 Server-Konfiguration (Dnsmasq) 978
dhcpcd 937, 949
dhcpcd 978
digiKam 240
Dillo 222
directory mask 1008
Directory Server 564
DirectoryIndex 1084
disable_vrfy_command (Postfix) 1179
discard 754, 807
Discover 655
Disk-Images (libvirt/KVM) 1336
Disk-Quotas 723
Display-Abschnitt (X) 715

Display Manager	707	Downloads-Verzeichnis	183
Distributionen		dpkg	641
Überblick	29	dpkg-reconfigure	572, 642
Updates	78	Beispiele	641
DKIM (Postfix)	1197	Multiarch	680
DKMS (Kernel)	887	Statuscode	642
VirtualBox	1300	DPMS (Raspberry Pi)	281
DLLs	678	dracut	825
DLNA	296	Dragon	256
DMARC	1203	Dreischritt (Kompilieren)	684
dm_crypt	809	DRI	693
dmesg	615	DriveFS-Dateisystem	1418
DNF (Paketmanager)	636	DRM	249, 691
dnf.conf	636	Dropbox	235
dnf-plugin-system-upgrade	668	Drucken	1037
downgrade	1307	automatische Datenkonversion	1038
systemd-Timer-Beispiel	460	Dämon (CUPS)	1040
DNS		Druckjobs verwalten	1045
Client-Konfiguration	923, 944	Filter	1038
Mail-Server	1150	GDI-Drucker (Windows)	1047
Proxy	994	Gnome	164
Reverse DNS	1153	KDE	203
Server-Konfiguration (Dnsmasq)	978	Konfiguration	1037, 1046
Ubuntu	917	MIME (CUPS)	1041
dns-nameservers	953	per Kommando	1045
Dnsmasq	978	PostScript	1037
NetworkManager	917	Server-Konfiguration	1037, 1050
do-release-update	676	Spooling-System	1038
Dock (Unity)	205	Warteschlange	1038
Docker	1363	DS1820	313
docker-compose	1388	DSC (PostScript)	475
Dockerfile-Syntax	1390	DTB-Dateien	883
DocumentRoot (Apache)	1075, 1082	dtoverlay-Schlüsselwort	883
Dokumente-Verzeichnis	183	dtparam-Schlüsselwort	883
Dokumentkonverter	469	Dual-Stack (IPv6)	929
Dolphin	196	Duplicity	1240
Verzeichnis freigeben	1012	DVD	
Domain-Level-Sicherheit	991	auswerfen	785
Domain-Nameserver siehe DNS	977	brennen in Gnome	157
DomainKeys Identified Mail	1197	brennen in KDE	199
Domainname	921	Dateisystem	746
Doppellizenzen	33	Devices	784
DOS-Dateien konvertieren	468	ins Dateisystem einbinden	784
DOS-Dateisystem	746	Ripper	262
dos2unix	468	umount-Problem	785
dotglob	356	Videos abspielen	786
Dovecot		wechseln	785
dovecot.conf	1180	Dynamic Host Configuration Protocol	976
IPv6	1182	Dynamisch gelinkte Programme	678
POP/IMAP-Authentifizierung	1184		
SMTP-Authentifizierung	1185		

E		Erweiterungen	544
E-Mail		farbiger Text	539
Alias	1171	Fenster	538
Blacklist	1150	Fließtext	531
DNS	1150	font-lock-mode	539
Evolution	229	fremdsprachige Zeichen	545
Grundlagen	1144	Hintergrundfarbe einstellen	542
Kontakt	232	Konfiguration	541
mutt	497	MELPA	544
Relaying	1150	Online-Hilfe	522
Server	1143	Puffer	537
Thunderbird	222	reguläre Ausdrücke	535
Viren	1193	Schnelleinstieg	335
e2label	762	Schriftart einstellen	542
e4defrag	763	suchen	534
EasyTAG	260	suchen und ersetzen	536
eBox	564	Syntaxhervorhebung	539
EDITOR	337	Tabulatoren	529
Editoren	335	Textmodus	533
Atom	547	Unicode	545
Emacs	519	emergency (Kerneloption)	906
Joe	337	Emergency-Target	852
Nano	337	Encoder	248
Vim	501	Encryption (Dateisystem)	722
VSCode	555	Energiesparfunktionen	601
EFI	41	Enigmail	228
Bootloader per GRUB starten	839	enscript	470
efibootmgr	844	ENTRYPOINT (Dockerfile)	1391
GPT	818	env	597
GRUB 2	818	Environment-Variablen	360
GRUB-Reparatur	843	EnvironmentFile (systemd)	863
Partition	43, 818	EPEL-Paketquelle	90, 665
Secure Boot	43, 77, 820	Epiphany	222
Systemstart	818	EPS-Konverter	472
EGL	691, 693	epsffit	474
Eingabefokus (X)	143	epstopdf	472
Eingabeumleitung	349	ErrorDocument	1083
sudo	444	ErrorLog	1082
eject	785	erweiterte Partition	53
Electronic Frontier Foundation	1106	ESP Ghostscript	473
Elementary OS	31, 122	ESR-Version	
ELinks	471, 496	Firefox	214
Elvis	335	Thunderbird	223
Emacs	519	ESSID (WLAN)	930
automatische Sicherheitskopie	521	/etc	426, 566
Bearbeitungsmodi	523, 539	/adduser.conf	577
Cursorbewegung	525	/adjtime	570
dynamische Abkürzungen	534	/aliases	1147, 1157, 1171
Editierkommandos	528	/alternatives	657
Ein- und Ausrückungen	530	/apparmor.d	1292
Einrückungen im Fließtext	532	/apt/apt.conf	644
.emacs-Datei	541, 542	/apt/sources.list	644
		/auto.master	747

/boot/grub.cfg	827	/mdadm/mdadm.conf	791
/chrony.conf	573	/mime.types	404
/cron.daily	457	/modprobe.conf	880, 935
/cron.hourly	457	/modprobe.d	880
/cron.monthly	457	/modules	881
/cron.weekly	457	/modules-load.d	880
/crontab	454	/mtab	750
/crypttab	811	/my.cnf	1126
/cups	1040	/netatalk/afp.conf	1033
/default/language	596	/network/interfaces	952
/default/grub	830	/nscd.conf	592
/default/prelink	681	/nsswitch.conf	591
/deluser.conf	577	/PackageKit/*	655
/denyhosts.conf	1062	/pam.conf	588
/dhcp3/dhclient.conf	956	/pam.d/*	588
/dnf/dnf.conf	636	/passwd	579
/dnsmasq.conf	979	/php.ini	1117
/dovecot	1180	/polkit-1	448
/dracut.conf	825	/postfix	1157
/file	404	/prelink.conf	681
/firewall.d	1268	/printcap (CUPS)	1040
/fstab	752	/profile	360, 362
/fstab (CIFS)	1022	/rc.d/*	859
/fstab (LABEL)	752	/rc.d/rc.local	867
/fstab (NFS)	1030	/rc.local	868
/fstab (SSD-TRIM)	807	/resolv.conf	944
/ftusers	1121	/resolv.conf (Ubuntu)	945
/group	580	/rsyslog.conf	612
/gshadow	586	/samba/smb.conf	993
/host.conf	943	/selinux	1289
/hostapd.conf	974	/services	871
/hostname	946	/shadow	582
/hosts	943, 979	/shells	342
/hosts (neuen Server testen)	1094	/skel	580
/hosts.allow	1258	/smartd.conf	806
/hosts.deny	1258	/ssh	1058
/idmapd.conf	1029	/ssl/certs	914
/inetd.conf	872	/sudoers	443
/init.d	869	/sysconfig	869
/init.d/boot.local	870	Dokumentation	952
/inittab	858	network-scripts	950
/inputrc	343	/sysconfig/authconfig	590
/ld.so.cache	679	/sysconfig/i18n	596
/ld.so.conf	679	/sysconfig/language	596
/letsencrypt	1107	/sysconfig/console	569
/libvirt	1334	/sysconfig/locate	406
/lightdm	708	/sysconfig/network	946
/locale.conf	596	/sysconfig/prelink	681
/localtime	571	/sysctl.conf	908, 972
/login.defs	420, 583	/systemd/network	956
/logrotate.conf	616	/systemd/journal.conf	621
/mailcap	404	/timezone	571
/manpath.conf	339	/udev	430

/updatedb.conf	406	Fastest Mirror (Yum)	634
/vconsole.conf	568	FAT-Dateisystem	780
/vsftpd.conf	1120	fbdev-Treiber (X)	714
/vsftpd/ftpusers	1121	fc-list	174
/vsftpd/user_list	1121	Fedora	30, 99
/wpa_supplicant.conf	282	automount	747
/wpa_supplicant	941	Distributions-Update	668
/X11/xorg.conf	710	DKMS	887
/xdg/user-dirs.conf	183	dracut	825
/xinetd.d/*	873	Firewall	1267
/yum.conf	632	Gateway-Konfigurationsdatei	944
Ethernet-Controller		initrd-Datei	825
IP-Adresse	924	LABEL in /etc/fstab	752
konfigurieren	934	Masquerading	972
MAC-Adresse	922	statische Netzwerkkonfiguration	950
ethtool	947	sudo	446
evim	517	Systemstart	865
Evolution	229	Tastatur	568
except-interfaces	980	VirtualBox	1307
Exchange-Server	1145	Fensterbuttons (Gnome)	174
*.exe-Datei	434	Fernwartung	171
Exec Shield	1284	feste Links	400
ExecCGI	1084	Festplatte	
ExecStart-Schlüsselwort (systemd)	862	4-KiB-Sektoren	55
exFAT-Dateisystem	780	formatieren	52
exfat-utils	781	partitionieren, Linux	64
EXIF-Informationen	465	überwachen (SMART)	803
exit (bash)	385	fetchmsttfonts	173
Expansion von Dateinamen	344	FFmpeg	467, 248
expect	584, 1059	fg	435
Experimental-Pakete	667	fglrx	696
export	361	FHS	425
ext-Dateisystem	745, 758	FIFO	350
Verschlüsselung	812	file	404
Windows-Zugriff	763	FileInfo	1084
Extended Attributes	421	Filesystem Hierarchy Standard	425
Extension Pack (VirtualBox)	1303	FileZilla	237
externe Laufwerke	786	Filter	
extractres	474	CUPS	1038
		IP-Paketfilter	1262
F		find	407
f.lux-Programm	168	findmnt	750
faac	466	Firefox	214
faad	466	MIME	216
FAI (Fully Automatic Installation)	564	Plugins	217
Fail2Ban	1063	Sync	215
faillock	585	Firewall	1251
faillog	585	AFP-Server (Netatalk)	1034
Fake-RAID	57	Beispiel	1274
Fan Control	604	CUPS	1049
Farbprofile	168	FTP	493
		Grundzustand herstellen	1277
		IPv6	1265, 1274

Mail-Server 1156
NFS 4 1029
openSUSE 119
Paketfilter 1262
Samba 996
Web-Server 1074
firewall-cmd 1156, 1270
 NFS 1029
FirewallD 1267
Firewire-Laufwerke 786
Firmware 933
 Debian 94, 99
fish (Konqueror) 199
fixfmps 474
fixmacps 474
fixscribeps 474
fixtpps 474
fixwfwps 474
fixwpps 474
fixwwps 474
flac 467
Flash 218
flashplugin-installer 219
Flatpak 661
Fokus (X) 143
FollowSymLinks 1084
font-lock-mode 539
Fonts 173, 595
for (bash) 381
force group 1008
forcefsck 757
Fork-Typ (systemd) 862
FORMAT (Windows) 52
Formatieren
 btrfs-Dateisystem 765
 extfat-Dateisystem 725, 781
 ext3/ext4-Dateisystem 760
 ntfs-Dateisystem 725, 781
 vfat-Dateisystem 725, 781
 xfs-Dateisystem 779
Forward Secrecy 1167
Fotodrucker 1047
Fragmentierung 763
Framebuffer (X) 714
free 601
Free Software Foundation 33
Freigaben
 Gnome-Einstellungen 1012
 Medien (Gnome) 159
 Nautilus 158, 1011
 WebDAV (Gnome) 159
fremdsprachige Zeichen (Emacs) 545
freshclam 1193

fsck 757
fsck.xfs 780
FSF 33
fsid (NFS) 1028
FSSTND 425
fstab 752
 CIFS 1022
 NFS 1030
fstrim 808
ftp-Kommando 491
FTP 491
 Client 237, 491
 Masquerading 973
 passiver Modus 493
 Secure FTP Server 1059
 Server 1118
 Server (sftp) 1059
ftputers (vsftpd) 1121
function (bash) 384
FUSE 748, 783
fuser 437

G

Gateway 923
 Client-Konfiguration 936, 944
 Server-Konfiguration 970
gconf 181
gconf-editor 181
gconftool-2 181
GDI-Drucker 1047
gdm 707
Geary 234
getafm 474
getcap 425
getfacl 422
getfattr 424
getsebool 1288
gfs-Dateisystem 748
gftp 492
Ghostscript 472
GID 580
gif2tiff 465
GIMP 242
 Screenshots 264
gimp-dcraw 247
GIMP-Print 473
GL (Open GL) 694
glibc 678
 Zeitzone 571
Global Filesystem 723
Global Unicast (IPv6) 926

globstar-Option 356
GLX 693
glx-utils 704, 1307
glxinfo 704
GMT (Greenwich Mean Time) 570
Gnome 147
 Bildschirmeinstellungen 718
 Extensions 176
 gdm 708
 Proxy-Einstellungen 919
 Screenshots 264
 Shell Extensions 176
 Shell Themes 179
 Tweak Tool 174
 Verzeichnis freigeben 1011
gnome-bluetooth 606
gnome-disk-utils 743
gnome-disks 743
gnome-keyring-daemon 1065
gnome-language-selector 595, 676
gnome-nettool 484
gnome-screenshot 264
gnome-software 655
gnome-software-Programm 665
gnome-sound-recorder 257
gnome-system-monitor 437
gnome-terminal 331
gnome-tweak-tool 162, 175
gnome-vfs.keys 182
gnome-vfs.mime 182
GNU 35
 Emacs 519
 General Public License 33
 Ghostscript 473
 GRUB 817
GoAccess 1113
Google Analytics 1112
Google Authenticator 1069
Google Chrome 220
GOsa 564
GParted 742
gpasswd 586
gpg 809, 1249
gpio 305
GPL 33
gpm 569
GPT 52, 734
 EFI 818
 GRUB 2 841
 Partitionsnummern 731
Grafik-Konverter 463
graphical-Target 705

grep 408
 Beispiele 364
grep-dctrl 641
Greylisting (Postfix) 1191
groupadd 577
Grsync 1228
GRUB 817
 Bedienung 826
 GPT 841
 Festplattenamen 834
 Kernel-Updates 825
 Konfiguration 827
 Partitionsnamen 834
 Secure Boot 820
 Version 2 827
 Reparatur (BIOS) 841
 Reparatur (EFI) 843
grub-editenv 834
grub-install 841
grub.cfg 827
grub2-mkconfig 828, 829
grubby 817
Gruppen 580
 neue Dateien 418
 von Dateien 409
gs 472
gsettings 181
gsox 467
GStreamer 250, 610
gtf 712, 718
GTUBE-Testnachricht 1190
gucharmap 174
guest account 1009
guest ok 1009
guest only 1009
Gufw 1274
gunzip 1232, 1233
Gutenprint 473, 1039
GVFS 158, 1232
gvim 502
gzip 1232, 1233

H

Hacker-Kernel 890
HAL 608
HandBrake 263
Hardware
 Devices 429
 Devices (udev-Dateisystem) 430
 RAID 57
 Referenz 599

Hardware Enablement Stack	130	Proxy	919
Hashicorp	1316	Webserver (Apache)	1073
hawkey	636	hwclock	570
hdO,O (GRUB)	834	HWE-Pakete	130
HDR-Bilder	247	hydra	1061
Heimatverzeichnis	144, 390, 579, 580	Hyper-Threading	907
Hello World	686		
help	340	I	
HereDoc-Syntax	385		
SSH	487		
HFS-Dateisystem (Apple)	746	i18n	593
Hibernate-Kerneloptionen	908	i915	696
HiDPI-Bildschirme	168	icedax	465
Hintergrundprozesse	434	IcedTea	687
History		ICMP	920, 1253
APT	650	Icon Themes	180
bash	344	iconv	468
Yum/DNF	634	IdentityFile	1066
ZYpp	640	idn	946
hold-Status (dpkg-Kommando)	643	if (bash)	378
Home-Partition	67	ifcfg-enp4s0	950
Home-Server	1013	ifconfig	936
Home-Verzeichnis	144, 390, 426	ifdown	956
host	1152, 1153	ifupdown	957
host.conf	943	Image Magick	463
hostapd	973	Images	
Hostname	921	Docker	1364
DHCP-Client-Konfiguration	985	Docker, Interna	1399
DHCP-Server (Dnsmasq)	981	erzeugen (qemu-img)	1337
einstellen	946	Formate	1336
HOSTNAME-Variable	362	lesen/manipulieren	1357
SUSE	118	libvirt/KVM	1334
hostnamectl	856	VirtualBox	1304
hosts	943	IMAP	1144
hosts allow (Samba)	997	Authentifizierung	1184
hosts deny (Samba)	997	Immunix	1291
hosts.allow (TCP-Wrapper)	1258	includeres (psutils-Kommando)	474
hosts.deny (TCP-Wrapper)	1258	Includes (Apache)	1084
Hotplug-System	608	Indexes (Apache)	1084
Hotspot einrichten (WLAN)	915	indicator-multiload	208
HP-Druckertreiber	1042	Indikatorprogramme (Ubuntu)	208
HPLIP	1042	inet6	953
hplip-toolbox	1042	inet_interfaces (Postfix)	1159
.htaccess-Datei (Apache)	1089	inetd.conf	872
html2ps	471	info	340
html2text	471	Infrastructure-Modus (WLAN)	930
htop	436	init	857
htpasswd	1087	init (Kerneloption)	906
HTST (HTTP Strict Transport Security) ..	1109	Init-System	847
HTTP		CentOS	865
Apache	1073	Debian	867
httpd.conf	1076	Fedora	865
HTTPS	1095	Init-V-Prozess	450, 857

Init-V-Scripts	859	IP-Adresse	920, 924
Kernelparameter	908	IP-Filter	1262
Protokoll	616	ip-Kommando	481
Raspbian	868	addr	935
RHEL	865	addr show	938
SUSE	869	link	935
Ubuntu	867	route	936
Initial-RAM-Disk	823, 906	IP-Nummer	920, 924
initrd (Kerneloption)	906	IP-Ports	920
initrd-Datei	820	Liste	1252
selbst erzeugen	823	ip6tables	1265
initrd (GRUB)	835	IPng	925
initrd16/initrdefi (GRUB)	836	IPP	1042
inittab	858	iptables	1265
inkrementelle Backups	1238	Beispiel	1274
InnoTek	1299	Masquerading	971
inputrc	343	IPv6	
insmod	877	Apache	1079
insserv	860	deaktivieren	906
install (in modprobe.conf)	882	Debian, Ubuntu	953
Installation	47	DenyHosts	1062
Anleitungen	83	Dovecot	1182
Benutzerverwaltung	72	Fedora, Red Hat	951
externe Festplatten	45	Firewall	1265, 1274
Grundkonfiguration	72	Grundlagen	925
Grundlagen	39	Mail-Server	1150
Linux deinstallieren	81	manuelle Konfiguration	938
Netzwerkinstallation	45	MySQL und MariaDB	1127
Netzwerkkonfiguration	73	Nameserver	945
Probleme	75	NFS	1027
root-Passwort	72	Postfix	1160
SUSE	112	Samba	997
Tastaturprobleme	75	SSH-Server	1060
Updates	78	TCP-Wrapper	1259
Varianten	44	IR-Empfänger	315
inted	871	IR-Fernbedienung	291
Interface (Netzwerkschnittstelle)	922	irrecord	291
interfaces	996	irw	293
interfaces (Samba-Konfiguration)	952	ISO-10646-Zeichensatz	593
Internationalisierter Domainname	946	ISO-8859-Zeichensätze	593
Internationalisierung	592	iso9660-Dateisystem	746, 754, 784
Internet		iw	932
Gateway	961		
Gateway (Client-Konfiguration)	944	J	
Gateway (Server-Konfiguration)	970		
Masquerading	970		
Netzwerkgrundlagen	920	j-Kommando	392
Printing Protocol (IPP)	1042	J8-Header	299
Router	970	Java	687
Sicherheit	1251	javac	688
Internet Service Daemon	870	jed	335, 519
ionice	440, 1245	Jessie	93
iotop	436	JetDirect (HP-Netzwerkdrucker)	1049

jmacs 335, 519
Jobs regelmäßig ausführen 454, 459
joe 337, 519
Jokerzeichen 355, 396
 Komplikationen 397
Joliet-Extension 746, 784
journal (Journaling-Modus) 759
Journal (systemd) 612
journal.conf 621
journalctl 620, 866
Journaling-Dateisysteme 755
 btrfs 764
 ext3/ext4 758
 xfs 778
jove 335, 519
jpico 337
jumpstats 392

K

K3b 199
kacpid 452
Kaffeine 256
Kalender
 Evolution 231
 Gnome 164
 Lightning (Thunderbird) 228
Kali Linux 30
Kamera (Raspberry Pi) 319
Kanal (WLAN) 931
Kazam 266
kbd 568
kblockd 452
KDE 190
 Bildschirmeinstellungen 719
 Screenshots 264
 Verzeichnis freigeben 1012
KDE Wallet 199
kdenetwork4-filessharing 1012
kdevtmpfs 452
Kernel 26, 890
 Boot-Optionen 904
 Boot-Optionen (GRUB) 826
 Device Trees 882
 Dokumentation 891
 Einstellungen ändern 908
 Hotplug-Funktion 608
 installieren 898
 IP-Filter 1262
 kompilieren 889, 897
 Konfiguration feststellen 894
 konfigurieren 894

Logging 615
 Module 876
 neueste Version 892
 Optionen 882
 Optionen (GRUB) 826
 Parameter 908
 Patches 892
 Prozesse 452
 Update (GRUB) 825
Kernel Mode Setting 694
kernel.img-Datei 281
kexec 899
keyboard-setup 568
keymap.cson-Datei (Atom) 553
keys-Dateien (Gnome MIME) 182
kGraft 901
khelperd 452
kill 439
killall 439
KIPI 241
Klammererweiterung 356
kmod 876, 880
KMS 691, 694
 video (Kerneloption) 907
knfsd 452
Kodi 283
Kommandos 433
 ausführen 352
 bedingt ausführen 353
 Eingabe 344
 im Hintergrund ausführen 353
 Kommandointerpreter 341
 regelmäßig ausführen 454, 459
 siehe auch Prozesse 433
 starten 434
 starten (bash) 345
 Substitution (bash) 357
Konfiguration 563
 bash 343
 Benutzer einrichten 576
 Dateisystem 752
 Kernel 889, 894
 LAN 911
 Maus 569
 Netzwerk 911
 Passwort 583
 Prompt 343
 Schriftart 569
 Tastatur (Textkonsole) 567
 Textkonsole 567
 X 710
 Zeitzone 570

Konsole 331
 Schriftart 569
 Tastatur 567
 wechseln 330
Kontakt 232
Kontakte
 Evolution 231
 Gnome 164
 Thunderbird 227
Konverter 463
 Audio 465
kpartx 1358
kPatch 901
krandrcc 719
KRename 198
ksh 342
ksnapshot 264
ksoftirqd 452
Ksplice 900
kswapd 452
ksysguard 437
kthread 452
KTorrent 237
Kubuntu 31, 122
KVM 1330, 1331
 Backup 1245
 Vagrant 1353
kvm-ok 1330
kworker 452

L

liOn 593
Lüftersteuerung 604
Label
 /etc/fstab 752
 root-Kerneloption 905
lame 466
LAMP-Server 1074
 WSL 1415
LAN 911
 NetworkManager 913
 Netzwerkkonfiguration 920
 Sicherheit 1251
Landscape (Ubuntu) 564
LANG 596, 597
language-selector-gnome 676
lapi (Kerneloption) 907
L^AT_EX 476
Latin-Zeichensätze 593
Laufwerke (gnome-disks) 743
Laufwerksbuchstaben (C:, D:) 724

Lautstärke 610
LC_ALL 597
LC_COLLATE 596
LC_CTYPE 596
LC_MESSAGES 596
LC_MONETARY 596
LC_NUMERIC 596
LC_PAPER 596
LC_TIME 596
LC_TYPE 592
ldconfig 679
ldd 588, 678
LD_LIBRARY_PATH 679
ld.so 679
Leap (openSUSE) 110
Lenses (Unity) 206
Lesezeichen synchronisieren (Firefox) 215
less 334
 /proc-Dateien 903
let 361
Let's Encrypt (Zertifikate) 1105
LFS 757
lftp 495
LGPL 33
/lib
 /firmware 933
 /modules//modules.dep* 881
 /modules 876, 877, 898
libav-tools 467
libc 678
libcap 425
libdbus 609
libdrm 695
libgudev 608
libguestfs 1358, 1359
libinput 691, 715
libinput-list-devices 715
libpam-google-authenticator 1069
Libraries 677
LibreELEC 284
libsvg2 465
libsolv 636
libtiff 465
libudisk2 608
libvirt 1332
 SSH 1340
 Vagrant 1353
libvirtd 1332
libwrap 1259
libzypp 638
lightdm 707
Lightning 1222
Limit 1084

Line Feed	468	Logical Volume Manager	59, 798
Link-Local-Adressen (IPv6)	926	Login	139
Links (Hard und Soft Links)	400	<i>Name</i>	579
Linus Torvalds	35	login.defs	583
Linux	25	loginitctl	855
<i>deinstallieren</i>	81	Logische Partition	53
<i>Distribution</i>	28	LogLevel	1082
<i>Entstehung</i>	32	LOGNAME-Variable	362
<i>Installation</i>	39, 47	logrotate	616
<i>Kernel kompilieren</i>	889	<i>Apache</i>	108
<i>Kernelmodule</i>	876	<i>Samba</i>	999
<i>Konfiguration</i>	563	logwatch	617
<i>Linux Standard Base</i>	29	Lokale Netze	911
<i>Shutdown</i>	141	<i>Sicherheit</i>	1251
<i>Startprobleme</i>	76	lokale Variablen	360
<i>Systemveränderungen</i>	78	Lokalisierung	592
<i>Updates</i>	78	Loopback-Device	748
<i>Voraussetzungen</i>	39	Loopback-Interface	922
linux-Schlüsselwort (GRUB)	835	lostfound	427, 762
Linux Mint	106, 122	lp	1045
<i>Paketverwaltung</i>	669	lpadmin	1041, 1046
LIRC	291	lpc	1046
lircd	293, 316	lpd	1040
Listen	1087	lpinfo	1046
Live-System	29, 45	lptions	1040, 1046
<i>Ubuntu</i>	123	lpq	1045
Livna-Paketquelle	105	lprm	1046
Lizenzen	32	lpstat	1046, 1049
llvmpipe	704	ls	393
lm-sensors	600	LSB	29
ln	400	lsblk	599, 731
.local-Verzeichnis	183	lsmod	878
locale	597	lsuf	1255
localectl	568, 595, 596, 856	lspci	599, 605, 878
Locales/Internationalisierung	592	lspeci	605, 703
localhost	921, 943	lsusb	599, 605
localmodconfig	896	LTS-Version (Ubuntu)	121
local_recipient_maps	1172	<i>Enablement Stack</i>	130
locate	405	<i>Support Status</i>	674
lockd	452	Lubuntu	122
log file	998	LUKS	809
Logrotate	1081	luksFormat	811
logger	615	Luminance HDR	247
Logging	611	lvcreate	800, 1244
<i>Apache</i>	1081	lvextend	801
<i>Docker</i>	1375	LVM	798
<i>Logrotate</i>	616	<i>Backup mit Snapshots</i>	1244
<i>Logwatch</i>	617	<i>Grundlagen</i>	59
<i>MySQL</i>	1141	<i>RAID</i>	798
<i>Postfix</i>	1162	<i>Snapshots</i>	802
<i>Samba</i>	998	<i>TRIM</i>	808
<i>X</i>	704	lvremove	1244
Logical Volume	60	LXDE (Raspbian)	279

LXSS	1417	Markdown	477
Lynx	471, 496	Emacs-Erweiterung	544
lzop	1232, 1233, 1245	Masquerading	970
		Fedora	972
		FTP	493, 973
		Probleme	973
		Master Boot Record	822
		wiederherstellen	82
m-a (module-assistent)	888	master.cf-Datei (Postfix)	1161
m23	564, 625	MATE	185
MAC	1257, 1284	Maus	
MAC-Adresse	922, 981	blockiert	439
feststellen	938	KDE	203
mac80211-Framework	933	per Tastatur steuern	143
Machine Owner Keys (Secure Boot)	821	Textmodus	569
macOS		X	143
Dateisystem	746	max log size	999
Samba	1024	maxcpus (Kerneloption)	907
Time Machine	1032	mb	1249
Macromedia Flash	218	mbox-Format	1146
MacVTap-Device	1337	Mbox-Postfach	1155
madplay	466	MBR	734, 822
Magic-Dateien	404	Partitonsnummern	730
Mail siehe E-Mail	1143	wiederherstellen	82
MAIL (Variable)	362	md	452
Mail-Server	1143	MDA	1144
Fehlersuche	1205	mdadm	790, 793
IPv4	1150	mdadm.conf	791
Mailbox	1146, 1155	md_mod (LVM)	798
Dovecot	1183	md_mod (RAID)	791
mailcap	404	mdnsd	451
maildir-Format	1146	/media	427
Maildir-Postfach		Medien-Server	1013
Dovecot	1183	Medienfreigabe	159
Mutt	498	medusa	1061
Postfix	1170	Meld	160
mailq	1162	MELPA (Emacs-Erweiterungen)	544
Main-Pakete	666	Memtest86	601
main.cf-Datei (Postfix)	1156, 1157	mencoder	468
Major Device Number	429	menu.lst (GRUB)	827
make	684	Mesa-Bibliothek	691, 694
make-ssl-cert	1102	mesa-utils	704, 1307
makepasswd	584, 1059, 1174	mhddfs	748
makethumbs	368	Microsoft	
man	339	Exchange Server	1145
Mandatory Access Control	1284	Joliet-Extension	746
Mangle-Tabelle (iptables)	1264	KVM-Installation	1346
Manjero	29	SMB-Protokoll	988
Manuelle Netzwerkkonfiguration	948	Subsystem for Linux	1405
map to guest = bad user	1009	TrueType-Fonts	173
mapfile	375	VSCoDe	555
MariaDB	1123	Windows-Partitionen	780
als Docker-Container ausführen	1374	WSL	1405

Midori	222	mogrify	464
migration	452	MOKs (Secure Boot)	821
Milter		Monitor (X-Konfiguration)	711
<i>ClamAV</i>	1193	monitors.xml	718
<i>OpenDKIM</i>	1202	Monolithischer Kernel	895
<i>SpamAssassin</i>	1188	more	334
MIME		Mosh	1068
<i>CUPS (drucken)</i>	1041	mount	750, 751
<i>Firefox</i>	216	<i>Beispiele</i>	751
<i>Gnome</i>	182	<i>Optionen</i>	754
<i>KDE</i>	203	<i>remount für Systempartition</i>	751
<i>Konfiguration</i>	402	mp32ogg	466
mime.convs	1041	mpage	470
mime.types	404, 1041	MPEG-2-Decodierer	288
Minor Device Number	429	mpg123	466
Mint	31, 106	mpg321	466
<i>mintbackup</i>	109	MPlayer	256
<i>mintdrivers</i>	109	msdos-Dateisystem	746
<i>mintinstall</i>	109, 669	msttcorefonts	173
<i>mintnanny</i>	109	MTA	1144
<i>mintstick</i>	109	mtab	750
<i>mintupdate</i>	108, 669	MUA	1144
<i>Paketverwaltung</i>	669	Muffin (Window Manager)	108
Mirroring	58	Mule (Emacs)	545
MIT-Lizenz	33	Multiarch-Verzeichnisse	680
mkconf	791	Multicast-Adressen (IPv6)	926
mke2fs	760	multiuser-Target	705
mkfs.btrfs	765	MultiViews	1084
mkfs.ntfs	784	Munin	437
mkfs.xfs	779	Musik (Gnome)	251
mkinitramfs	824	Musik-Verzeichnis	183
mklabel	735	Musique	254
mkpasswd	584, 1059	mutt	497
mkswap	789, 790	Mutter-Programm (Gnome Shell)	691
mlocate	406	mv	398
/mnt	427	<i>Dateien umbenennen</i>	398
mode2	316	<i>Sicherheitsabfragen</i>	105
ModeLine	712	MX-Eintrag (DNS)	1150
modinfo	879	mydestination	1159, 1175
modprobe	878	myhostname	1158
modprobe.conf	880, 935	mylvmbackup	1141
Module	876	mynetworks	1159
<i>Abhängigkeiten</i>	881	myorigin	1158
<i>automatisch laden</i>	881	MySQL	1123
<i>Device Trees</i>	882	<i>Administration</i>	1133
<i>Device-Dateien</i>	881	<i>Backups</i>	1138
<i>kompilieren</i>	885, 897	<i>IPv6</i>	1127
<i>Optionen</i>	882	<i>mysql-Kommando</i>	1133
<i>Parameter</i>	879	<i>mysqladmin</i>	1134
<i>Versioning</i>	877	<i>mysqldump</i>	1139
<i>verwenden</i>	877	<i>Workbench</i>	1135
module-assistant	888	<i>WSL</i>	1415
modules.dep	881	Mythbuntu	122

N		newaliases	1147, 1171
Nachtmodus	168	newgrp	
nachträgliche Installation	78	<i>Beispiel</i>	418
Nagios	437	Nextcloud	1207
Name Service Switch	590	<i>Backups</i>	1215
Nameserver		<i>Dateien synchronisieren</i>	1217
<i>Client-Konfiguration</i>	923, 944	<i>Interna</i>	1215
<i>Server-Konfiguration (Dnsmasq)</i>	978	<i>Updates</i>	1216
<i>Ubuntu</i>	917	nfs-Dateisystem	746
Namespaces (Docker)	1385	NFS	1025
NameVirtualHost	1090	<i>/etc/fstab</i>	1030
nano	337	<i>Geschwindigkeit (Server)</i>	1027
NAS-Geräte (Backups)	1231	<i>IPv6</i>	1027
NAT	970	<i>NFS 4</i>	1025
NAT-Tabelle (iptables)	1264	<i>root</i>	1028
Nautilus		<i>Server</i>	1025
<i>MIME</i>	182	nfsd	452
<i>nautilus-compare</i>	160	nft	1266
<i>nautilus-image-converter</i>	160	nftables	1266
<i>nautilus-image-manipulator</i>	160	nginx	1074
<i>nautilus-open-terminal</i>	160	NIC	921
<i>nautilus-pastebin</i>	160	nice	440
<i>nautilus-share</i>	1011	nl80211-Schnittstelle	933
<i>Verzeichnis freigeben</i>	1011	nmap	1255
ncrack	1061	nmbd	992
negativo-Paketquelle (Fedora)	700	nmcli	916
Nemo (Datei Manager)	108	noapic (Kerneloption)	907
Neon	31, 190	noauto	754
net-tools	1254	nodeadkeys	567
Netatalk	1032	nodev	754
NetBIOS	988	nodev-Dateisysteme	747
Netfilter	1262	noexec	754
Netpbm	464	nohide (NFS)	1028
netplan	957	noht (Kerneloption)	907
netstat	1254	nolapic (Kerneloption)	907
Network File System	746	nomodeset (Kerneloption)	907
Network Time Protocol	572	Non-Free-Pakete	666
Network-Maske	922	none-Dateisystem	748
networkd	956	noresume (Kerneloption)	908
NetworkManager	911	no_root_squash (NFS)	1028
Netzwerk	911	nosmp (Kerneloption)	907
<i>Aktivität überwachen</i>	1254	no_subtree_check (NFS)	1028
<i>Brücke</i>	1354	nosuid	754
<i>Ethernet-Controller konfigurieren</i>	934	Notebook	
<i>Drucker</i>	1049	<i>Batterie</i>	601
<i>Grundlagen</i>	920	<i>Lüftersteuerung</i>	604
<i>Konfiguration</i>	948	Notfall	
<i>Netzwerk-Controller</i>	934	<i>Dateisystem reparieren</i>	756
<i>Schnittstelle</i>	922	<i>Init-V-Prozess umgehen</i>	906
<i>Server-Konfiguration</i>	961	<i>Linux-Startprobleme</i>	76
<i>Sicherheit</i>	1251	<i>Windows-Startprobleme</i>	77
Neustart des Hostsystems	1334	NPAPI-Plugins	217
		nplan	957

nproc 600
nscd 591, 592
NSS 591
ntfs-Dateisystem 746, 780
 Streams 783
ntfsclose 784
ntfsinfo 784
ntfslabel 784
ntfsprogs 726, 781, 784
ntfsresize 784
ntfsundelete 784
NTP 572
ntpd 572
ntpdate 572
ntpq 575
Nuvola Player 255
nvidia-Treiber (X)
 Debian 99
 Fedora 106
 nvidia-settings 702
 openSUSE 120
 Treiberinstallation 700
 Ubuntu 129
NWID (WLAN) 931

O

OCFS 723
ocfs-Dateisystem 748
OCICLI 671
Öffentlich-Verzeichnis 183
oggdec 466
oggenc 466
One-Click-Install (openSUSE) 671
Oneshot-Typ (systemd) 862
Online-Dokumentation 145
Online-Konten (Gnome) 164
Open GL 694
Open Source 32
openbsd-inetd 871
OpenCL 694
OpenDKIM 1197
OpenJDK 687
openresolv 946, 949
openssh 1057
openssl 1097, 1166
openSUSE 31
 Samba 1011
 Snapper 775
OpenWrt 963
/opt 427
Optimus-Hybrid-Grafik 697
Optionen (Kernel) 894

options (modprobe.conf) 882
Options (Apache) 1084
Oracle
 Cluster Filesystem 723
 Java 687
 Linux 31, 85
 MySQL 1123
 VirtualBox 1299
Order (Apache) 1085
ordered (Journaling-Modus) 758
Origin-Patterns 651
os-prober 833
OSMX 284
Overclocking (Raspberry Pi) 320
Overlay-Dateisystem 748
 Docker 1399
Overlays (Device Trees) 883
ownCloud 1207
owner 754

P

P1-Header 299
p7zip 1233
PackageKit 655
packagekitd 655
Packer 1316
PAE 886
Pakete 627
 Abhängigkeiten 628
 Debian 640, 666
 Format ändern 656
 Multiarch 680
 Paketmanager 670
 Red Hat 627
 Ubuntu 673
 Verwaltung 623
Paketfilter 1262
PAM 587
 Google Authenticator 1069
 pam-auth-update 588
 pam_cracklib 584
 pam_faillock 585
 pam_pwquality 584
 pam_unix 584
 systemd 854
Pandoc 479
 Atom 554
Panel
 Gnome 148
 KDE 193
 Unity 208

Papierkorb (Samba) 1010
Parallel SSH 1068
Parametersubstitution 375
Paravirtualisierung 1335
Parity Striping 58
parted 737
 EFI-Partition 818
Partition
 ändern, Linux 64
 Bezeichnung unter Linux 728
 Dateisystem 69
 EFI 818
 Grundlagen 51
 ideale Partitionierung 66
 im Verzeichnisbaum 724
 Partitionsname 752
 remount 751
 Typen 53
passdb-backend 994, 1001
PasswordAuthentication (sshd_config) 1412
Passwort 581
 ändern 583
 Ablaufdatum (chage) 583
 aging 583
 Apache 1087
 für Gruppen 586
 PAM 587
 Qualität 584
 root 583
 Samba 1000
 vergessen 584
patch-Kommando 683, 893
Patches (Kernel) 892
Patente 36
path 1006
PATH 345, 362
 Einstellung ändern 362
Pattern (ZYpp) 640
pavucontrol 258, 611
pci (Kerneloption) 906, 908
PCI-Bus 605
PCM-Lautstärke 610
pdbedit 1001
PDC 991
PDF
 pdf2ps 472
 pdf90 476
 pdfedit 476
 pdfimages 476
 pdfinfo 476
 pdfjam 476
 pdfjoin 476
 pdfnup 476

pdftops 472
 pdftotext 476
 pdksh 342
 PostScript-Konverter 471
 Tools 475
Pepper-Plugins (Flash) 219
Perfect Forward Secrecy 1167
Periodische Ausführung von Jobs 454, 459
pesign 892
PGP 1150
Phonon 611
PHP 1116
 Emacs-Erweiterung 544
 phpMyAdmin 1136
 Unicode 1079
Physical Device 60
Physical Extent 60
Physical Volume 60
pico 337
PID 436
PID-Datei 437
pidof 437
pinfo 340
ping 482
pip 1247
Pipes 350
Pixel-Desktop 279
pkcon 655
pkexec 442, 448
pkmon 655
Plasma 190
Plasmoids 192
Plesk Panel 564
Plex 284
Pluggable Authentication Modules 587
Plugins
 Firefox 217
 Flash 218
 Yum 633
pnuke 1068
Policy-Dateien (X) 449
polycoreutils-gui 1287
PolicyKit 447
POP-Server 1144, 1179
 Authentifizierung 1184
Poppler 476
Port-Nummer
 FTP (20, 21) 1252
 HTTP (80) 1252
 IMAP (25, 587) 1146
 Liste 1252
 Referenz 1252
 SMTP (25, 587) 1146

Port-Scan 1255

Portable Bitmap Utilities 464

Portland-Projekt 182

Ports (TCP/IP) 920

Liste 1252

postconf 1160

Postfach

Mbox-Format 1155

virtuell 1176

Postfix 1154

Alias 1171

IPv6 1160

Logging 1162

virtuelle Domänen 1175

postgrey 1191

postmap 1157, 1173

postqueue 1162

PostScript 1037

DSC 475

HTML-Konverter 471

PDF-Konverter 471

Printer Definition (PPD) 1041

Text-Konverter 469

Unicode-Konverter 471

Utilities 474

powertop 602

PPAPI-Plugins 219

PPAs (Ubuntu) 674

PPD-Dateien 1041

ppds.dat 1042

PPP 920

Präfix-Notation (Netzwerkadressen) 922

Pre Shared Key (WPA) 932

prelink 681

Presto (Yum) 634

pri (Swap-Priorität) 788

primäre Partition 53

Primary Domain Controller 991

printcap

CUPS 1040

printenv 361

printers.conf 1040

/proc 427, 747, 902

/asound 609

/sys 908

/config.gz 895

/crypto 811

/mounts 750

/pci 605

Procmail 1144

profile-Dateien 360, 362

Programm 433

kompilieren 682

starten 434

starten (bash) 345

Prompt (bash) 343

PROMPT_COMMAND (Variable) 344, 362

Protokoll-Dateien (Logging) 611

Provisioning (Vagrant) 1324

Proxy-Konfiguration 919

Client (Firefox) 215

Prozesse 433

gewaltsam beenden 439

Größe begrenzen 440

Hierarchie 438

Hintergrundprozesse 434

Priorität 440

Rechenzeit 440

regelmäßig ausführen 454, 459

unter anderer Identität ausführen 441

unterbrechen 435

verwalten 435

Vordergrundprozesse 434

ps 435

PS1 (Variable) 343, 362

ps2pdf 471

psbook 474

psnup 474

psresize 474

psselect 474

pssh 1068

pstops 474

pstree 438

psutils 474

PulseAudio 611

push-Kommando (Docker) 1394

Q

QCOW2-Format 1336

QED-Format 1336

QEMU 1330

qemu-img 1337

qemu-kvm 1337

Qt 190

Quellpaket 628

queue (Druckerwarteschlange) 1038

quiet (Kerneloption) 906

Quotas 723

R

radeon-Treiber 696

Firmware-Dateien 99

RAID 57

LVM 798

RAID-0 58

RAID-1 58

RAID-10 58

Scrubbing 797

TRIM 808

Überwachung 791

RANDOM-Variable 363

RANDOM_DELAY-Variable 459

RandR 694, 716

Raspberry Pi 267

Device Trees 882

Kamera 319

Kodi 283

Raspbian 272

Raspbian 31, 271, 272

NTP 575

Systemstart 868

WLAN-Konfiguration 954

Raspbmc 284

raspistill 319

raspidvid 319

Rasplex 284

RAW-Bilddateien 246

RAW-Format 246, 1336

rb 1249

rc-Dateien 859

rc.local 867

rdiff-backup 1238

RDP-Server 692

Wayland 692

read 377

readline 343

Reboot des Hostsystems 1334

reboot-required-Datei 652

Rechnername siehe Hostname 985

Rechnerstart 817

Probleme 76

recode 468

recordMyDesktop 265

recover-file (Emacs) 521

recycle 1010

Red Hat 31, 84

automount 747

Gateway-Konfigurationsdatei 944

initrd-Datei 825

LABEL in /etc/fstab 752

RHN (Red Hat Network) 564, 625

statische Netzwerkkonfiguration 950

sudo 446

redshift-Programm 168

ReFS-Dateisystem 780

Regelmäßige Ausführung von Jobs 454, 459

Reguläre Ausdrücke (Emacs) 535

reject 1046

Rekonq 222

relayhost 1159

reload (Init-V-Prozess) 859

RemainAfterExit-Schlüsselwort
 (systemd) 862

Remmina 171

remount (Systempartition) 751

remove (modprobe.conf) 882

Remove-Unused-Dependencies (APT) 652

Rendezvous 958

renice 440

Require 1085, 1089

reserve (Kerneloption) 906

reset 335

resize2fs 762

resolv.conf-Datei 944

Ubuntu 945

resolvconf-Paket 946, 953

restart (Init-V-Prozess) 859

restorecon 1286, 1290

Retina-Bildschirme 168

Reverse DNS 1153

RFCs 146

RHEL 84

Systemstart 865

Tastatur 568

RHN 564

RHSM 84

Rhythmbox 251

Richard Stallman 35

Ripper (CDs/DVDs einlesen) 261

rlogin 485

rm-Sicherheitsabfragen 105

rmmod 879

ro (Kerneloption) 906

Rockridge-Extension 746, 784

Rolling Release 111

Ubuntu-Kernel 130

/root 427

root 72, 583

Kerneloption 905

MySQL 1127

NFS 1028

Root-Partition 66

root-Passwort vergessen 584

Root-Server 1057

root_squash (NFS) 1028
route 936
Router (Masquerading) 970
Routing-Tabelle 922
rpc.idmapd 1026
rpcinfo 1031
rpciod 452
rpi-update 281
rpm 627
 Beispiele 629
 cannot open packages database 629
 Datenbank reparieren 629
 Quellcodepakete installieren 683
RPM Fusion 105
RPMS 628
rsnapshot 1240
rsvg 465
rsvg-convert 465
rsync 1235
rsyslog.conf 612
rsyslogd 612
Ruhezustand 602
/run 427, 747
 /log/journal 620
RUN (Dockerfile) 1391
run-crons 458
run-Kommando (Docker) 1381
run-parts 458
Runlevel 858
runtime linker 679
rygel 1012

S

S/MIME 1150
S3-Cloud-Dienst 1246
Samba 987, 988
 /etc/fstab 1022
 Fedora 1011
 Firewall 996
 Gäste 1009
 Inbetriebnahme 992
 IPv6 997
 Nautilus 157
 Netzwerkverzeichnisse einrichten 1006
 Papierkorb 1010
 Passwörter 1000
 RHEL 1011
 SELinux 998
 Sicherheitsmechanismen 990
 SUSE 1011
 Ubuntu 1011

Sandbox (Flatpak/Snap) 660
/sbin 427
 /init 857, 858
 /init.d 869
Schlüssel
 HTTPS (Apache) 1095
 POP/SMTP (Dovecot) 1184
 SSH 1064
Schlafmodus 601
Schleifen 381
Schnittstelle 922
Schriftart 173, 595
 Emacs 542
 Textkonsolen 569
Scientific Linux 30, 85
scp 488
Screen-Abschnitt (X) 714
Screencast 264
Screenshots 264
 Wayland 692
Script
 bash 371
 Programmierung 364
ScriptAlias 1083
Scripts
 bash 363
 SSH 487
 Vagrant 1324
Scrubbing (RAID) 797
SCSI 728
scsi_eh 452
SD-Karte formatieren 725
seahorse 1065
seahorse-nautilus 160
search (GRUB) 835
Secure Boot 43, 77, 820
Secure Shell 1057
 WSL 1412
Secure Sockets Layer 1095
security (Samba) 994
securityfs 1292
sed-Beispiel 398
Selektor (Syslog) 612
SELinux 1283, 1284
 AirPrint 1053
 Apache 1075
 opendkim 1199
 Samba 998
 selinux-policy-mls 1287
 SSH 1066
 SSH-Port 1060
Sender Policy Framework 1195
sensors 600

Server
 crond 454
 Datenbank (MySQL) 1123
 DHCP 976
 FTP (vsftpd) 1118
 Nameserver (DNS) 977
 Netzwerk 961
 NFS 1025
 Samba 988
 SSH 1057
 Webserver (Apache) 1073
 X 690
Server Message Block 988
server role (Samba) 994
server string 994
ServerAdmin 1082
ServerAlias 1090
ServerName 1090
ServerName (Apache) 1079
ServerSignature 1082
service-Kommando 857, 859
services-Datei 871
Services (Hintergrunddienste) 450
sestatus 1289
set 356
set 361
setcap 425
setenforce 1290
setfacl 422
setfattr 424
Setgid-Bit 416
setsebool 1288
Setuid-Bit 415
setup.exe 627
sfconvert 467
sftp 493
 Server 1059
SGI-Dateisystem 745
shadow 582
/share 428
Share-Level-Sicherheit 990
Shared Folder (VirtualBox) 1311
Shared Libraries 677, 678
Shares (Samba) 990
Sharing (Vagrant) 1327
Shebang 364
Shell 341
 Programmierung 363
 Script-Beispiele 363, 461
 Standard-Shell ändern 342
 Variablen 359, 371
Shim 44, 820
shopt 356

Shotwell 238
showmount 1031
shutdown 141
 des Hostsystems 1334
Shutter 265
Shuttleworth, Mark (Ubuntu) 120
Sicherheit 1251
 Apache 1087
 WLAN 931
Sicherheitskontext 1285
Sid 667
Simple Storage Service (S3) 1246
single (Kerneloption) 906
Single-User-Modus (systemd) 852
Site-Local-Adressen (IPv6) 926
skip-networking (MySQL/MariaDB) 1126
Smack 1285
SMART 803
smartd 806
SMB-Protokoll 988
 Version 1 deaktivieren 997
SMB-Versionen 989
smb.conf 993
smbclient 1023
smbd 992
smbfs-Dateisystem 746, 1020
smbpasswd 1001
smbstatus 993, 996
smbtree 1024
SMTP 1146
 Authentifizierung 1185
 Fehlersuche 1205
smtp_tls-Parameter (Postfix) 1164
smtpd_tls-Parameter (Postfix) 1164
Snakeoil-Zertifikat und -Schlüssel
 Apache 1102
 Postfix 1162
Snap (Ubuntu) 662
snapd 664
Snapper (openSUSE) 775
Snapshots
 btrfs 770
 LVM 802
socat 1351
Socket-API (Netzwerkdrucker) 1049
Socket-Dateien 400
soft_bounce (Postfix) 1160
Software-Installation 623
Software-Patente 36
software-properties 675
software-properties-gtx 645
Software-RAID 57
Solus Desktop 31

Sonderzeichen (bash)	386	SSLCipherSuite (Apache)	1101, 1104
Sound Converter	259	SSLEngine (Apache)	1101
Sound Juicer	261	SSLProtocol (Apache)	1101, 1104
Sound-System (ALSA)	609	SSLStrictSNIVHostCheck (Apache)	1095
source	372	SSLxxxFile (Apache)	1101
sources.list	644	Stable-Pakete	666
sox	467	Stallman, Richard	33
Spam-Schutz	1187	Standardausgabe	349
spamass-milter	1189	Standardeingabe	349
SpamAssassin	1187	star	423
speaker-test	610	start.elf-Datei	281
special bits (Zugriffsrechte)	415	Startprobleme	76
SPF-Eintrag (Mail-Server)	1195	STARTTLS	
Spice	1337	Dovecot	1184
Spin (Fedora)	100	Postfix	1162
splash	904	startx	709
Spooling-System (drucken)	1038	stat	412
Spracheinstellung	592	Statisch gelinkte Programme	678
squashfs-Dateisystem	748	Statische Netzwerkkonfiguration	948
Snap	664	Sticky-Bit	417, 419
Squeeze	93	Streams (NTFS-Dateisystem)	783
SRPM-Pakete	628, 683	Strict Transport Security	1109
/srv	428	stripcomments (bash-Beispiel)	365
/ftp	1120	Striping	58
/www	1075	Stromsparfunktionen	601
SSD-TRIM	807	su	442
SSH	485, 1057	grafische Variante	442
absichern	1059	Wayland	692
Dateisystem	490	submission (Postfix)	1161
Google Authenticator	1069	Substitutionsmechanismen (bash)	354
Konqueror	199	subtree_check (NFS)	1028
IPv6	1060	Subvolumes (btrfs)	768
libvirt	1340	suchen	
Login vermeiden	1064	Dateien	404
Port ändern	1060	Emacs	534
Portumleitung	1044	find und grep	407
SELinux	1066	sudo	443
Server	1057	Ein-/Ausgabeumleitung	444
Tunnel	488	Fedora	446
unter Windows	1412	Raspbian	280
ssh-agent	1065	Ubuntu	445, 447
ssh-keygen	1064	Wayland	692
sshd	1057	suid	415
WSL	1412	SUSE	
sshfs-Dateisystem	490, 747	AppArmor	1291
SSID	974	CIFS	1022
SSID (WLAN)	930	Firewall	1271
SSL	1095	Gateway-Konfigurationsdatei	944
SSL (Apache)	1095	Init-Prozess	869
ssl-cert-snakeoil.key	1102	Kernelkonfiguration	895
ssl-cert-snakeoil.pem	1102	Paketverwaltung	670
SSLCACertificateFile (Apache)	1103	Samba	1011
SSLCertificateChainFile	1104	Snapper	775

Updates	672	Systemeinstellungen (KDE)	199
VirtualBox	1306	Systempartition	66
Suspend to Disk	601	remount	751
Kerneloptionen	908	systemsettings	199
SVG-Konverter	465	Systemstart	139
Swap-Datei	789	GRUB	817
Swap-Partition	68	Init-V	857
einbinden	787	systemd	848
einrichten	789		
swapon	789, 790	T	
swappiness-Parameter	788		
symbolische Links	401	Tabulatoren (Emacs)	529
Synaptic	653	tail	334
ohne Passwort ausführen	446	Taktfrequenz	600
sync (S3)	1248	tar	683, 1232, 1234
sync (NFS)	1027	targeted	1287
Syntaxhervorhebung	539	Tartarus	1245
/sys	428, 904	tasksel	649
/kernel/security	1292	Tastatur	141
sysctl	908, 970	bash	343
sysfs-Dateisystem	747	blockiert	439
syslog	999	Gnome	162
System Security Services Daemon	592	KDE	203
system-config-lvm	798	Konfiguration	567
system-config-printer	165, 1047	US-Tastaturtabelle	75
system-config-samba	1011	Tastenkürzel	141
system-config-selinux	1287	Linux	141
system-config-services	867	TCP-Wrapper-Bibliothek	1258
system-config-users	576	TCP/IP	920
System-V-Init-Prozess	857	tcsh	342
Systemadministration	563	TDB	1001
systemctl	459, 850	TeamViewer	172
systemd	848	tee	351
als Cron-Ersatz	459	telnet	485, 490
CentOS	865	SMTP-Fehlersuche	1205
eigene Service-Datei	861	Temperatur (CPU)	600
Fedora	865	Temperaturmessung (Raspberry Pi)	313
Firewall-Beispiel	1280	Terminal	330
Grafiksystem starten	705	Termine	
Netzwerk-Device-Namen	947	Evolution	231
Netzwerkkonfiguration	956	Lightning (Thunderbird)	228
Netzwerkschnittstellen	922	test-Kommando (bash)	379
Prozesse periodisch ausführen	459	Tethering	915
RHEL	865	Text-Konverter	468
Timers	459	Textdatei	
systemd-journald	619	durchsuchen	408
systemd-networkd	956	PostScript-Konverter	469
systemd-sysv-generator	856	Texteditoren	335, 519
systemd-timedated	570	Textkonsole	330
systemd-timesyncd	573	Konfiguration	567
systemd-udev	430	Schriftart	569
systemd-udevdev	430	Tastatur	567
systemd-vconsole-setup	569		

Themen (KDE) 202
Themes (Gnome) 179
Thumbnails 368, 463
Thunderbird 222
 CalDAV/CardDAV 1222
tiff2pdf 465
tiff2ps 465
tigervnc-viewer 171
Tilde 144, 390
time-sync 572
timedactl 570, 572, 856
Timers (systemd) 459
TinyCore 32
TLS 1149
 Dovecot 1184
 Postfix 1162
/tmp 428
tmpfs-Dateisystem 747
top 436
Torrent 237
Torvalds, Linus 35
Totem 256
Touchpad deaktivieren 163
traceroute 483
Transmission 237
Transport Layer Security 1149
Treiberinstallation (Ubuntu) 675
TRIM (SSDs) 807
Troll Tech 190
Trusted TLS Connection (Postfix) 1164
TSOP4838 315
Tumbleweed 111, 672
tune2fs 761
Tunnel (SSH) 488
TurboPrint 1047
tvservice 319
type name 346
Type-Schlüsselwort (systemd) 862

U

uappexplorer-cli 664
Ubuntu 31
 AirPrint 1052
 als Docker-Image 1368
 Bildschirmeinstellungen 719
 DKMS 887
 Dnsmasq 917
 initrd-Datei 824
 Paketverwaltung 673
 sudo 445, 447
 Systemstart 867

Tastatur 567
 Unity 204
 VirtualBox 1306
Ubuntu Server 122
Ubuntu Studio 122
ubuntu-drivers 129, 675
ubuntu-restricted-extras 129
ubuntu-support-status 674
udev 430, 608
 Netzwerk-Device-Namen 947
udf-Dateisystem 746, 784
udisks2 609
UDP 920, 1253
UEFI 41
 Partition 43
 Secure Boot 43, 77, 820
ufw 1272
 Docker 1367
Uhrzeit 570
UID 579
ulimit 440
umask 419
Umgebungsvariablen 360
umount 785, 1030
unattended-upgrades 650
Unicode 593
 Apache 1079
 Dateisystem 389
 drucken 471
 Emacs 545
 Konsole 569
 PHP 1079
 PostScript 471
 UTF 593
 Zeichensatz 593
unionfs-Dateisystem 748
Unity 204, 205
Unity Tweak Tool 212
Univention Corporate Server 564
Universal Disk Format 746
Unix 25
Unix Pseudo TTYS 747
unix2dos 468
unset 361
Unstable-Pakete 666, 667
Untrusted TLS Connection (Postfix) 1164
unxz 1232
unzip 1233
update-alternatives 658
update-ca-certificates 1164
update-grub 828
update-initramfs 824
update-manager 676

update-ms-fonts 173
Update-Patch 892
updatedb 406
Updates 78
 LibreELEC 293
UPnP 296
upower 609
Upstart (Ubuntu) 867
US-Tastaturliste 75
USB 605
 Laufwerke 786
 USB-Stick formatieren 725
usb-creator-gtk 129
User einrichten 576
User Shares (Samba) 1009
User Themes 179
User-Level-Sicherheit 990
useradd 577
username map 1003
usershare allow guests 1010
user_xattr 421
/usr 428
UTC (Universal Time, Coordinated) 570
UTF-16 593
UTF-8 593
 MySQL/MariaDB 1126
UUID
 einsetzen (ext3/ext4) 762
 einsetzen (xfs) 780
 ermitteln 752
 in /dev/disk 732
 in /etc/fstab 752

V

Vagrant 1316
 libvirt/KVM 1353
VagrantFile 1317, 1322
valid users 1006
/var 428, 747
 /ftp 1120
 /lib/docker 1400
 /lib/dpkg/alternatives 659
 /lib/rpm/alternatives 659
 /log/Xorg.O.log 704
 /log/journal 620
 /run 437
 /spool/cron/tabs 454
 /www 1075
Variablen (bash) 359, 371, 377
varlock-Dateisystem 747
varrun-Dateisystem 747

vboxadd 1306
vboxdrv 1300
vboxmanage 1314
vboxnetadp 1300
vboxnetflt 1300
vboxpci 1300
vboxsf-Dateisystem 1312
vboxvideo 1306
VCI-Decodierer 288
vcgencmd 289, 322
VDP AU 694
Vergleiche (bash) 379
Verschlüsselung 61
 Dateien 809
 Dateisysteme 722, 809
 Mail-Server 1149
Verzeichnis 144, 390, 425
 Multiarch 680
 Partitionen 724
Verzweigungen (bash) 378
VESA-Modi 714
VESA-Treiber (X) 713
vfat-Dateisystem 746, 780
vga-Treiber (X) 714
vgcreate 800
vgscan 800
Vi 335, 501
video (Kerneloption) 907
Video-Codecs 288
Videos (DVDs) abspielen 786
Videos-Verzeichnis 183
Vim 335, 501
 Cursorbewegung 505
 Easy-Modus 517
 Konfiguration 514
 Makros 517
 Maus 516
 Optionen 513
 suchen und ersetzen 510
 Swap-Datei 515
 Tabulatoren 516
 Unicode 515
 Zeichensatz 515
vimrc-Datei 514
Vinagre 171
Virenschutz 1193
virsh 1332, 1348
virt-cat 1361
virt-clone 1351
virt-df 1359
virt-edit 1361
virt-filesystems 1360
virt-inspector 1360

virt-make-fs 1361
virt-manager 1332, 1339
virt-resize 1361
virt-tar-in 1361
virt-tar-out 1361
virt-top 1353
virt-viewer 1352
virt-viewer vmname 1352
virtio-Treiber 730, 1335
 Windows 1346
Virtual Private Networks 915
virtual_alias_domains 1175
VirtualBox 1299
VirtualHost (Apache) 1090
virtual_mailbox_domains 1177
Virtuelle Dateisysteme 747
Virtuelle Domänen (Postfix) 1175
Virtuelle Hosts 1089
 mit HTTPS 1095
Virtuelle Postfächer 1176
VISUAL 337
Visual Studio Code 555
visudo 444
vmlinuz 898
vmlinuz-Datei 819
VNC 171
 Server 692
 Wayland 692
vncviewer 172
VolFS-Dateisystem 1418
vol_id 752
VOLUME (Dockerfile) 1393
Volume Group 60
Volumes
 Docker 1384
 Docker, löschen 1401
 LVM 60
vorbis-tools 466
Vordergrundprozesse 434
Vorlagen-Verzeichnis 183
VRFY-Kommando (Postfix) 1179
VSCode 555
vsftpd 1119

W

w3m 471, 496
WannaCry-Schad-Software 997
Warteschlange 1038
watchdog 452
Wayland 689
 Einschränkungen 692

Web-Apps (Ubuntu) 205
Webalizer 1112
Webbrowser (Textmodus) 496
WebDAV 1119
Webmin 564
Webserver 1073
website 1249
WebUpd8-Paketquelle (Java) 688
Webverzeichnis absichern 1087
well-known-DAV-Umleitungen 1211
WEP 931
Weston 692
wget 494
Wheezy 93
whereis 346, 405
which 405
while (bash) 383
Whitelist (SpamAssassin) 1190
WiFi (WLAN) 929
Wildcard-Zertifikate 1109
Window Manager 690
Windows
 CUPS-Netzwerkdrucker drucken 1052
 Dateisystem 746, 780
 Drucker 1047
 Hibernate 781
 KVM-Installation 1346
 MBR wiederherstellen 82
 Netzwerkverzeichnisse 988, 1020
 Samba-Freigaben nutzen 1024
 Startprobleme 77
 Subsystem for Linux 1405
winff 467
WINS 988
WiringPi 305
WLAN 929
 Access Point 929, 966
 Access-Point 915
 Adapter 929
 Authenticator (hostapd) 973
 LibreELEC 286
 NetworkManager 914
 Raspberry Pi 282
 Router 929
 Sicherheit 931
wmf2eps 465
wmf2gd 465
wmf2svg 465
workgroup 994
Workgroup-Sicherheit 990
WPA 931, 941, 975
WPA2 931, 941
wpa_passphrase 941

wpa_supplicant 282
wpasupplicant 941
 in /etc/network/interfaces 954
writeable 1006
writeback (Journaling-Modus) 759
WSL 1405

X

X 689
 Auflösung 714
 Farbanzahl 714
 Grafikkarte 713
 Konfiguration 710
 Logging 704
 Maus 143
 Monitor-Konfiguration 711
 Protokoll 704
 Server 690
 SSH 487
 Version feststellen 703
 Window Manager 690
 Window System 689
X11R6 690
xargs 358
XBian 284
XBMC 283
xconsole 615
XDG 183
xdg-desktop-icon 183
xdg-desktop-menu 183
xdg-email 184
xdg-icon-resource 183
xdg-mime 183
xdg-open 184
xdg-screensaver 184
xdg-user-dirs 183
xdg-user-dirs-gtk 183
xdm 707
xdpyinfo 703
xdpyinfo 703
xfs-Dateisystem 745, 778
xfs_admin 780
xfs_check 779
xfs_growfs 780
xfs_repair 779
xine 256
xinetd 871
xinput 715
xkill 439
Xorg.O.log 704
xorg.conf 710

xpdf-utils 476
XPI 217
xrandr 716
XRender 694
xsensors 601
xterm 331
Xubuntu 31, 122
XWayland 691
xz 1232, 1233

Y

YaST 110
 Online Updates 672
 Paketverwaltung 670
 YOU 672
Yorba 234
YOU (YaST Online Update) 669, 672
yum 631, 634
 automatische Updates 636
yum-cron 636
yum-utils 635
yumdownloader 635
yumex 636

Z

Zahlenvergleiche (bash) 379
Zeichenketten
 bash 358
 Parametersubstitution (bash) 375
Zeichensatz 592, 593
 ändern 468
 Apache 1079
 PHP 1079
Zeitgesteuerte Job-Ausführung 454, 459
Zeitzone 570
 glibc 571
Zentyal 122, 564
ZENworks 564, 625
Zero Install 627
Zeroconf 958
Zertifikate
 HTTPS (Apache) 1095
 Let's Encrypt 1106
 POP/SMTP (Dovecot) 1184
 Postfix 1162

selbst erstellen und signieren 1097

Snakeoil-Zertifikat 1102

ZFS-Dateisystem 745

zile 519

zip 1233

zipinfo 1233

Zorin OS 31

zsh 342

Zugriffsbits 410

bei neuen Dateien 419

setuid, setgid 415

sticky 417

Zugriffsrechte 409

Zugriffssteuerung 576

Zwei-Faktor-Authentifizierung 1069

Zwischenablage (VirtualBox) 1311

ZYpp 638

zypper 639



Michael Kofler

Linux – Das umfassende Handbuch

1.450 Seiten, gebunden, 15. Auflage, September 2017
49,90 Euro, ISBN 978-3-8362-5854-8

 www.rheinwerk-verlag.de/4465



Dr. Michael Kofler studierte Telematik an der TU Graz. Er zählt zu den erfolgreichsten und vielseitigsten Computerbuchautoren im deutschen Sprachraum. Zu seinen Themengebieten zählen neben Linux auch OS X, MySQL, KVM, Visual Basic und Excel-VBA. Viele seiner Bücher wurden übersetzt. Michael Kofler arbeitet auch als Software-Entwickler, Berater sowie als Lehrbeauftragter an zwei Fachhochschulen.

Wir hoffen sehr, dass Ihnen diese Leseprobe gefallen hat. Gerne dürfen Sie diese Leseprobe empfehlen und weitergeben, allerdings nur vollständig mit allen Seiten. Die vorliegende Leseprobe ist in all ihren Teilen urheberrechtlich geschützt. Alle Nutzungs- und Verwertungsrechte liegen beim Autor und Verlag.

Teilen Sie Ihre Leseerfahrung mit uns!

