OpenSuse Leap Micro:
Basis für Container S. 56

11.2025

linuxUSER

linuxUSER
CHECKIP • QUARTET • LEAP MICRO • RUST-CLI-TOOLS • YUNOHOST • XARGS • PROFI-BÜRO
11

Optimale Lösungen und handliche Werkzeuge für das heimische Office

PROFI-BÜRO

To-do-Manager: Aufgaben und Termine jederzeit im Griff S. 10

Dokumentenmanagement: Intelligentes papierloses Büro einrichten mit Paperless-NGX und Paperless-AI S. 22, S. 28

PdfDing: PDF-Dokumente unkompliziert verwalten, organisieren und editieren S. 34

Hardware abfragen auf der Kommandozeile S. 64
Im Rechner verbaute Hardware schnell und zuverlässig identifizieren mit Cpu-x, Dmidecode, Inxi, Hw-probe, Hardinfo2 und Nvme-cli

Handlicher Server S. 48
500 Dienste bequem per Mausklick bereitstellen mit YunoHost

Moderne Werkzeuge S. 76
Rust-basierte Kommandozeilen-Tools mit Mehrwert im Überblick

Indie-Rollenspiel Quartet S. 42
Tiefgründige Charaktere, viele fesselnde Abenteuer und taktische Kämpfe

Turbolader fürs Terminal S. 72
Dateianpassungen, Suchvorgänge und Recodierungen mit Xargs beschleunigen

www.linux-user.de

EUR 8,99 Deutschland
EUR 9,70 Österreich
sfr 14,00 Schweiz
EUR 10,40 Benelux

4 195111 008994 11

Biegen und Brechen

Carina Schipper
Stellv. Chefredakteurin

Sie soll das Herzstück der Digitalisierung im Gesundheitswesen sein: die elektronische Patientenakte. Wer als gesetzlich Versicherte(r) nicht widersprochen hat, verfügt inzwischen über eine solche ePA. In Zahlen ausgedrückt betrifft das rund 70 der etwa 74 Millionen gesetzlich Versicherten. Das klingt erst einmal gar nicht schlecht. Seit Beginn des vierten Quartals 2025 sind Leistungserbringer zudem gesetzlich dazu verpflichtet, Diagnosen, Behandlungen sowie Medikationen in die digitale Akte einzutragen. Die technischen Voraussetzungen dazu stehen sogar einigermaßen. Doch bei genauerem Hinsehen krankt das Projekt nach wie vor in längst bekannten Punkten.

Außer an anhaltenden Sicherheitsbedenken, fehlender Software in Arztpraxen und vielfältigen anderen Problemen leidet die ePA vor allem unter einem: Lediglich drei Prozent der Versicherten nutzen sie aktiv. Damit verfehlt das Projekt eines seiner Hauptziele. Die ePA soll eine „versichertengeführte elektronische Akte" sein, so will es zumindest der dazugehörige Paragraph 341 des Sozialgesetzbuchs. Woran dieses offensichtliche Desinteresse liegt, lässt sich einfach beantworten. Eine Erklärung findet sich sicherlich in der Aufklärungs- und Informationspolitik rund um das Thema, die in den Zuständigkeitsbereich der Krankenkassen fällt.

Auf Nachfrage des Deutschen Ärzteblatts berichtet die Techniker Krankenkasse (TK) davon, schon im Mai 2024 mit einer Informationskampagne begonnen und bis heute 10,3 Millionen Menschen benachrichtigt zu haben – immerhin, bei insgesamt 12,1 Millionen Versicherten. Ebenfalls seit Mai 2024 versorgt der AOK-Bundesverband unter www.aok.de/epa die Mitglieder mit Informationsmaterial. Man verzeichnet seither mehr als 2 Millionen Seitenaufrufe – bei 20,9 Millionen Mitgliedern. Nun ja.

Ich selbst habe der ePA nicht widersprochen, obwohl ich mir der berechtigten Kritik bewusst bin. Ernsthaft aktiv mit ihr und der dazugehörigen App meiner Krankenkasse habe ich mich nicht beschäftigt. Noch suche ich nach einem echten Mehrwert. Meine Motivation steigt bestimmt, sobald ich den Zugriff auf meine Gesundheitsdaten tatsächlich sinnvoll einschränken kann. Standardmäßig dürfen nämlich alle behandelnden Einrichtungen sämtliche medizinischen Daten in der ePA einsehen.

Klassisches Beispiel: Mir gelingt es zwar, das Dokument mit der Diagnose von meiner Psychologin für andere zu sperren. Da Medikationslisten und Abrechnungsdaten automatisiert in die ePA einlaufen, kann mein Zahnarzt anhand dieser Anhaltspunkte trotzdem relativ einfach Rückschlüsse ziehen. Selbstverständlich möchte ich aber, dass er sich ausschließlich mit meinem Gebiss befasst. So viel zur „versichertengeführten" elektronischen Akte. Souveränität sieht für mich anders aus.

Bianca Kastl bringt in einem Artikel bei Netzpolitik.org auf den Punkt, wo es aus meiner Sicht hinsichtlich der ePA mit Abstand am schmerzhaftesten hakt: „Die Digitalisierung des Gesundheitswesens ist sehr stark getrieben von der Selbstverwaltung und Interessen von Krankenkassen und Industrie, weniger von den Interessen der Patient:innen."

Herzliche Grüße,

Carina Schipper

Weitere Infos und interessante Links
www.linux-user.de/qr/52047

10 Wir vergleichen sechs **To-do-Manager**, die Ihnen helfen, in der Flut der täglichen Aufgaben nichts zu übersehen oder zu vergessen.

34 **PdfDing** ermöglicht unter einer übersichtlichen Oberfläche das Organisieren, Verwalten und Teilen Ihrer PDF-Sammlung im Team. Daneben bietet es grundlegende Bearbeitungsfunktionen für PDF-Dokumente.

42 Stürzen Sie sich mit dem brandneuen rundenbasierten **Indie-RPG Quartet** von Something Classic Games in ein fesselndes Abenteuer.

96 Das Live-System **Tails 7.0** zum anonymen Surfen über das Tor-Netzwerk liegt in einer neuen Major-Version vor. Sie aktualisiert den Unterbau auf Debian 13 „Trixie“, komprimiert die Pakete mit dem Zstd-Algorithmus und erfordert mindestens 3 GByte Hauptspeicher. Mit dem Umstieg auf „Trixie“ kommt auch der neuere Kernel 6.12.43 zum Zug, der vor allem moderne Hardwarekomponenten unterstützt.

48 Die auf Debian basierende Server-Distribution **YunoHost** bietet derzeit rund 550 mit wenigen Mausklicks installierbare Dienste an.

72 **Xargs** gehört zu den heimlichen Helden unter Linux. Gerade im Zusammenspiel mit Find spielt es alle Trümpfe aus. Das gilt vor allem bei aufwendigen Anpassungen an vorhandenen Dateien wie Kodierungsvorgängen.

76 Rust ersetzt zunehmend in C oder C++ formulierten Code. Wir sehen uns an, welche Rust-Tools für die Kommandozeile es bereits gibt.

Praxis

easyLINUX

Netz&System

Know-how

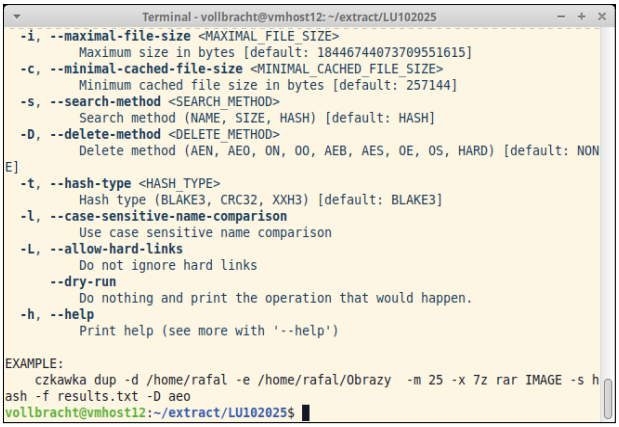
84 Mit **Checkip** erfahren Sie mehr über suspekte IP-Adressen. Damit erhalten Sie bei Bedarf eine Übersicht darüber, wer von außen am System anklopft. Umgekehrt überprüfen Sie Ihre eigenen IP-Adressen hinsichtlich Missbrauchs. Das Tool lässt sich hervorragend in eigenen Shell-Skripten verwenden.

Service

Saubermann

Mit Czkawka 10.0.0 entfernen Sie platzfressende Dubletten auf einfache Weise.

Egal, wie groß die Festplatte oder SSD ist, irgendwann platzt sie doch aus allen Nähten. Häufig liegt das an zahlreichen platzfressenden Dateiduplikaten. Beim Auffinden solcher Dubletten hilft das Rust-basierte Tool Czkawka. Sie müssen es nicht selbst kompilieren, da das Projekt auf Github zahlreiche Binärpakete bereitstellt. Neben einem reinen Kommandozeilen-Tool bieten die Entwickler eine Variante mit grafischer Oberfläche an, die auf GTK4 basiert. Für Rechner ohne das GTK4-Framework steht unter dem Namen Krokiet eine auf Slint basierende Spielart zur Verfügung. Den Verzeichnisbaum, in dem das Tool nach Dubletten suchen soll, geben Sie beim Aufruf als Parameter an. Mit dem Unterbefehl `dup` leiten Sie die Suche nach Dubletten ein. Soll Czkawka auch leere Verzeichnisse entfernen, nutzen Sie den Unterbefehl `empty-directories`. Besonders Multimediadateien wie Bilder, Musik oder Videos beanspruchen viel Platz. Czkawka bietet mit den Unterbefehlen `image`, `video` und `music` eigene Vergleichsroutinen. Musikdateien vergleicht die Software dabei anhand ihrer Tags. Defekte Dateien spüren Sie mit `broken` auf. Das Tool liefert mit `--help` eine einfache Onlinehilfe, die auch Anwendungsbeispiele für die einzelnen Befehle anzeigt. Zum Archivieren der Ergebnisse geben Sie nach `-f` eine Ausgabedatei an. Zudem haben Sie die Möglichkeit, Dateien oder Verzeichnisse vom Vergleich auszuschließen und die Suchmethode anzupassen. Mit `--dry-run` gelingt vorab ein folgenloser Testlauf. Alles in allem bietet die App eine Vielzahl an Einstellungsmöglichkeiten, um Dubletten aufzuspüren, und lässt sich ideal in Skripte integrieren. Haben Sie es lieber bequemer, greifen Sie auf eine der grafischen Versionen zurück.



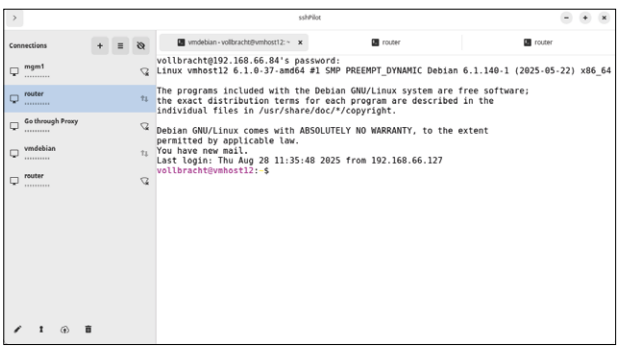
Lizenz: CC-BY-4.0
Quelle: https://github.com/qarmin/czkawka

Telefonbuch

Mit Sshpilot 2.8.6 verwalten Sie SSH-Verbindungen ganz bequem.

Dateien zum Artikel herunterladen unter
www.linux-user.de/dl/52017

Benutzen Sie parallel mehrere SSH-Verbindungen, geht schnell die Übersicht verloren, welcher Terminal mit welchem Rechner verbunden ist. Das GUI-Frontend Sshpilot hilft Ihnen dabei, bequem mehrere Verbindungen zu verwalten. Das Python-basierte Tool greift im Hintergrund auf den Standard-OpenSSH-Client und auf Sshpass zurück, die deshalb beide auf dem Rechner installiert sein müssen. Sie richten die Software wahlweise von Hand ein oder greifen auf eines der DEB- oder RPM-Pakete von der Github-Seite des Projekts zurück. Am linken Fensterrand listet die Software alle aktuell konfigurierten Verbindungen übersichtlich auf. Durch einen Klick auf eine davon öffnen Sie die gewünschte Connection. Ist in der Verbindungskonfiguration kein Schlüssel hinterlegt, fragt Sshpilot nach dem Passwort. Nach erfolgreicher Anmeldung öffnet sich ein Reiter mit dem Terminal des Remote-Systems. Sie können mehrere Verbindungen gleichzeitig offenhalten und zwischen ihnen wechseln. Neue Connections legen Sie über das Konfigurationssymbol links unten im Fenster an. Jede Verbindung erhält einen eindeutigen Namen. Außerdem tragen Sie Hostname und Port, einen Benutzer und eine Authentifizierungsmethode ein. Bei letzterer wählen Sie zwischen einem Passwort, das Sie gleich hinterlegen können, und einem Schlüssel. Daneben besteht auch die Möglichkeit, für Verbindungen eine Port- und eine X.org-Weiterleitung einzurichten. Neben dem Konfigurationssymbol finden Sie ein Wolken-Icon, über das Sie Dateiübertragungen starten. In den globalen Einstellungen der Software passen Sie außer dem Erscheinungsbild globale SSH-Parameter wie Timeouts für die Verbindungen oder Keepalive-Intervalle an.



Lizenz: GPLv3
Quelle: https://github.com/mfat/sshpilot

Schlüsseldienst

Mit Gpg-tui 0.11.1 verwalten Sie komfortabel Ihre GPG-Schlüssel auf der Konsole.

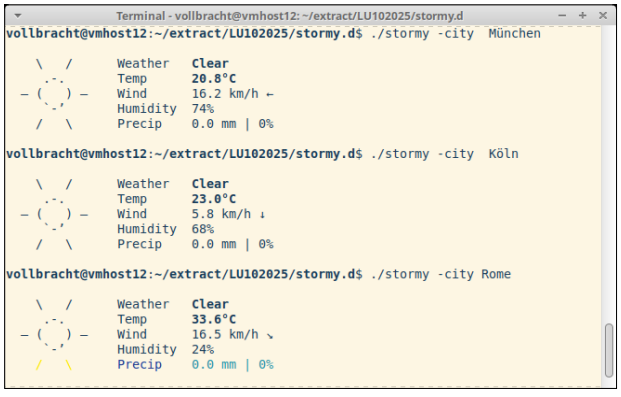
GNU Privacy Guard oder kurz GPG ist für viele Anwender das Mittel der Wahl, wenn es um das Verschlüsseln und das Signieren von E-Mails oder Dateien geht. Auf der Kommandozeile sind das Verwalten der Schlüsselsammlung und das Erzeugen neuer Schlüssel aber mitunter eine umständliche Prozedur. Hier will das Rust-basierte Gpg-tui Abhilfe schaffen. Verwenden Sie die Software auf einem 64-Bit-PC, können Sie auf ein Binärpaket von Github zurückgreifen, für andere Plattformen müssen Sie das Tool selbst kompilieren. Ohne Parameter aufgerufen, wertet die App die in `$HOME/.gnupg/` abgelegten Keys aus und zeigt sie in einer einfachen Curses-Oberfläche an. Damit eignet sich Gpg-tui auch für das Verwenden in Remote-Verbindungen. Die Steuerung erfolgt komplett über Tastenkürzel. Eine Übersicht dazu finden Sie auf der Github-Seite des Projekts. Bevorzugen Sie eine andere Zuordnung, passen Sie das Mapping in einer TOML-basierten Konfigurationsdatei an. Im Github-Repository von Gpg-tui gibt es eine Beispielkonfiguration, die sich als Vorlage eignet. In der Oberfläche des Programms navigieren Sie mit den Pfeiltasten durch die Schlüsselliste. Das Drücken der Eingabetaste über einem Key öffnet ein Menü, in dem Sie zwischen verschiedenen Funktionen wählen. Sie können beispielsweise Schlüssel exportieren, importieren oder an einen Schlüsselserver übertragen. Auch das Erzeugen neuer Schlüsselpaare ermöglicht die App. Gpg-tui zeigt nur die wichtigsten Daten zu jedem Key an; über die Zifferntasten erweitern Sie die Detailanzeige. Mit dem Befehl `:help` gelangen Sie in die Onlinehilfe, die weitere Informationen zu den einzelnen Funktionen bietet und sie bei Bedarf sofort ausführt. Insgesamt präsentiert sich Gpg-tui als guter Helfer für alle, die ihre GPG-Keys bequemer verwalten möchten.

Lizenz: MIT
Quelle: https://github.com/orhun/gpg-tui

Wetterfrosch

Das Tool Stormy 0.3.3 hält Sie über das Wetter auf dem Laufenden.

Wollen Sie wissen, wie das Wetter wird, dann ist das Terminal-Tool Stormy das richtige für Sie. Das Go-basierte Werkzeug steht als Binärpaket auf Github zum Abruf bereit. Um das aktuelle Wetter an einem Ort zu erfahren, geben Sie dessen Namen mit dem Parameter `–city` beim Aufruf an. Die Software zeigt Daten wie Bewölkung, Temperatur, Windstärke, Luftfeuchtigkeit und Luftdruck an. Mit `–compact` erhalten Sie eine kompakte Anzeige. Bei Temperatur und Windgeschwindigkeit wechseln Sie mit dem Schalter `–units` zwischen metrischen und angelsächsischen Einheiten. Seine Daten erhält Stormy von der Plattform Open-Meteo. In der Konfigurationsdatei der App können Sie für die Variable `provider` auch OpenWeatherMap als Datenquelle vorgeben. Die Konfiguration erlaubt es überdies, einen Live-Modus zu aktivieren und die farbige Ausgabe abzuschalten. Anwendungsbeispiele finden Sie auf der Github-Seite des Projekts. Alles in allem ist Stormy ein kleines Tool, das sich auch zur Integration in eigene Skripte oder als Banner beim System-Login eignet.



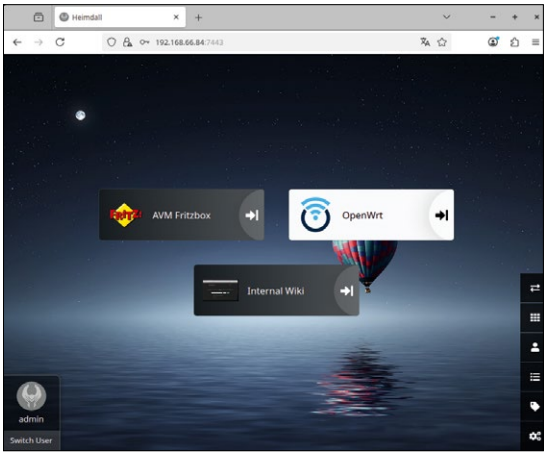
Lizenz: MIT
Quelle: https://github.com/ashish0kumar/stormy

Sammelstelle

Mit Heimdall 2.7.4 behalten Sie selbst gehostete Webdienste bequem im Griff.

In Firmen wie in heimischen Netzwerken sind mittlerweile zahlreiche webbasierte Dienste im Einsatz. Wer sich nicht alle internen URLs merken möchte, um auf die jeweiligen Dienste zuzugreifen, findet in der Dashboard-Lösung Heimdall eine gute Alternative.

Sie betreiben das PHP-basierte Dashboard wahlweise mit einem vorhandenen Webserver oder in einem Docker-Container. Letzteres klappt in der Regel am einfachsten und schnellsten. Der Docker-Container stellt sowohl einen HTTP- als auch einen HTTPS-Zugang mit selbst signiertem Zertifikat zur Verfügung. Beim ersten Webzugriff nach der Installation legt das Tool einen Admin-Benutzer ohne Passwort an. Es verteilt Kacheln mit den Links zu den jeweiligen Diensten gleichmäßig im Dashboard, am unteren rechten Fensterrand finden Sie die Einstellungsmöglichkeiten. Über den Konfigurationspunkt *Applikationsliste* legen Sie neue URLs an. Ein Dienst lässt sich wahlweise als Webseite oder Applikationstyp anlegen. Das Tool kennt viele Anwendungen wie Adguard, Kodi, Immich oder AVM und hinterlegt direkt das passende Symbol in der Dashboard-Kachel. Sie müssen nur noch die URL eintragen und eine Beschreibung hinterlegen. Für Anwendungen verwenden Sie optional ein eigenes Symbol, bei Webseiten ist das zwingend. Standardmäßig hängt Heimdall die Kachel an das Dashboard. Das lässt sich bei Bedarf unterbinden, sodass der Dienst lediglich in der Applikationsübersicht erscheint. Es besteht die Möglichkeit, weitere Benutzer anzulegen, von denen jeder sein eigenes Dashboard erhält. Die Vergabe eines Passworts und die Eingabe bei der Anmeldung unterstützt die App nur mit einer HTTPS-Verbindung. Platzbedingt können wir hier nicht alle Funktionen des etablierten Web-Dashboards vorstellen, ein eigener Blick darauf lohnt sich.

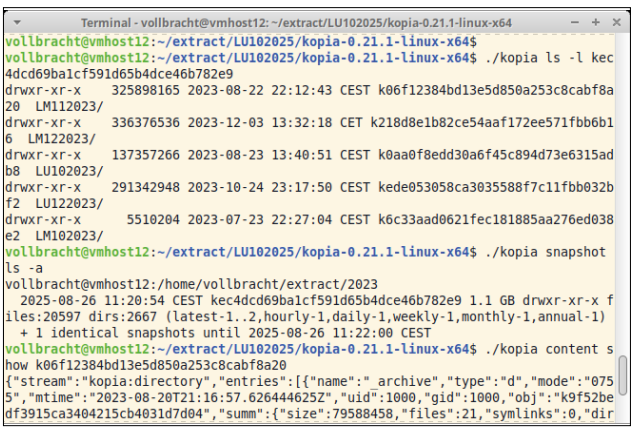


Lizenz: MIT
Quelle: https://github.com/linuxserver/Heimdall

Datenbunker

Mit Kopia 0.21.1 erstellen Sie im Handumdrehen Backups Ihrer Daten und stellen sie ebenso einfach wieder her.

Kopia verspricht ein schnelles und sicheres Backup und Restore. Auf Github stellt das Projekt diverse Binärpakete bereit. Die App sichert die Daten wahlweise unterhalb eines lokalen Einhängepunkts oder im Netz. Dabei unterstützt Kopia nicht nur die gängigen Cloud-Anbieter wie Google, AWS oder Azure, sondern bedient dank Rclone-Unterstützung auch eigene Netzwerkspeicher. Darüber hinaus unterstützt es WebDAV und SFTP. Sie können mit Kopia sogar einen eigenen Repository-Server betreiben, auf den das Tool die Daten TLS-verschlüsselt überträgt. Vor einem Backup legen Sie mit dem Unterbefehl `repository create` einen Speicherplatz für das Repository fest. Während des Einrichtens vergeben Sie das Passwort für den Zugriff auf das Repo und für die Datenverschlüsselung.



Wie lange Kopia gesicherte Dateien vorhält, legt es über eigene Standard-Retentions fest. Die passen Sie bei Bedarf mit `policy set` an. Mit `snapshot create`, gefolgt vom zu sichernden Verzeichnis, legen Sie die initiale Sicherung an. Je nach Größe dauert das einige Minuten. Das Tool hält Sie derweil mit Statusausgaben auf dem Laufenden. Rufen Sie den Befehl erneut auf, erzeugt Kopia nur eine inkrementelle Sicherung. Eine Übersicht aller vorhandenen Backups liefert `snapshot list`. Neben dem Inhalt zeigt die App für jedes Objekt eine individuelle Hash-ID an. Mit ihr können Sie gezielt auf ein Objekt zugreifen. Der Unterbefehl `ls` zeigt den Inhalt eines Snapshots an. Mithilfe der Erweiterung `mount` binden Sie einzelne Snapshots ein und greifen direkt auf deren Inhalte zu. Eine ausführliche Dokumentation ist auf der Github-Seite verlinkt. (Uwe Vo llbracht/*tle*)

Lizenz: Apache 2.0
Quelle: https://github.com/kopia/kopia

LINUXUSER

IHRE DIGITALE AUSGABE ÜBERALL DABEI!

LinuxUser begleitet Sie jetzt überall hin – egal, ob auf dem Tablet, dem Smartphone, dem Kindle Fire oder im Webbrowser. LinuxUser ist ab sofort immer dabei!

Available on the App Store

JETZT BEI Google Play

amazon apps Available on kindle fire

1x im Shop registrieren – überall mobil lesen.

Mit Ihren Login-Daten erhalten Sie überall Zugriff auf Ihre gekauften Digital-Ausgaben, im Shop-Account, in der Kiosk-Computec-App und auf epaper.computec.de.

shop.linuxuser.de

Paperless-NGX: Das papierlose Büro für kleine und mittlere Unternehmen

Ende des Papierkriegs

Das leistungsfähige Dokumentenmanagementsystem Paperless-NGX glänzt besonders in kleineren Büros als praxisnahe, kostengünstige Alternative zu Papierablagen oder teuren proprietären Systemen.

Dr.-Ing. Fabian Deitelhoff

README

Dokumentenchaos im Büroalltag ist keine Seltenheit, besonders in kleinen und mittleren Unternehmen. Die Anforderungen an Datenschutz, Aufbewahrung und Zugriff steigen stetig. Paperless-NGX verspricht eine einfache, sichere und lokal gehostete Lösung – ohne Cloud-Abhängigkeit, aber mit voller Power.

Als Open-Source-Dokumentenmanagement-System (DMS) wandelt Paperless-NGX Papierdokumente in ein durchsuchbares digitales Archiv um. Gerade für kleine und mittlere Unternehmen (KMU) bietet die Lösung große Vorteile: Wichtige Unterlagen gehen nicht mehr in Aktenordnern verloren. Dank Volltextsuche sind Informationen innerhalb von Sekunden auffindbar.

Ein Dokumentenmanagement spielt für KMU eine wichtige Rolle, um Compliance-Vorgaben im Kontext von DSGVO und GoBD einzuhalten und eine revisionssichere Archivierung zu gewährleisten. Paperless-NGX adressiert solche Anforderungen, ohne auf cloudbasierte Drittanbieter angewiesen zu sein. Alle Daten bleiben in der eigenen Kontrolle und liegen lokal auf dem Server.

Von Paperless über NG bis hin zu NGX

Die Software blickt auf eine ungewöhnliche Historie zurück. Ursprünglich als schlichtes Projekt Paperless gestartet, entstand daraus durch Jonas Winkler der populäre Fork Paperless-ng, als Abkürzung für Next Generation, als offizieller Nachfolger. Nachdem der Hauptentwickler sich zurückgezogen hatte, formierte sich Anfang 2022 eine Community, um das Projekt unter dem Namen Paperless-NGX weiterzuführen. Dieses Community-getriebene Modell zeigt beeindruckende Ergebnisse.

Allein die erste Paperless-NGX-Version vereinte Beiträge von über 50 Entwicklern und mehr als 250 Commits, inklusive neuer Features, Fehlerbehebungen und Sicherheitsupdates. Paperless-NGX verteilt die Verantwortung auf mehrere Schultern und veröffentlichte seither in enger Taktung zahlreiche Releases. Momentan, zum Stand 2025, befindet sich Paperless-NGX in der stabilen Versionsreihe 2.x und erfährt regelmäßige Aktualisierungen mit Fokus auf Performance, Sicherheit und Funktionalität.

Ein Alleinstellungsmerkmal ist die eingebaute Unterstützung für mehrere Benutzer und Rollen. Das System verfügt nun über ein robustes Berechtigungskonzept, das sowohl globale Zugriffsrechte als auch freigegebene Zugriffe für einzelne Dokumente oder Objekte erlaubt. So können mehrere Abteilungen oder Mitarbeitergruppen in einem System arbeiten, ohne dass jeder alles sieht.

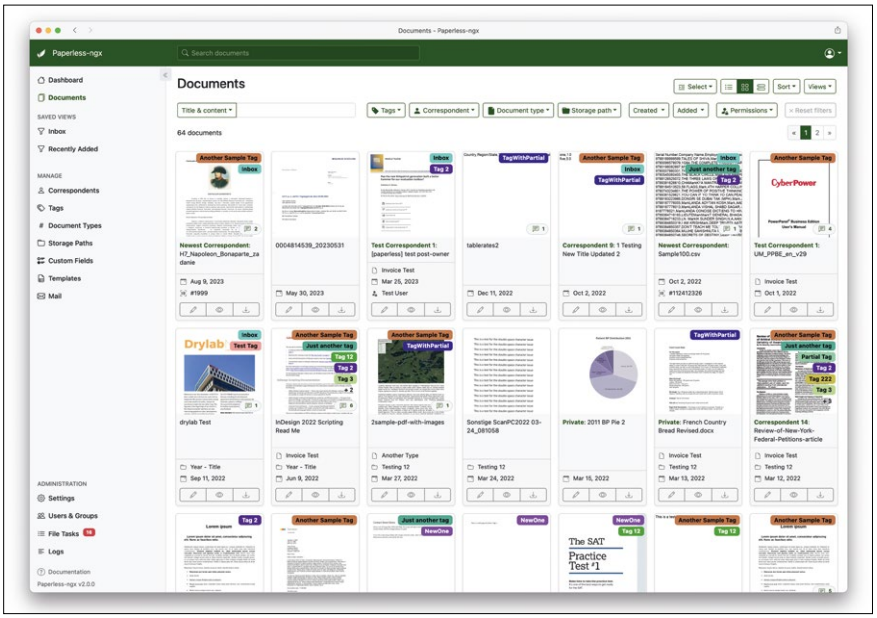
Auch bei der Texterkennung (OCR) und Klassifizierung gab es Fortschritte. Paperless-NGX wandelt gescannte Dokumente automatisch in durchsuchbaren Text um und verwendet dafür OCRmyPDF sowie die Tesseract-Engine. Die Anwendung

unterstützt dabei über 100 Sprachen. Deutsch, Englisch, Französisch und weitere Sprachen sind bereits vorinstalliert. Hervorzuheben ist außerdem der Einsatz von Machine Learning. Das System verwendet lernende Algorithmen, um neu importierten Dokumenten automatisch Schlagwörter (Tags), den Absender (Korrespondent) sowie Dokumententypen zuzuordnen. Diese automatische Klassifizierung – die Einstellung des Matching-Algorithmus auf Auto – wird mit jeder Nutzer-Korrektur besser und erkennt beispielsweise wiederkehrende Rechnungen oder Verträge immer treffsicherer.

Die aktuelle Version optimiert auch das Zusammenspiel im Mehrbenutzerbetrieb. Zum Beispiel dürfen nur noch Superuser weitere Admin-Konten anlegen. Überdies führt die aktuelle Version eine versionierte REST-API ein, sodass Integrationen auf eine stabile Schnittstelle setzen können. Feinschliffe wie hervorgehobene Treffer bei der Volltextsuche im Dokumenten-Viewer oder Performance-Verbesserungen bei paralleler Verarbeitung setzten die Entwickler jüngst ebenfalls um.

Im Unternehmensumfeld spielt die sichere Zugangskontrolle eine große Rolle, und auch hier hat Paperless-NGX aufgeholt. Neben der bereits erwähnten feingliedrigen Rechtevergabe pro User/Dokument gestattet die Software heute die Anbindung an bestehende Verzeichnisdienste und Single-Sign-on-Lösungen. So lässt sich beispielsweise ein zentraler LDAP- oder OpenID-Connect SSO-Dienst heranziehen, um Benutzer in Paperless-NGX anzulegen und deren Gruppen automatisch zu synchronisieren.

Alternativ lässt sich ein vorgeschalteter Reverse Proxy mit Header-Authentifizierung einsetzen, etwa in Kombination mit Authelia oder Authentik, um die Anmeldung mit Firmenzugängen zu ermöglichen. Für zusätzliche Sicherheit sorgen Features wie der Support von 2-Faktor-Authentifizierung (MFA). Passende Felder für OTP-Codes sind im aktuellen UI vorhanden sowie detailreiche Protokollierungen aller Benutzeraktionen. Nicht zuletzt erfüllt Paperless-NGX wichtige Kriterien für die revisionssichere Archivierung: Dokumente werden intern stets im PDF/A-Format für die Langzeitarchivierung gespeichert und unverändert neben den Originaldateien abgelegt. Dadurch und durch regelmäßige Backups lässt sich eine prüfungssichere Ablage gemäß gesetzlichen Vorgaben erreichen.



© Paperless-NGX

1 Die Dokumentenansicht zeigt unterschiedliche Informationen an: Neben farbcodierten Tags sehen Sie auch Korrespondenten und Dokumenttypen.

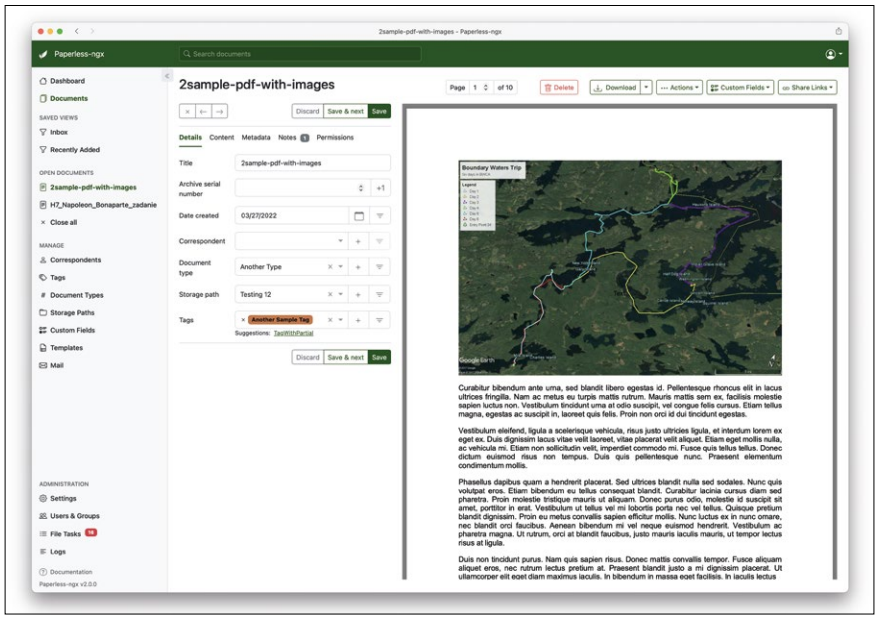
Typische Einsatzszenarien in Unternehmen

Eine Kernaufgabe von Paperless-NGX 1 besteht in der digitalen Archivierung von Unternehmensdokumenten aller Art. Im Alltag entstehen in Betrieben unzählige Unterlagen wie Verträge, Personalakten, Schriftverkehr und technische Dokumentationen, die häufig noch in Papierform vorliegen. Mit Paperless-NGX lassen sich diese Dokumente scannen, zentral ablegen und gemäß Aufbewahrungsfristen archivieren. Die Software indexiert Inhalte automatisch per OCR, sodass später eine Volltextsuche funktioniert.

Das Speichern als PDF/A und die Datenhaltung im eigenen Systemen stellen sicher, dass Dokumente langfristig lesbar bleiben und keine unbemerkten Änderungen erfahren. Für Prüfungen, etwa im

Listing 1: Docker installieren

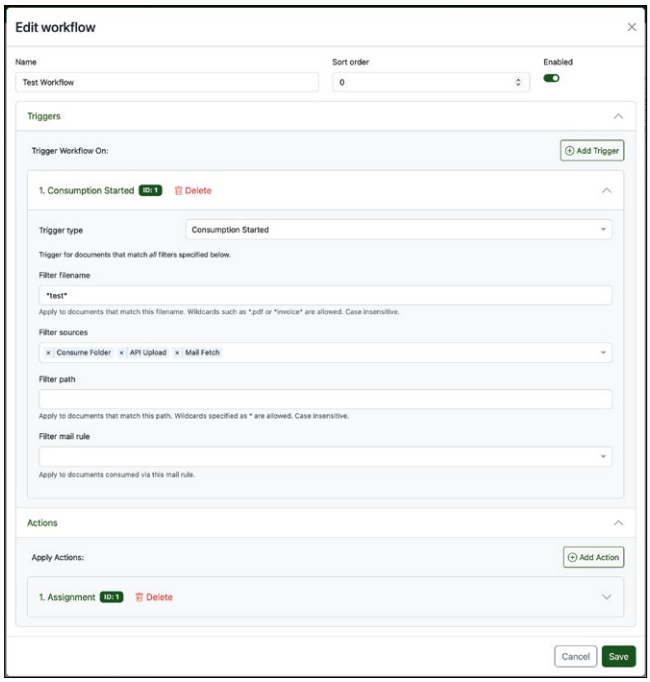
```
01 ### Docker Engine installieren (falls noch nicht geschehen)
02 $ sudo apt-get install -y docker.io
03 ### Docker-Compose-Plugin installieren
04 $ sudo apt-get install -y docker-compose-plugin
```



© Paperless-NGX

2 Die Ansicht zum Editieren von Dokumenten ist zweigeteilt: Links finden sich die Metadaten, rechts erscheint das betreffende Dokument.

Rechnungswesen, kann eine solche Lösung die revisionssichere Ablage unterstützen, indem alle Dokumente unveränderlich gespeichert und Änderungen versioniert protokolliert sind. Häufiges Anwendungsszenario ist die Verarbeitung von Eingangsrechnungen und Belegen. Mit Paperless-NGX sammeln Sie per Post oder E-Mail eingegangene Rechnungen zentral und verarbeiten sie digital weiter 2. Papierrechnungen lassen sich mittels Dokumentenscanner einlesen und digitale Rechnungen als PDF-Dokumente importieren. Paperless-NGX erkennt dabei automatisch Metadaten wie Rechnungsnummer, Datum, Betrag oder Absender und speichert diese als durchsuchbare Tags oder Felder.



© Paperless-NGX

3 Die Workflows gestatten eine genauere Steuerung der Dokumentenpipeline, inklusive der Möglichkeit, Aktionen auszulösen.

Über vordefinierte Regeln richten Sie Workflows 3 wie interne Prüfprozesse ein. Die Integration in die Buchhaltung oder ein ERP-System fällt über die API denkbar leicht. Ein Buchhaltungssystem kann via REST-API eine Rechnung samt Buchungsinformationen aus Paperless abfragen oder umgekehrt neue Belege in Paperless-NGX einstellen. Auch wiederkehrende Rechnungen wie Miete und Strom identifiziert das System durch Machine Learning nach einiger Zeit automatisch und ordnet sie passenden Kategorien oder Lieferanten zu, was die Arbeit der Buchhaltung weiter erleichtert.

Neben Rechnungen lassen sich auch alle anderen eingehenden Dokumente im Unternehmen digitalisieren und mit Paperless-NGX verarbeiten. Ein klassisches Beispiel ist die gescannte Briefpost: Kundenanschreiben oder behördliche Briefe lassen sich zentral von der Verwaltung öffnen, scannen und in Paperless-NGX importieren. Das System verteilt die Dokumente virtuell an die Zuständigen, indem es etwa anhand erkannter Schlagwörter oder Absender die richtigen Tags wie Kundenanfrage oder Behörde vergibt und Benachrichtigungen auslöst. Dank der E-Mail-Importfunktion 4 von Paperless-NGX lässt sich dieser Vorgang automatisieren. Das funktioniert beispielsweise über ein spezielles Postfach wie eingang@firma.de, an das Scans oder digitale Briefe geschickt werden.

Systemarchitektur und technische Einbettung

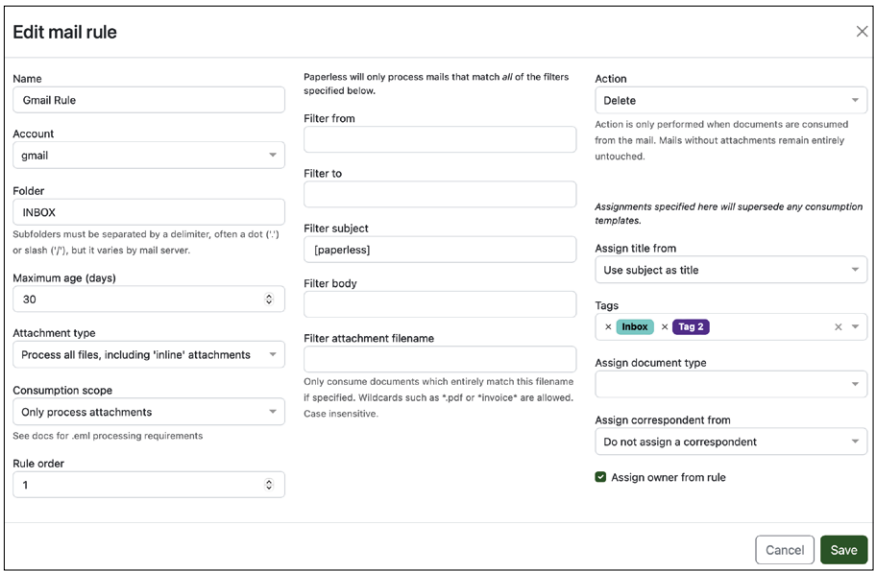
Unter der Haube ist Paperless-NGX eine klassische Webanwendung mit modularen Komponenten. Die Backend-Serverkomponente basiert auf Python/Django und stellt die Geschäftslogik sowie eine REST-API bereit. Im Frontend kommt ein in TypeScript und Angular geschriebenes Webinterface zum Einsatz. Für die Datenbank nutzt Paperless-NGX bevorzugt PostgreSQL und alternativ MariaDB zur Speicherung aller Metadaten, Indizes, Benutzer und weiterer Daten.

Zusätzlich koordiniert ein Redis-Server als Message-Broker die Hintergrundjobs, darunter asynchrone OCR-Verarbeitung und E-Mail-Import. Diese Architektur ermöglicht es Paperless-NGX, rechenintensive Aufgaben in Hintergrund-Worker

auszulagern und mehrere Dokumente parallel zu verarbeiten. Auf einem Multicore-System können zum Beispiel mehrere OCR-Vorgänge parallel laufen, was die Verarbeitung deutlich beschleunigt.

Die Dokumentendateien selbst, wie PDFs, Bilder und Office-Dokumente, legt das Paperless-Backend im Dateisystem ab, in einen von Paperless verwalteten Ordner mit konfigurierbarer Struktur. Für erweiterte Dateitypen greift Paperless-NGX auf zusätzliche Dienste zurück. So kann ein angebundenes Apache-Tika-Modul Inhalte aus Office-Dokumenten oder E-Mails extrahieren, während der Gotenberg-Service für die Konvertierung von Word oder Excel in PDF zuständig ist. Die beiden Dienste werden optional genutzt, um eine breite Palette an Dokumentformaten verarbeiten zu können. In einer typischen Installation laufen daher mehrere Container oder Prozesse: der Paperless-Webserver, inklusive Backend und UI, der Datenbankserver, der Redis-Broker sowie gegebenenfalls Tika und Gotenberg zum Konvertieren. Das komponentenbasierte Design lässt sich skalieren und relativ einfach einrichten.

Zudem ist Paperless-NGX offen konzipiert. Sie binden es per REST-API nahtlos in die bestehende IT ein. Die API erlaubt Funktionen wie Dokumenten-Upload, Suche, Tagging und Abruf, unterstützt diverse Authentifizierungsverfahren und bietet ein Webinterface für Entwickler. Webhooks versprechen Automatisierungen wie Benachrichtigungen in Chat-Systeme bei neuen Dokumenten. Die offene Architektur gestattet Integrationen mit Systemen wie ERP, CRM oder Nextcloud. Auch mobile Apps für Android und iOS nutzen die API, um Dokumente unterwegs zu verwalten. Damit eignet sich Paperless-NGX für den Einsatz in komplexen Unternehmensumgebungen.



© Paperless-NGX

4 E-Mail-Regeln unterstützen verschiedene Filter und Aktionen für eingehende E-Mails.

Paperless-NGX mit Docker bei Hetzner

Paperless-NGX läuft auf moderater Hardware. Vor allem CPU-Leistung und Speicher sind für die OCR-Verarbeitung wichtig. Als guter Startpunkt dient für kleine Unternehmen ein Cloud-Server wie die von Hetzner, konkret vom Typ CX22 oder größer. Im Standard verbirgt sich dahinter eine VM mit zwei vCPU, 4 GByte RAM, 40 GByte NVMe-SSD. Diese Konfiguration bringt genug Arbeitsspeicher mit, um außer dem Betriebssystem auch PostgreSQL und Tesseract flüssig zu betreiben. Die beiden virtuellen CPUs reichen für parallele OCR-Jobs aus.

Bei wachsendem Dokumentenvolumen oder vielen gleichzeitigen Nutzern können Sie auf eine höhere Stufe mit beispielsweise vier CPUs und 8 GByte RAM aufrüsten. Docker erleichtert hier einen späteren Umzug, da Container relativ einfach zu migrieren sind. Wichtig ist ausreichender Speicherplatz: 40GByte könnten zwar anfangs genügen, aber je nach Scan-Aufkommen sollten Sie lieber großzügig planen oder ein Storage-Volume anbinden, vor allem wenn zahlreiche hochauflösende PDFs anfallen. Die meisten Cloud-Anbieter erlauben das einfache Hochstufen von Speicher und rechnen nach Gigabytes ab.

Bevor Sie Paperless-NGX aufsetzen, sollten grundlegende Sicherheits- und Konfigurationsmaßnahmen am Server erledigt sein. Als Betriebssystem empfiehlt sich eine aktuelle, langzeitunterstützte Distribution wie Ubuntu Server LTS oder Debian. Nach dem Starten der Cloud-VM richten Sie zuerst einen Nicht-Root-Benutzer mit SSH-Zugang ein und härten die SSH-Login-Parameter. Dazu gehört es zum Beispiel, den Passwort-Login zugunsten von Schlüsselauthentifizierung zu deaktivieren.

Eine initiale Systemaktualisierung mittels `apt update && sudo apt upgrade` gewährleistet, dass alle Pakete auf dem neuesten Stand sind. Danach ist die Firewall an der Reihe, entweder die Cloud-seitige Firewall von Hetzner oder lokale Tools wie UFW. Wichtig ist dabei, dass eingehende Verbindungen nur auf die nötigen Ports erlaubt sind: typischerweise Port 22 (SSH) für Admin-Zugang, Port 80/443 für Web (HTTP/HTTPS) und eventuell intern genutzte Ports wie 8000, falls ein Test von Paperless zunächst ohne Reverse Proxy erfolgt.

Nun geht es an das Installieren von Docker. Unter Ubuntu geschieht das via offiziellem Docker-APT-Repository oder einfach per Convenience-Skript. Die Kurzform zeigt Listing 1.

Für Paperless-NGX ist eine Docker-Compose-Konfiguration erforderlich, die sämtliche benötigten Container definiert. Beispiele für Docker-Compose-Dateien finden Sie im Repository von Paperless-NGX. Eine exemplarische `docker-compose.yml` sehen Sie in Listing 2.

Die Konfiguration definiert die Container: `broker` (Redis), `db` (PostgreSQL), `webserver` (Paperless-NGX-App), `gotenberg` und `tika`. Über benannte Volumes wie `data`, `media` etc. ist die Persistenz sichergestellt. Selbst wenn Container neu erstellt werden, bleiben Daten und Dokumente erhalten, da sie außerhalb des Containers im Volume liegen.

Einige Pfade wie `./export` und `./consume` (Listing 2, Zeile 33 und 35) sind hier als lokale Bind-Mounts konfiguriert. Das gestattet, die Ordner auf dem Host zu nutzen, etwa für einfache Dateiübergabe oder Zugriff via SMB/NFS im Netzwerk. Die meisten Konfigurationsoptionen steuern Sie über Umgebungsvariablen. Der Übersicht halber lagern Sie sie in eine separate Datei aus, in Listing 2 `paperless.conf` (Zeile 37) genannt.

Wichtig sind die Verbindungsparameter zur Datenbank und Redis – Paperless-NGX erwartet sie per `PAPERLESS_DBHOST`, `PAPERLESS_REDIS` (Zeile 39) etc. – sowie Einstellungen zur Zeitzone und optional OCR-Sprache. Hier ist `PAPERLESS_OCR_LANGUAGES: deu+eng` (Zeile 46) gesetzt, sodass deutsche und englische Texte erkannt werden. Standardmäßig berücksichtigt Paperless-NGX nur Englisch, weitere Sprachen müssen angegeben und im Image vorhanden sein.

Die Integration von Tika und Gotenberg wurden durch `PAPERLESS_TIKA_ENABLED: 1` (Zeile 47) und die Endpunkte aktiviert. Damit kann Paperless-NGX zum Beispiel Office-Dateien verarbeiten.

Listing 2: docker-compose.yml

```
01 version: "3.4"
02 services:
03  broker:
04   image: redis:7-alpine
05   restart: unless-stopped
06   volumes:
07    - redisdata:/data
08  db:
09   image: postgres:15-alpine
10   restart: unless-stopped
11   volumes:
12    - pgdata:/var/lib/postgresql/data
13   environment:
14    POSTGRES_DB: paperless
15    POSTGRES_USER: paperless
16    POSTGRES_PASSWORD: SicheresPasswort
17  webserver:
18   image: ghcr.io/paperless-ngx/paperless-
   ngx:latest
19   restart: unless-stopped
20   depends_on:
21    - db
22    - broker
23    - gotenberg
24    - tika
25   ports:
26    - "8000:8000"
27   volumes:
28    # Persistente Daten (Originale/PDFs)
29    - data:/usr/src/paperless/data
30    # Index und Vorschau-Dateien
31    - media:/usr/src/paperless/media
32    # Export-Ordner (falls genutzt)
33    - ./export:/usr/src/paperless/export
34    # Import-Ordner (Watch-Folder)
35    - ./consume:/usr/src/paperless/consume
36   # Ausgelagerte Konfiguration
37   env_file: paperless.conf
38   environment:
39    PAPERLESS_REDIS: redis://broker:6379
40    PAPERLESS_DBHOST: db
41    PAPERLESS_DBNAME: paperless
42    PAPERLESS_DBUSER: paperless
43    PAPERLESS_DBPASS: SicheresPasswort
44    PAPERLESS_TIME_ZONE: Europe/Berlin
45    # OCR-Sprachen (Deutsch+Englisch)
46    PAPERLESS_OCR_LANGUAGES: deu+eng
47    PAPERLESS_TIKA_ENABLED: 1
48    PAPERLESS_TIKA_ENDPOINT: http://tika:9998
49    PAPERLESS_TIKA_GOTENBERG_ENDPOINT:
   http://gotenberg:3000
50    # ... weitere Settings ...
51  gotenberg:
52   image: gotenberg/gotenberg:7.10
53   restart: unless-stopped
54   # optional anpassen, z.B. Javascript
55   # in Chromiumde aktivieren etc.
56  tika:
57   image: ghcr.io/paperless-ngx/tika:latest
58   restart: unless-stopped
59 volumes:
60  data:
61  media:
62  pgdata:
63  redisdata:
```

Benötigen Sie das nicht, verzichten Sie auf beide Dienste inklusive der Variablen, um Ressourcen zu sparen.

Nachdem die YAML-Datei angelegt, mit realen Passwörtern versehen und angepasst ist, starten Sie das Setup mit `docker compose up -d` im selben Verzeichnis. Docker lädt die Images herunter und fährt alle Container im Hintergrund (–d) hoch. Beim ersten Start initialisiert Paperless-NGX seine Datenbanktabellen. Mit dem Kommando aus der ersten Zeile von Listing 3 können Sie die Logs beobachten und auf die Meldung warten, dass der Server auf Port 8000 lauscht.

Beim allerersten Start ist es sinnvoll, einen Admin-Benutzer anzulegen. Das erfolgt entweder interaktiv via Befehl (Listing 3, zweite Zeile), oder Sie setzen zuvor die Umgebungsvariablen `PAPERLESS_ADMIN_USER` und `PAPERLESS_ADMIN_PASSWORD`. In der Compose-Datei aus Listing 2 könnten Sie also unter `environment:` die Zeilen aus Listing 4 ergänzen.

Paperless-NGX würde beim ersten Start automatisch diesen Account erzeugen. Alternativ erledigt dies das Installationsskript. Sie sollten sicherstellen, auf die Weboberfläche Zugriff mit Admin-Rechten zu haben, um danach im UI die nötigen Einstellungen vorzunehmen.

Sobald der Webserver gestartet ist, rufen Sie die IP des Servers mit Port 8000 im Browser auf. Dabei sollte die Login-Seite von Paperless-NGX erscheinen. Nach dem Einloggen mit dem Admin-Account empfiehlt es sich, Dokumente zum Prüfen der Grundfunktionen hochzuladen. Ziehen Sie per Drag & Drop ein PDF in die Oberfläche oder wählen Sie im Menü unter *Dokumente* | *Upload* eine Datei aus. Paperless-NGX wird die Datei in den Consume-Ordner legen, per OCR verarbeiten, und wenige Sekunden später sollte das Dokument in der Liste zu sehen sein. Falls bei der Compose-Datei der Port 8000 nicht nach außen geöffnet ist, ist ein Test mit `ssh -L 8000:localhost:8000 user@server` möglich, um lokal via Tunnel auf den Port zuzugreifen.

Listing 3: Logs und Admin

```
$ docker compose logs -f webserver
$ docker compose exec webserver python manage.py createsuperuser
```

Reverse Proxy und HTTPS in der Produktivumgebung

Im Standard-Setup lauscht Paperless-NGX intern auf Port 8000. Für den sicheren öffentlichen Betrieb empfiehlt sich ein Reverse Proxy wie Nginx oder Traefik. Die Vorteile sind TLS-Verschlüsselung (etwa via Let's Encrypt), eine nutzerfreundliche Domain (dms.meinefirma.de) sowie Funktionen wie Load Balancing. Besonders wichtig ist HTTPS, um vertrauliche Dokumente zu schützen. Nginx kann direkt auf dem Host oder als Docker-Container laufen. Traefik übernimmt automatisch das Zertifikatsmanagement.

Ist der Reverse Proxy aufgesetzt, sollte die Anwendung unter https://paperless.firma.de erreichbar sein, mit gültigem Zertifikat über den Standard-Port 443. In der Praxis sollten Sie Zugriffe auf die Paperless-Oberfläche auf HTTPS beschränken. Das aktuelle Setup leitet daher alle http-Anfragen auf 443 um. So sind die Datenübertragungen, inklusive Benutzer-Login und Dokumentendownloads, durchgängig verschlüsselt.

Wer die manuellen Schritte zum Aufsetzen eines Reverse Proxy scheut, kann direkt in der Docker-Compose mit einem Traefik-Container arbeiten. Traefik kann als Reverse Proxy fungieren und unterstützt Let's Encrypt automatisch. Das erweist sich als nützlich, wenn mehrere selbst gehostete Dienste auf einer Maschine in Betrieb sind, die alle über verschiedene Subdomains laufen. In unserem Szenario genügt allerdings Nginx, da wir uns auf einen Dienst konzentrieren. Wichtig ist am Ende nur: HTTPS und Domain sollten eingerichtet sein, bevor der Echtbetrieb startet, um Sicherheit und Nutzerkomfort, keine Portnummer und gültiges Zertifikat zu gewährleisten.

Fazit: einsatzbereit und zukunftssicher

Paperless-NGX zeigt eindrucksvoll, wie eine Open-Source-Community ein leistungsfähiges Dokumentenmanagementsystem zur Marktreife bringen kann. Für kleine und mittlere Unternehmen glänzt es als praxisnahe, kostengünstige Alternative zu Papierablagen oder teuren proprietären Systemen. Die Vorteile dabei bestechen deutlich: schnelle Volltextsuche, ortsunabhängiger Zugriff, digitale Workflows und volle Datenhoheit auf dem eigenen Server. Hinzu kommt ein Plus für Datenschutz und Compliance.

Technisch ist Paperless-NGX ausgereift und flexibel. Docker-basierte Installation, moderne Weboberfläche, REST-API und Machine Learning machen das System einfach zu nutzen und gut zu integrieren. Es lässt sich auf günstiger Cloud-Infrastruktur betreiben und bei Bedarf skalieren. Funktionen wie SSO/LDAP sowie Schnittstellen zu Drittsystemen ermöglichen die nahtlose Einbettung in bestehende IT-Umgebungen.

Die aktive Entwickler-Community arbeitet kontinuierlich an neuen Features wie KI-gestützter Klassifikation, digitalen Signaturen, mobilen Optimierungen und workflowbasierten Freigaben. Dank offener Lizenz und reger Beteiligung bleibt das Projekt zukunftssicher. Für Unternehmen bedeutet das: ein sofort einsatzbereites, flexibles und modernes DMS mit hoher Alltagstauglichkeit. Schon ein kleiner Pilot, etwa mit Eingangsrechnungen, zeigt schnell den Nutzen. Weniger Suchaufwand, mehr Übersicht und begeisterte Nutzer – Paperless-NGX macht das papierlose Büro effizient, sicher und besonders geeignet für KMU. (*csi*)

Listing 4: Admin anlegen

```
PAPERLESS_ADMIN_USER: admin
PAPERLESS_ADMIN_PASSWORD: Geheim
```

Weitere Infos und interessante Links
www.linux-user.de/qr/52466

Dateien zum Artikel herunterladen unter
www.linux-user.de/dl/52466

OpenSuse Leap Micro 6.2 als Remote-Server für grafische Anwendungen

Nützliche Behälter

Eine Leap-Micro-Installation bringt auf Containerbasis bereitgestellte Anwendungen aus Ubuntu, Fedora oder dem OpenSuse Build Service ins Heimnetz. Peter Kreußel

README

Die MicroOS-Architektur, bei der Anwendungen abgeschottet vom Grundsystem in Containern ablaufen, hatte Suse einst zum Standard für OpenSuse Leap auserkoren. Davon übrig geblieben sind MicroOS auf Tumbleweed- und Leap Micro auf Leap-Basis. Letzteres erscheint parallel zu Leap 16.0 in der Version 6.2. Wir zeigen den Einsatz als praktischen Remote-Server.

Im Oktober 2025 macht nicht nur OpenSuse Leap den Major-Versionssprung auf Version 16. Auch Leap Micro 1 erscheint im selben Monat in Version 6.2 mit denselben Softwareversionsständen wie Leap 16. Leap Micro ist als reine Laufzeitumgebung für Container und virtuelle Maschinen ausgelegt 2. Software, die über dieses Einsatzgebiet hinausgeht, wie eine grafische Umgebung oder gar Desktop-Programme, fehlt. Das Konzept dieses Minimalsystems besteht darin, dass jegliche Nutzsoftware („Payload") isoliert vom Grundsystem in Containern oder virtuellen Maschinen abläuft.

Sowohl Container als auch virtuelle Maschinen isolieren die in ihnen laufende Software vom Grundsystem. Als zusätzlicher Schutz ist das Root-Dateisystem nur lesbar eingehängt. Aktualisierungen funktionieren auf Basis von Snapshots im – unter Suse schon lange üblichen – Dateisystem Btrfs. Das System spielt die Updates ganz ohne Zutun des Anwenders ein, der nächste Neustart aktiviert sie. Dabei prüft das Tool Health-

Checker, ob das System wieder den normalen Betriebszustand erreicht, und kehrt andernfalls zum letztbekannten funktionierenden Zustand zurück.

Die remote im Browser aufrufbare Benutzeroberfläche Cockpit 3 gestattet eine einfache, grafische Handhabung der Container, virtuellen Maschinen und des gesamten Systems im Browser. MicroOS ist für ein Remote-Management prädestiniert und kann dabei in weiten Bereichen sogar auf die Konsole verzichten.

Der von OpenSuse nicht deutlich kommunizierte Unterschied zwischen Leap Micro und OpenSuse Micro besteht in der Basis: hie Leap, da Tumbleweed. Erstere garantiert einen störungsfreien Betrieb mit seltenen Aktualisierungen beim schmalen System aus wenigen Paketen.

Insgesamt unterscheidet sich Leap Micro 6.2 bis auf erneuerte Versionen der eingesetzten Software wenig vom Vorgänger 6.1. Die auffälligste Änderung besteht in der Anpassung der Cockpit-Management-Konsole an die OpenSuse-Farben. Die wichtigste Neuerung ist aber der nun auf zwei Jahre aufgestockte Support-Zeitraum.

Heimvorteil

Die Stärken von Containern wie Distributionsunabhängigkeit, Kapselung, Effizienz sowie ein Versions- und Instanzenmanagement haben ihnen in der Unternehmens-IT einen festen Platz gesichert. Viele der dort geschätzten Vorteile erweisen sich im heimischen Umfeld auch als wertvoll. Zum Testen einer Software aus einem experimentellen Build-Service-Repository, das grundlegende Systembibliotheken auf instabile Versionen auffrischt, können Sie mit wenigen Klicks einen Container mit einem Leap- oder Tumbleweed-Testsystem einrichten 4.

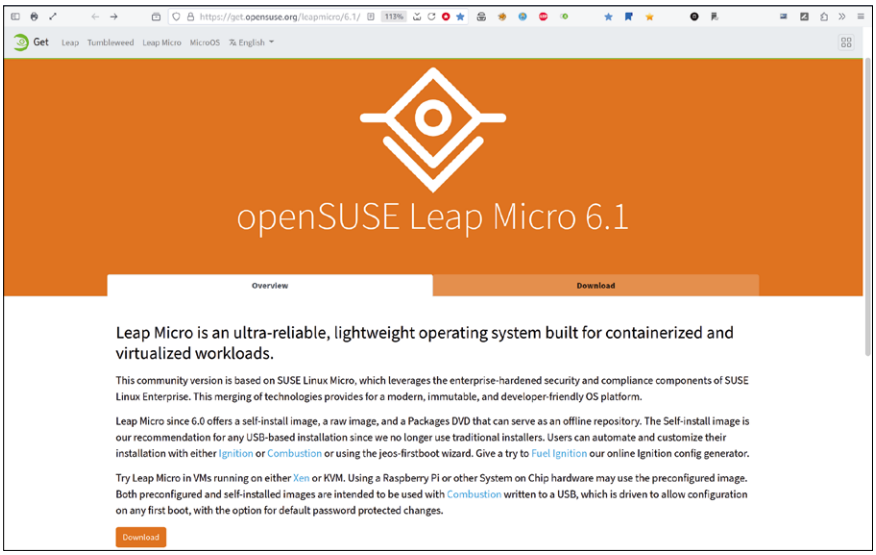
Vielleicht möchten Sie nur bestimmte Programme in der neueren Tumbleweed-Version nutzen oder peilen den Einsatz einer Software an, die es nur für Ubuntu gibt. Dann hilft es, ein Container-Image der gewünschten Distribution einzurichten, das praktisch nur den Paketmanager enthält. Für alle gängigen Linux-Derivate findet sich ein solches über die Suchfunktion von Cockpit.

Installieren Sie dann noch OpenSSH und Xauth (oder Waypipe, das Wayland-Pendant zum bekannten Remote-X.org) in den Container 5, lassen sich die dort vorgehaltenen Desktop-Programme mit grafischer Benutzeroberfläche im ganzen Heimnetz nutzen.

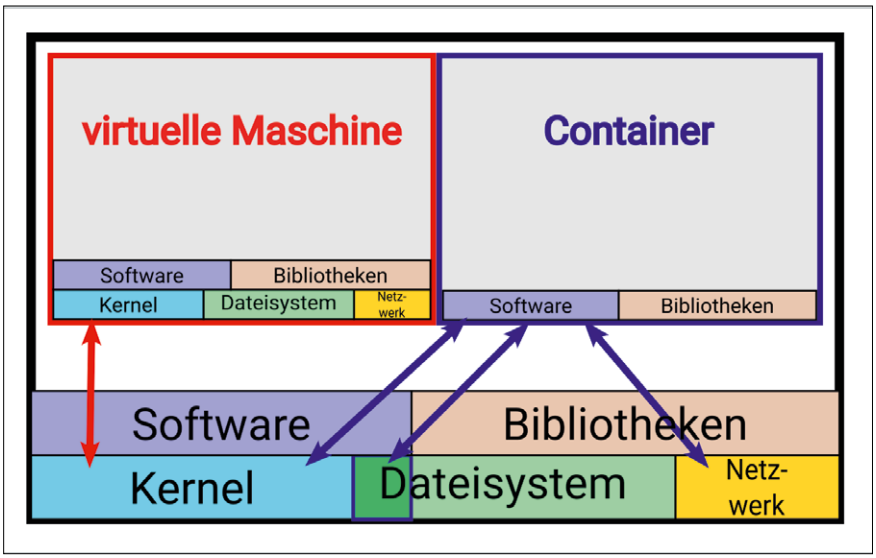
OpenSuse Leap Micro 6.2 Self-Install auf Heft-DVD

Ausgelagert

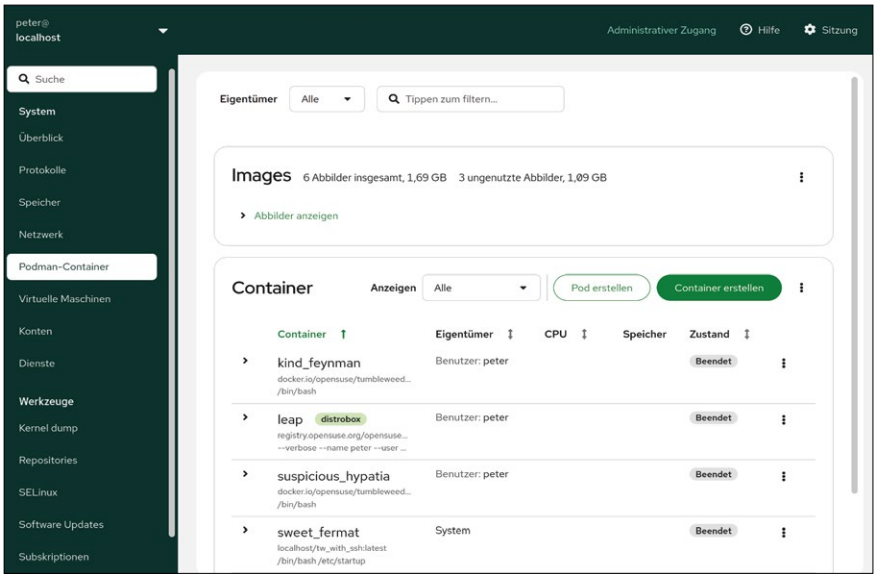
Haben Sie einen ausrangierten, halbwegs leistungsfähigen PC in Reserve, installieren Sie Leap Micro darauf, ohne Bildschirm und Tastatur. Erforderlich sind dann zwei USB-Sticks: Auf den ersten,



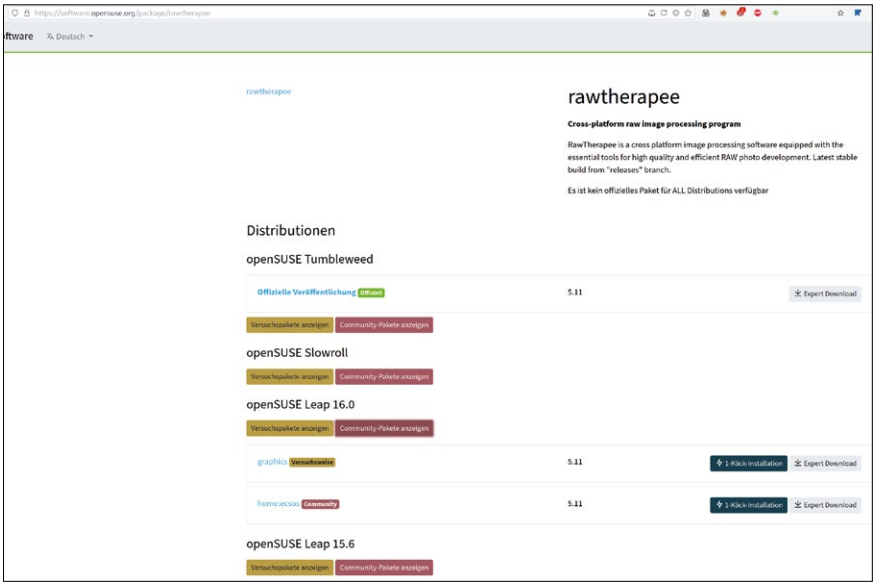
1 Die Produktseite von Leap Micro, bei Redaktionsschluss noch auf Versionsstand 6.1, preist diese OpenSuse-Spielart als besonders zuverlässig und leichtgewichtig an.



2 Während virtuelle Maschinen das gesamte Linux-System duplizieren, greift containerisierte Software auf die Systemressourcen des Hosts zurück.



3 Die Containerverwaltung ist das wichtigste Modul von Cockpit unter Leap Micro, doch das webbasierte Frontend administriert noch weitere Aspekte des Systems.



4 Community- oder Testpakete liefern oft aktuellere oder fehlende Pakete für Leap, wie hier die Fotobearbeitung Rawtherapee. Der Test per Container klappt gefahrlos.

von dem der neue Server booten muss, schreiben Sie per Suse-Image-Writer das Self-Install-ISO-Image für MicroOS.

Auf den zweiten Stick übertragen Sie auf demselben Weg das von einem von OpenSuse online bereitgestellten Konfigurationsgenerator erzeugte Konfigurations-Image. Das laden Sie über den Button unterhalb von *Convert to Ignition-Combustion-Ready Filesystem IMG in the Browser* herunter. Den Quellcode des von OpenSuse herausgegebenen Generators, der auch Passwörter verarbeiten kann, können Sie bei Interesse in seinem Github-Account einsehen.

Die vollautomatische Installation startet nach dem Einschalten des Rechners mit beiden eingesteckten Sticks nach mindestens zweimaligem Drücken der Eingabetaste. In vielen Fällen dürfte es aber die einfachere Lösung sein, Display und Tastatur anzuschließen. Findet MicroOS nach dem ersten Start keinen Konfigurationsdaten-USB-Stick, fragt ein Wizard die Daten am Bildschirm ab 6.

Nach wenigen Minuten steht Leap Micro zum Hinzufügen von Software bereit, die für den Anwender nützlich ist. Als Beispiel dafür dient hier ein leichtgewichtiger Tumbleweed-Container, in den sich mit dem Kommandozeilen-Tool Zypper beliebige GUI-Anwendungen aus den OpenSuse-Repos oder dem Build Service installieren lassen. Sie stehen remote per SSH über X.org oder dessen Wayland-Pendant Waypipe für jeden Rechner im Heimnetz zur Verfügung.

MicroOS enthält nur Leap-Pakete. Seine Container-Laufzeitumgebung lässt sich in identischer Form auch unter Leap (Listing 1, erste zwei Zeilen) und Tumbleweed installieren und die Libvirt-Verwaltungsumgebung für virtuelle Maschinen scharfschalten (letzte Zeile).

Benutzen Sie einen separaten MicroOS-Server, dann loggen Sie sich dort per SSH ein. Bei angeschlossenem Bildschirm sehen Sie die dafür benötigte IP-Adresse am Login-Prompt. Ansonsten lässt sie sich über den heimischen Router herausfinden. Aktivieren Sie nach der Anmeldung den Cockpit-Dienst mit `sudo systemctl enable cockpit.socket --now`.

Dann rufen Sie unter *https://Rechner-IP*:9090 die für die Administration zuständige Benutzeroberfläche von Cockpit auf. Eine Zertifikatswarnung im Browser lässt sich dabei nicht vermeiden, denn ein akzeptiertes Zertifikat müssten Sie kostenpflichtig bei einer dem Browser bekannten Zertifizierungsstelle bestellen oder per Let's Encrypt beziehen.Das ist jedoch bei Rechnern im Heimnetz nicht ohne Weiteres möglich.

Für das Cockpit-Login verwenden Sie ein systemweites Benutzerkonto, die Webkonsole läuft zunächst mit limitierten Rechten. Ein Klick auf *Turn on admi-*

Listing 1: Container-Runtime installieren

```
$ sudo zypper in -t pattern cockpit
$ sudo zypper in libvirt cockpit-podman
$ systemctl enable libvirtd.socket --now
```

nistrative access behebt das nach Eingabe des Root-Passworts für diese und alle folgenden Sitzungen.

Ladung aufnehmen

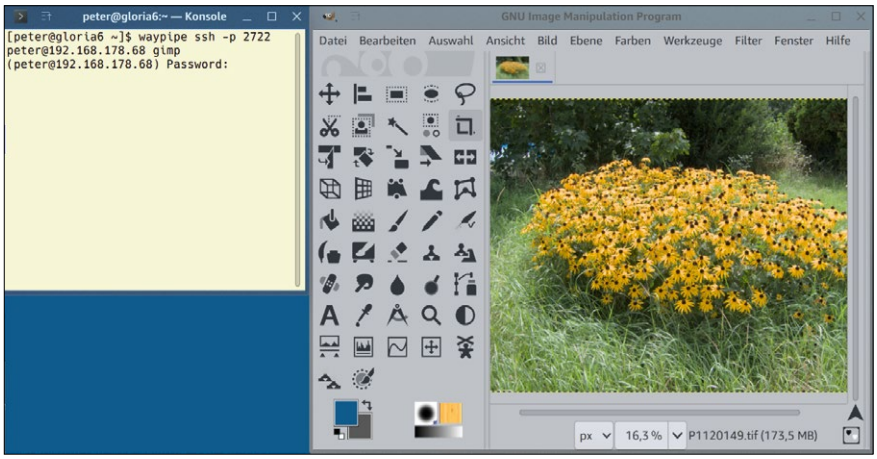
Öffnen Sie zum Einrichten des angesprochenen Tumbleweed-Containers den Menüpunkt *Podman-Container*. Images beziehen Sie über die Schaltfläche *Neues Abbild herunterladen* am rechten Rand der gleichnamigen Rubrik. Abbildung 7 zeigt den Vorgang für Leap.

In unserem Fall führt der Text „opensuse tumbleweed" für *Suchen nach* zum gewünschten *Official openSUSE Tumbleweed image* des Anbieters *opensuse*. Das Label (deutsche Übersetzung: *Etikett*) *latest* sorgt dafür, dass nur die neuesten Versionen als Suchergebnis unter *Abbilder anzeigen* erscheinen. Mit von der Partie ist der Button *Container erstellen*, der laufende Instanzen der im Image verpackten Software erstellt.

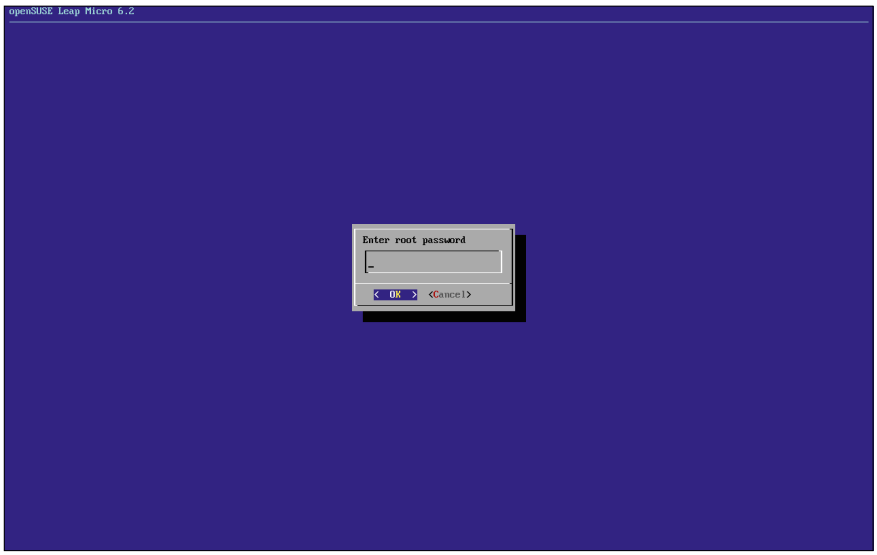
Ein Mausklick darauf öffnet einen Dialog, in dem Sie dem Container einen Namen geben. Klicken Sie als Nächstes auf *Erstellen und ausführen*. Daraufhin erscheint in der Rubrik *Container* ein Eintrag, der den *Zustand* als *Läuft* ausweist. Klappen Sie die zugehörige Zeile mit der Pfeiltaste nach rechts aus und öffnen Sie dann den Reiter *Konsole*. Das bringt innerhalb des Containers ein Root-Terminal zum Vorschein.

Hier installieren Sie einen SSH-Server (Listing 2, erste Zeile) und erstellen dafür einen Host-Schlüssel (zweite Zeile). Um später grafische Programme per SSH remote ausführen zu können, installieren Sie noch das Tool Xauth und – falls sie die entfernten Programme von einer Wayland-Session aus aufrufen möchten – Waypipe. Der automatische Start des SSH-Daemons erfordert ein Bash-Skript, zu dessen Anlegen Sie einen Konsolen-Editor wie Nano benötigen.

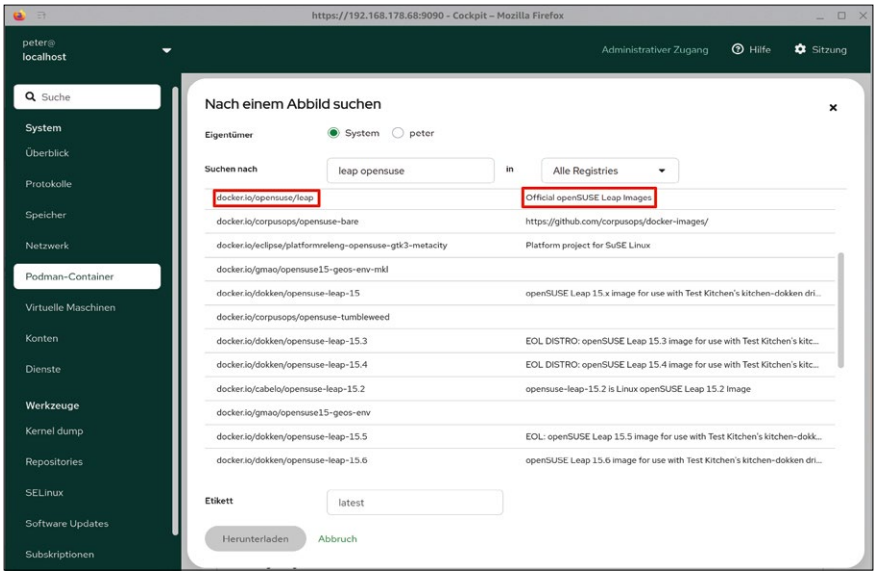
Sie installieren das Trio mit dem Kommando aus der dritten Zeile und legen dann mit Nano (letzte Zeile) das Startskript wie in Listing 3 zu sehen an. Es startet zum einen den SSH-Daemon, der ohne Aufrufparameter in den Hintergrund verschwindet, ohne das Skript zu blockieren, sowie zum anderen eine Log-in-Shell. Der Start der Shell am Ende stellt sicher, dass der Reiter *Konsole* weiter



5 Via Waypipe lässt sich das Grafikprogramm Gimp aus dem Container auf dem Leap-Micro-Server (IP-Adresse 192.168.178.68, Port 2722) remote verwenden.



6 Ist kein Ignition-Konfigurations-Stick am System angeschlossen, fragt der MicroOS-Installer die wenigen einzugebenden Daten am Bildschirm ab.



7 Die Suchfunktion nach Auswahl des Menüpunkts *Neues Abbild herunterladen* findet das offizielle Leap-Image direkt vom Anbieter OpenSuse.

funktioniert, wenn beim Containerstart der Aufruf dieses Skripts erfolgt.

Listing 2: SSH-Server einrichten

```
$ sudo zypper in openssh
$ sudo ssh-keygen -A
$ sudo zypper in xauth waypipe nano
$ sudo nano /etc/startup
```

Datentausch

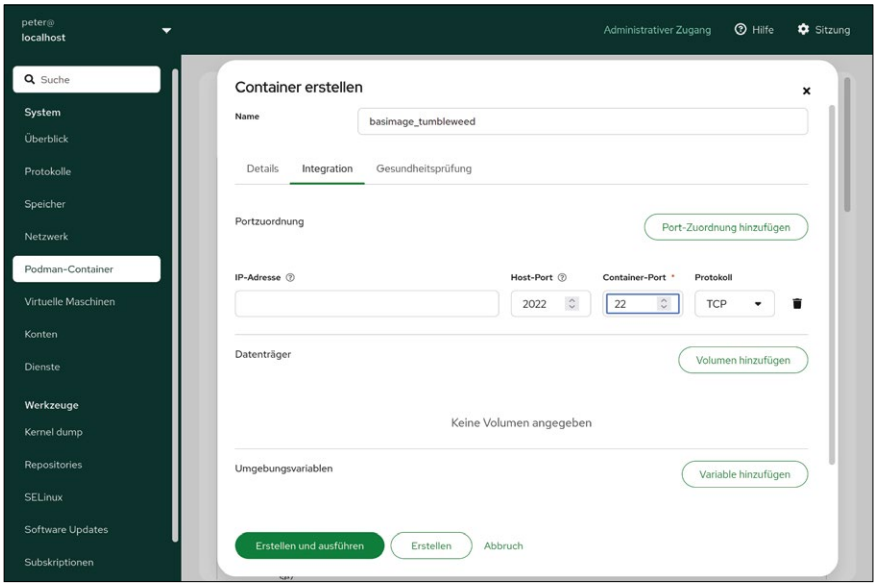
Um Anwendungen remote zu nutzen, brauchen Sie eine Möglichkeit zum Austausch von Dateien. Das gelingt über ein in den Container eingebundenes Verzeichnis aus dem MicroOS-Gastsystem, das dieser wiederum per NFS als Netzwerk-Volume exportiert. Dazu eignet sich dann ein Ordner im Home eines Benutzer-Accounts (`/home/User/shared/`, denn das Root-Dateisystem ist nur zum Lesen eingehängt.

Erstellen Sie dazu mittels `sudo nano /etc/exports` in der dafür vorgesehenen Konfigurationsdatei die Zeile `/home/User/shared/ 192.168.0.0/16(rw)`, die das Verzeichnis im Heimnetz bereitstellt. Starten und aktivieren Sie danach mit `systemctl enable nfs-server.service --now` den NFS-Server des Gastsystems. Nun sollte sich das geteilte Verzeichnis beschreibbar im lokalen Verzeichnis `/mnt` einhängen lassen (Listing 4). Klappt das, ergänzen Sie die `/etc/fstab` und die Zeile aus Listing 5. Sie hängt das Verzeichnis so ein, dass der Bootvorgang des Rechners nicht hängenbleibt, falls er den MicroOS-Server nicht erreicht.

Der eben erstellte Container ist nicht für den direkten Einsatz gedacht, sondern als Basis für weitere Container, in denen Sie schließlich beliebige Software installieren, die sich dann remote per SSH ausführen lässt. Stoppen Sie den Container also über den entsprechenden Menü-Button am Ende der Zeile. Dann wählen Sie im selben Menü den Eintrag *Comitten* aus, der aus dem gegenwärtigen Zustand ein Image als Basis für neue Container erstellt.

Geben Sie einen Abbildnamen aus Kleinbuchstaben und ohne Leerzeichen ein. Ersetzen Sie außerdem den beim Starten des Containers ausgeführten Befehl `/bin/bash` durch `/bin/bash /etc/startup`, also durch den Aufruf des eben erstellten Skripts, das den SSH-Server und eine Bash-Instanz startet.

Nach einem Mausklick auf den Eintrag *Comitten* erscheint dann in Cockpit unter *Podman-Container* | *Images* ein neuer Eintrag. Auf der Basis dieses Abbilds entstehen neue Container, die bereits alle Vorbereitungen für die Remote-Ausführung grafischer Programme enthalten 5. Diese Container müssen den Netzwerk-Port 22 des SSH-Servers im



8 Vor dem Erzeugen eines Containers sind die relevanten Netzwerk-Ports, hier der SSH-Port 22, einem freien Port des Gastsystems zuzuordnen.

Listing 3: SSH-Daemon-Startskript

```
/usr/sbin/sshd
/bin/bash -l
```

Container auf einen freien Port des MicroOS-Gastsystems weiterleiten.

Dies gelingt im Dialog *Container erstellen* über den Reiter *Integration* mit einem Klick auf *Port-Zuordnung hinzufügen*. Lassen Sie in der danach erscheinenden Zeile die *IP-Adresse* leer, wählen Sie für den *Host-Port* die *22* und für *Container-Port* zum Beispiel *2022*. Das *Protokoll* belassen Sie auf *TCP*. Da sich ein Port nur einmal belegen lässt, verwenden Sie als *Container-Port* für weitere Container die *2122*, *2222* und so weiter.

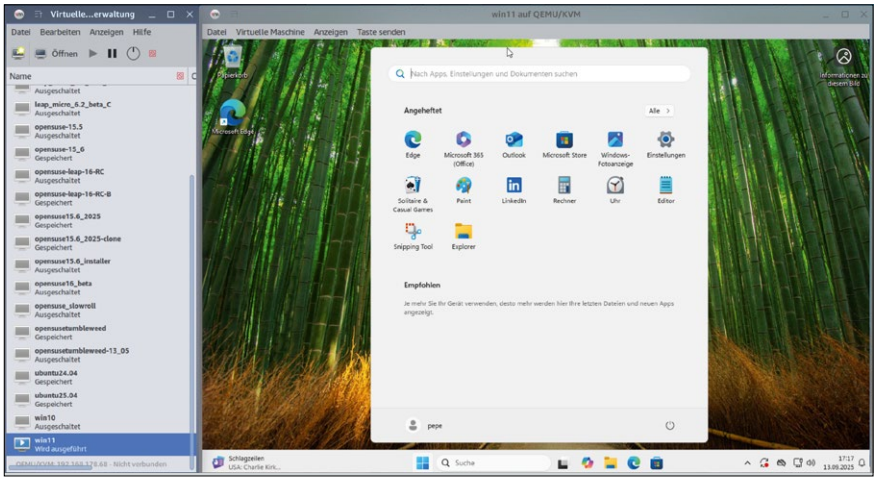
Um das per NFS geteilte Verzeichnis `/home/User/shared/` im Container verfügbar zu machen, klicken Sie auf den Dialog *Volume hinzufügen*. Hier tragen Sie bei *Host-Pfad* `/home/User/shared/` ein. Als *Container-Pfad*, an dem die Daten für die im Container laufenden Anwendungen erscheinen, dient ein ansonsten nicht belegter Ordner direkt im Stammverzeichnis wie `/shared`.

Aufmachen!

Sobald die Portzuordnungen und das Einbinden des externen Verzeichnisses erledigt sind, legt *Erstellen und ausführen* den Container an und startet ihn 8. Klappen Sie die zum Container gehörige Zeile in der Rubrik *Container* auf und setzen Sie in der *Konsole* zunächst das Root-Passwort per `passwd`. Legen Sie dann mit `useradd User` einen neuen Benutzer an und vergeben Sie für ihn mittels `passwd User` ein eigenes Passwort.

Dann gelangen Sie von jedem Linux-Rechner im Netzwerk aus mit dem Aufruf aus der ersten Zeile von Listing 6 zum Container. Den Port im Parameter `-p` passen Sie an den vom Container genutzten Port an. Er lässt sich im laufenden Container nach Ausklappen der Informationen im Reiter *Integration* herausfinden. Bei `192.168.0.X` handelt es sich um die IP des MicroOS-Gastsystems. Als Root (zweite Zeile) installieren Sie via Zypper Programme, die Sie im Container nutzen möchten (dritte Zeile). In der Regel müssen Sie dann noch zusätzlich Schriften installieren, zum Beispiel über das Paket *bitstream-vera-fonts*.

Es gibt Programme, deren interaktive Reaktion über Remote-X quälend langsam ausfällt, beispielsweise den Bitmap-Editor Gimp. In einem solchen Fall sorgt dann der Aufruf aus Listing 7 von einem Wayland-Desktop aus für eine flüssigere Arbeitsweise. Zuvor müssen Sie der Datei `/home/User/.bashrc` im Container die Zeile `export XDG_RUNTIME_DIR=/home/User` für das zum Remote-Login genutzte Benutzerkonto hinzufügen.



9 Um Fremdsysteme auszuführen, müssen Sie das schwerere Geschütz der virtuellen Maschine auffahren, ein Container genügt nicht.

Listing 4: Manuelles Einhängen

```
$ mount 192.168.0.X:/home/User/shared /mnt
```

Listing 5: Einhängen via Fstab

```
192.168.0.X:/home/User/shared /mnt/ nfs noauto,x-systemd.automount,
nofail 0 0
```

Komplettpaket

Es gibt immer noch Anwendungsfälle, die nur virtuelle Maschinen abdecken: Manchmal möchte man nicht nur einzelne Anwendungen begutachten, sondern ein ganzes Linux-System. Nicht-Linux-Systeme oder -Anwendungen sind auch auf virtuelle Maschinen angewiesen 9, in Containern funktionieren sie nicht.

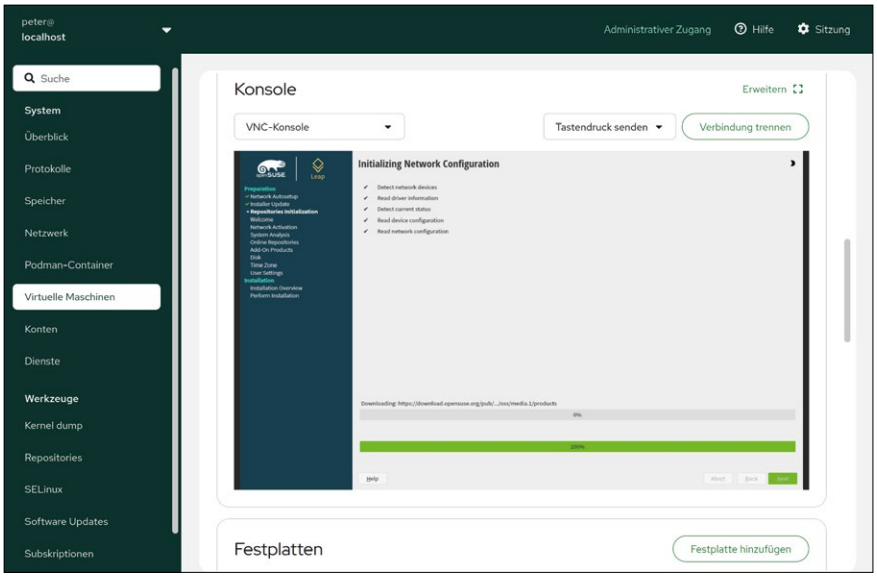
Dem trägt MicroOS Rechnung, indem es neben der Container-Laufzeitumgebung auch eine Verwaltung für virtuelle Maschinen mitbringt. In der Cockpit-Rubrik *Virtuelle Maschinen* laden Sie nicht nur gängige Distributionen wie Ubuntu, Fedora, Debian oder OpenSuse herunter, sondern erstellen auch virtuelle Maschi-

Listing 6: Programme installieren

```
$ ssh User@192.168.0.X -X -p 2022
$ su
$ zypper install Paket
```

Listing 7: Aufruf per Waypipe

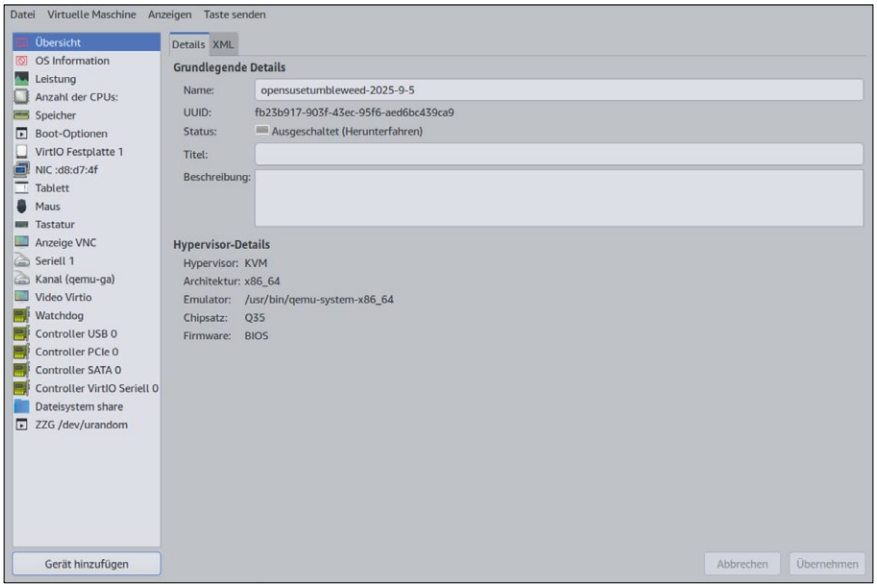
```
$ waypipe ssh -p 2022
User@192.168.0.X Programm
```



10 Nach dem erfolgreichen Anlegen einer virtuellen Maschine lässt sich die Installation dann in der VNC-Konsole im Webfrontend Cockpit erledigen.

nen. Alternativ lässt sich ein per Netzwerk-Share auf den Server übertragenes ISO-Image als Installationsbasis nutzen.

Nach der Einrichtung der virtuellen Maschine startet der Installer direkt in einem Unterfenster im Browser 10. Das Cockpit-Frontend beherrscht die grundlegende Verwaltungsoperationen wie etwa das Bearbeiten von Festplatten und auch das Anlegen von Snapshots oder



11 Der Virt-Manager bietet zwar keine komfortable Einrichtungsfunktion für Distributionen, dafür dann jedoch eine leistungsfähige Konfigurationsfunktion für virtualisierte Hardwarekomponenten bestehender Maschinen.

das Erstellen von mit dem Host-System geteilten Verzeichnissen.

Prinzipiell lassen sich virtuelle Maschinen zudem nach der Installation per Unterfenster im Browser bedienen, doch komfortabel ist das nicht. Deshalb sollte Cockpit eigentlich das Programm Desktop Viewer starten können, einen Remote-Betrachter für virtuelle Maschinen. Das funktionierte im Test allerdings nicht.

Das stört nicht weiter, denn mit der Einrichtung virtueller Maschinen auf dem MicroOS-Server hat die browserbasierte Managementsoftware ihre Schuldigkeit schon getan. Im Virt-Viewer, einer verbreiteten Managementsoftware für virtuelle Maschinen, gelingt es leicht, eine SSH-basierte Remote-Verbindung zur unter MicroOS laufenden Libvirt-Instanz aufzubauen: *Datei | Neue Verbindung* öffnet den entsprechenden Dialog. Dort aktivieren Sie die Option *mit dem entfernten Rechner über SSH verbinden* und tragen den Benutzernamen sowie die IP-Adresse oder den Hostnamen des entfernten Rechners ein.

Der Virt-Manager startet und stoppt virtuelle Maschinen. Er zeigt ihren Bildschirm in einem Fenster an und dieses auf Wunsch auch in der Vollbildansicht, in der sich die virtuelle Maschine kaum von einem lokal installierten System unterscheiden lässt. Er erstellt auf Wunsch Snapshots, und seine Management-Funktionen für virtuelle Hardwarekomponenten sind deutlich leistungsfähiger als diejenigen von Cockpit 11.

Bei der unter MicroOS eingerichteten Netzwerkverbindung für virtuelle Maschinen per NAT ist es nicht möglich, per SSH auf die virtualisierten Systeme zuzugreifen. Diese Einschränkung lässt sich jedoch leicht beheben. Öffnen Sie dazu in Cockpit die Rubrik *Netzwerk* und klicken Sie auf *Bridge hinzufügen*. Wählen Sie dann das Netzgerät *enp1s0*. Nach dem *Hinzufügen* erhält der Rechner schnell wieder seine bisherige IP-Adresse. Das aktive Netzgerät des Rechners heißt dann ab sofort `bridge0`.

In diese Netzwerk-Bridge kann sich ein virtuelles Netzwerkgerät einklinken, das sich von allen Rechnern im Heimnetz erreichen lässt. Allerdings beherrscht Cockpit das Anlegen von virtuellen Schnittstellen von Typ Bridge nicht. Das Anlegen per Hand verursacht jedoch keinen

großen Aufwand. Klicken Sie zunächst in der Rubrik den Link *1 Netzwerk* links oben an und löschen Sie dort über den Menü-Button am Ende der entsprechenden Zeile das Netzwerk *default*.

Legen Sie dann die Datei `/tmp/network.xml` mit dem Inhalt aus dem Listing 8 an. Der Befehl `virsh net-define network.xml` erzeugt daraus ein neues Netzwerk für das Gerät `bridge0`, als *Weiterleitungs-Modus* dient *Bridge*. Das prüfen Sie an derselben Stelle in Cockpit, an der Sie zuvor schon das alte Netzwerk *default* gelöscht haben.

Aktivieren Sie dort das Bridge-Netzwerk und wählen Sie nach dem Aufklappen der Zeile ganz unten *Starten, wenn der Host hochfährt*. Nun lassen sich Konsolen- und grafische Programme auf der virtuellen Maschine von allen entsprechend vorbereiteten Geräten im Heimnetz aus starten. Dort muss der SSH-Server laufen, der Firewall-Port 22 offenstehen und für Anwender eines Wayland-Desktops zusätzlich noch das Paket *waypipe* installiert sein.

Fazit

Das schlanke und dank Read-only-System robuste Leap Micro 6.2 bootet in wenigen Sekunden, das Basissystem mit laufendem Cockpit-Verwaltungsdienst belegt weniger als 700 MByte RAM. Wollen Sie experimentelle Software testen, andere Distributionen ausprobieren oder Anwendungen von einem zentralen Server im Heimnetz aus zur Verfügung stellen, nutzen Sie dazu komfortabel im Webfrontend administrierte Container mit minimalem Ressourcen-Overhead.

Bevorzugen Sie mehr Ressourcen beanspruchende virtuelle Maschinen, finden Sie selbst dafür eine handliche Verwaltung in Leap Micro. Übrigens baut Leap Micro ausschließlich auf Leap-Paketen auf. Egal, ob Leap oder Tumbleweed: Im Arbeitssystem selbst lassen sich Container und virtuelle Maschinen auf dieselbe Weise verwalten. Leap belegt allerdings mehr Systemressourcen, die der virtualisierten Anwendung dann nicht zur Verfügung stehen. (*uba/jlu*)

Listing 8: Bridge anlegen

```
<network>
  <name>default</name>
  <forward mode="bridge" />
  <bridge name="bridge0" />
</network>
```

Weitere Infos und interessante Links
www.linux-user.de/qr/52043

Xargs macht Dampf auf der Kommandozeile

Turbolader

Unter klassischen Kommandozeilen-Utilities ist Xargs so etwas wie das Phantom. Nach der Lektüre dieses Artikels könnte es Ihr treuer Begleiter sein. Thomas Reuß

README

Xargs gehört zu den heimlichen Helden unter Linux. Gerade im Zusammenspiel mit Find spielt es alle Trümpfe aus. Das gilt vor allem, wenn es um aufwendige Anpassungen an Dateien geht, etwa Kodierungsvorgänge, das Ändern von EXIF-Daten und vieles mehr.

Wer kennt nicht die klassische Problemstellung: Wie finde ich Text in großen Mengen von Dateien? Früher oder später erwischt diese Aufgabe jeden. Dann wird es oft hektisch – völlig zu Unrecht. Ich habe in den Ausgaben 11/2023 und 12/2023 schon einmal der Macht von Find gehuldigt, dabei aber dessen wohl fähigsten Begleiter sträflich vernachlässigt. Das muss sich ändern.

Der eingangs formulierten Aufgabenstellung könnte ich mich halbwegs naiv nähern, indem ich schlicht und einfach Grep mit entsprechenden Parametern (Listing 1, erste Zeile) auf das Problem loslasse. Diese Herangehensweise endet oft mit dem Fehler aus der zweiten Zeile von Listing 1. Da haben wir den Salat: Im Zielordner gibt es zu viele Dateien. Zu viele, um sie Grep als Parameter durch die Bash übergeben zu lassen.

Daher muss ich anders ansetzen und greife auf Find zurück. Wer sich kurz mit dem Tool beschäftigt, stolpert schnell über den Parameter –exec. Mit diesem Ansatz übergibt Find dem Kommando nach –exec die aktuelle Fundstelle, gekennzeichnet durch ein geschweiftes Klammerpaar (Listing 1, letzte Zeile). Ein abschließendes, per Backslash entschärftes Semikolon sorgt dafür, dass Find das Ende des Arguments von –exec erkennt.

In der Tat löst dieser Ansatz mein Problem, er hat allerdings seinen Preis. Was hier passiert, ist langsam und vor allem verschwenderisch. Für jede Fundstelle muss Find einen eigenen Grep-Prozess starten, egal, ob es den Suchbegriff, den Grep finden soll, überhaupt in der jeweiligen Datei gibt. Das geht viel effizienter, und zwar mit Xargs.

Traumpaar

Anstatt die Fundstellen einzeln per Grep zu bearbeiten, ist es deutlich besser, sie über eine Pipe an Xargs zu delegieren. Das unterscheidet sich nicht arg von der Variante via –exec. Das erste Kommando in Listing 2 sucht nach Dateien, die auf `.txt` enden. Doch dazu wird nicht für jede einzelne Datei Grep gestartet; stattdessen übernimmt xargs die Übergabe der Fundstellen an Grep.

Das ist erheblich effizienter, denn Xargs leitet nicht nur eine Datei als Argu-

ment weiter, sondern mehrere. Wie viele es genau sind, hängt stark vom Betriebssystem ab, etwa von den Security Limits. Klar: Wer einem Programm unbegrenzt Parameter übergeben darf, schafft es vielleicht, darüber eine Attacke gegen das Betriebssystem zu fahren. Letztlich bestimmen ein paar Variablen den Spielraum von Xargs – dazu später mehr.

Ich habe also Find und Grep mit Xargs dazwischen verkittet. Das läuft wie geschmiert und deutlich flotter als zuvor. Doch dummerweise erhalte ich jetzt zahlreiche Fehler. Es scheint, als gäbe es Probleme bei Ordnern und Dateien mit Leerzeichen im Namen. Eine kurze Überprüfung zeigt, dass Xargs in diesem Fall Wort für Wort an Grep schickt, was mangels existierender Dateien gar nicht funktionieren kann. Bei Grep kommt möglicherweise der Code aus der zweiten Zeile von Listing 2 an.

Dass Linux, die Bash sowie alle Userspace-Werkzeuge mit Leerzeichen im Dateinamen zurechtkommen, steht außer Frage. Ungünstig ist lediglich, dass Find keinen Unterschied zwischen Leerzeichen im Dateinamen und einem Leerzeichen als Trennzeichen zwischen Dateien macht. Genau da liegt das Problem. Glücklicherweise verfügen sowohl Find als auch Xargs über eine Lösung dafür.

Ich kann Find veranlassen, die Funde per binärer 0 terminiert an Xargs zu übergeben. Das klappt mithilfe des Schalters `print0`. Xargs wiederum bietet mit dem Schalter `-0` das passende Gegenstück dazu. Somit wird aus dem Kommando in der ersten Zeile das aus der dritten Zeile von Listing 2.

Optimal durchdacht

Selbstverständlich bleibt es nicht bei einem trivialen Aufruf von Xargs. Das Tool wartet mit weitaus mehr Möglichkeiten auf. Mit `--max-args` (kurz `-n`) schnürt es handliche Pakete, die ich wiederum an Grep weiterreichen kann. Das hat den Vorteil, dass ich die Suche nicht, wie erwähnt, über nur eine Datei abfeuere, sondern gleich über beispielsweise 50 Dateien. Das reduziert Overhead.

Dabei sollten Sie im Hinterkopf behalten, dass sich bestimmte Parameter wie `--max-args` und `--max-chars` (kurz `-s`) gegenseitig beeinflussen. Logischerweise kann `--max-args` nicht größer als `--max-chars` sein. Immerhin enthält `--max-chars` zu den Parametern und deren Argumenten auch noch die Trennzeichen. Zur Beruhigung kann ich anführen, dass es mir noch nie unbeabsichtigt gelungen ist, Xargs zu überfordern.

Mit dem Parameter `--max-procs` bringt Xargs ein mächtiges Feature mit. Der Parameter – mein persönlicher Liebling – ermöglicht, die Anzahl der parallel auszuführenden Prozesse zu steuern. Das heißt, mit `--max-procs 4 grep ...` (kurz `-P`) könnte ich die Find-Suchergebnisliste mit vier Grep-Instanzen durchsuchen. Auf modernen CPUs erscheint das dem geneigten User wie der Heilige Gral. Tatsächlich gibt es beim parallelen Greppen noch andere Parameter. Grep gewinnt

Listing 1: Grep

```
01 $ grep -Hni needle haystack/*
02 bash: /usr/bin/grep: Argument list too long
03 $ find . -type f -exec grep -H -n -i needle {} \;
```

Listing 2: Grep und Xargs

```
$ find . -type f -name "*.txt" | xargs grep -Hni needle
$ grep -Hni needle haystack/2025-10 Artikel Linux Magazin.txt/
$ find . -type f -print0 -name "*.txt" | xargs -0 grep -Hni needle
```

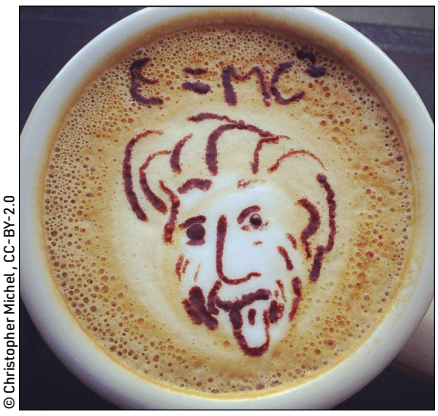
Listing 3: Schnelligkeit

```
01 $ find . -type f -print0 -name "*.txt" | \
02   xargs -0 grep -Hni NEEDLE
```

```
03 $ find . -type f -print0 -name "*.txt" | \
04   xargs -0 -P 4 -n 40 grep -Hni NEEDLE
```

Listing 4: Oggenc vs. Oggenc parallel

```
01 $ time oggenc -q9 *.wav
02 $ time find . -type f -iname "*.wav" -print0 | xargs -0 -n 1 -I % -P $(nproc) /usr/bin/oggenc -q9 %
```



1 Mit den Ausgabewerten von `time` wäre Albert Einstein wohl nicht einverstanden.

durch das Parallelisieren nicht notwendigerweise Schnelligkeit, vor allem nicht auf rotierenden Platten – schließlich müssten die Schreib-/Leseköpfe sehr häufig bewegt werden. Selbst auf einer NVMe-SSD ist also der erste Befehl aus Listing 3 schneller als der zweite.

Ganz anders sieht die Situation bei der Parallelisierung aufwendiger Tasks aus, etwa beim Encoding oder Transcoding. Deutlich spürbar wird das, wenn ich nicht greppe, sondern zum Beispiel eine Vielzahl von WAV-Dateien zu Ogg Vorbis encodiere. In diesem Fall lassen sich die einzelnen Tasks gut auf eigene Prozesse verteilen. Hier besteht kein Synchronisierungsbedarf, da es nicht vorkommt, dass mehrere Prozesse gleichzeitig schreibend auf ein gemeinsames Objekt zugreifen – beinahe. Tatsächlich teilen sich die Prozesse die Standardausgabe, was durchaus verwirren kann.

Auf einer Multi-Core-CPU kann je ein Kern einen eigenen Oggenc-Prozess bedienen. Die Anzahl der CPU-Kerne stellen Sie über das Standard-Tool Nproc ein. Ich habe mir für diesen Benchmark zwei Fälle vorgenommen (Listing 4). Im ersten lasse ich testhalber eine Reihe von WAV-Dateien sequenziell direkt von Oggenc umwandeln. Im zweiten Fall übergebe ich die Dateien per Find an Xargs. Dabei soll Xargs die von Find übergebenen Werte, ersetzt durch `%`, an Oggenc weiterreichen. In beiden Fällen wird mittels Time die Zeit gestoppt.

Bevor Sie sich auf die Ergebnisse stürzen, beachten Sie bitte die Xargs-Parameter `-n 1` und `-I %`. Mit dem ersten erzwinge ich, dass genau ein Dateiname zu genau einer WAV-Datei an Oggenc übergeben wird. Das ist notwendig, weil Oggenc sonst mit der sequenziellen Encodierung beginnen würde. Der zweite Parameter `-I %` ist noch interessanter. Er bewirkt, dass die an Xargs übergebenen Dateinamen im Platzhalter `%` gespeichert werden, den ich am Schluss der Parameterkette verwende. Der Schalter `-q9` veranlasst hier nur, dass Oggenc bei der Enkodierung nahezu das Maximum an Qualität ausschöpft.

Der Unterschied ist beachtlich. Time misst die Laufzeiten der Befehle und zeigt bei der Echtzeit eine deutliche Abweichung. Dank paralleler Verarbeitung beendet die Xargs-Variante dieselbe Aufgabe in rund einem Drittel der Zeit wie die sequenzielle Variante.

Selbst nach vielen Jahren mit Linux war mir nie ganz klar, was genau die drei Zeiten bedeuten, die Time (Listing 5) liefert. Dank Xargs bin ich nun ein wenig schlauer, denn die Manpage von Time liefert die gesuchten Informationen. Mit `real` ist die Wall-Clock-Zeit gemeint, also die tatsächlich vergangene Zeit vom Start bis zur Beendigung des Prozesses. Beim Begriff tatsächliche oder reale Zeit rotiert Albert Einstein 1 vermutlich im Grab, darum ist „Zeit nach Newton" womöglich die bessere Formulierung.

Die `user`-Zeit wiederum ist die Zeit, die die CPU im User-Mode operiert, also Nicht-Kernel-Code innerhalb des Prozesses ausführt. Somit handelt es sich bei der `sys`-Zeit um die Zeit, die die CPU im Prozess in Kernel-Code verbringt. Seien Sie gewarnt: Die `user`- addiert mit der `sys`-Zeit muss nicht notwendigerweise die `real`-Zeit ergeben: Es fehlen die Zeiten für den Prozesswechsel sowie Wartezeiten auf andere Betriebsmittel wie unter anderem Disk I/O.

Außer den genannten Parametern möchte ich noch den Schalter `-t` erwähnen. Entgegen aller gewohnten Konventionen aktiviert er den Verbose-Mode, zum Beispiel zu Debugging-Zwecken. Außerdem bietet Xargs mit `--show-limits` die Möglichkeit, die tatsächlichen Größen für Parameter auf dem aktuellen System zu erfragen. Dementsprechend finden Sie darüber die maximale Gesamtlänge der Argumente sowie die maximale Befehlslänge heraus.

Listing 5: Laufzeiten

```
# sequenzielle Verarbeitung
        real    0m53,998s
        user    0m52,653s
        sys     0m1,265s
# parallele Verarbeitung:
        real    0m16,871s
        user    1m5,848s
        sys     0m1,581s
```

Cleveres Sammeln

Es kommt immer wieder vor, dass ich Dateien mit bestimmten Merkmalen in einem separaten Ordner zusammenfassen möchte – Protokolle etwa, um sie leichter nebeneinander korrelieren zu können. Dazu hat Xargs mit Platzhaltern das richtige Werkzeug parat.

Angenommen, ich möchte die Log-Dateien von Datenbank, Webserver, Betriebssystem und so weiter der letzten drei Tage im Ordner `LogAnalyse/` haben. Mit Find wähle ich die Protokolldateien aus, Xargs darf sich anschließend um die Kopiervorgänge kümmern. Genau das erledigt der erste Aufruf aus Listing 6.

Find schreibt wieder per `-print0` Null-terminierte Dateinamen nach Stdout. Xargs nimmt sie per `-0` entgegen und bewahrt sie im Platzhalter `{}` auf, um sie anschließend so dem Kopierwerkzeug Cp als ersten Parameter zu übergeben. Bitte beachten Sie dabei, dass bei Cp Quelle und Ziel vertauscht wären, falls Sie diesen Platzhalter weglassen würden. Ohne den Platzhalter würde Xargs einfach die jeweils aktuelle Log-Datei als letzten Parameter an Cp übergeben, sie also als Kopierziel und eben nicht als Kopierquelle verwenden.

Selbstverständlich kann Xargs nicht nur an ein zusätzliches Kommando weiterleiten. Es gibt Möglichkeiten, eine per Platzhalter gespeicherte Zeichenkette auch an mehrere Programme zu übergeben. Statt direkt einen Prozess zu starten, stößt der zweite Aufruf aus Listing 6 eine weitere Shell an, die dann durch Xargs die Platzhalter erhält. So lässt sich in einem Befehl das Verzeichnis per Ls auflisten und direkt hinterher mit Du auch der Speicherbedarf anzeigen.

Fazit

Xargs gehört zu den heimlichen Helden unter Linux. Es ist definitiv eines der Tools, bei denen ich mich stets gefragt habe, warum ich mich nicht schon viel früher damit beschäftigt habe. Gerade im Zusammenspiel mit Find spielt Xargs alle Trümpfe aus. Das gilt vor allem, wenn es um aufwendige Anpassungen an vorhandenen Dateien geht, zum Beispiel Kodierungsvorgänge, Anpassungen an EXIF-Daten und vieles mehr. (*csi/jlu*)

Dateien zum Artikel herunterladen unter
www.linux-user.de/dl/52666

Weitere Infos und interessante Links
www.linux-user.de/qr/52666

Listing 6: Platzhalter

```
$ find /var/log/ -mtime -3 -print0 | \
  xargs -0 -I {} cp -av {} ~/temp/LogAnalyse
```

```
$ find . -type f -name "*.txt" | \
  xargs -I {} sh -c 'ls -l {}; du -h {}'
```

COMPUTEC
marquard group

Ein Unternehmen der MARQUARD MEDIA GROUP AG
Verleger: Jürg Marquard

Redaktion/Verlag	Computec Media GmbH Redaktion LinuxUser Dr. -Mack-Straße 83 90762 Fürth Telefon: (0911) 2872-110 E-Mail: redaktion@linux-user.de Web: www.linux-user.de
Geschäftsführer	Rainer Rosenbusch
Chefredakteur, Brand/Editorial Director	Jörg Luther (jlu, v. i. S. d. P.), joerg.luther@computec.de
Stellv. Chefredakteurin Strategy & Operations	Carina Schipper (csi), carina.schipper@computec.de
Redaktion	Uli Bantle (uba), ulrich.bantle@computec.de Thomas Leichtenstern (tle), thomas.leichtenstern@computec.de
Linux-Community	Jörg Luther, joerg.luther@computec.de
Datenträger	Thomas Leichtenstern (tle), cdredaktion@linux-user.de
Ständige Mitarbeiter	Erik Bärwaldt, Hans-Georg Eßer, Frank Hofmann, Peter Kreußel, Claudia Meindl, Thomas Reuß, Tim Schürmann (tsc), Anna Simon, Daniel Tibi, Ferdinand Thommes, Uwe Vollbracht, Harald Zisler
Titel & Layout	Titel: Alexandra Böhm Titelmotiv: nerthuz, 123RF.com Layout: Alexandra Böhm
Sprachlektorat	Stefan Gneiting, Sabine Schmitt
Produktion	Martin Closmann (Ltg.), martin.closmann@computec.de Uwe Hönig, uwe.hoenig@computec.de
Anzeigen	Verantwortlich für den Anzeigenteil: Bernhard Nusser Es gilt die Anzeigenpreisliste vom 01.01.2024.
Mediaberatung D/A/CH	Bernhard Nusser, bernhard.nusser@computec.de Tel.: (0911) 2872-254, Fax: (0911) 2872-241
Mediaberatung UK/USA	Brian Osborn, bosborn@linuxnewmedia.com
New Business	Viktor Eippert (Project Manager)
E-Commerce & Affiliate	Daniel Waadt (Head of E-Commerce & Affiliate), Veronika Maucher, Andreas Szedlak, Frank Stöwer
Abo	Die Abwicklung (Rechnungsstellung, Zahlungsabwicklung und Versand) erfolgt über unser Partnerunternehmen: DPV Deutscher Pressevertrieb GmbH Leserservice Computec 20080 Hamburg Deutschland
Einzelhefte und Abo-Bestellung	https://shop.computec.de
Leserservice Deutschland	Ihre Ansprechpartner für Reklamationen und Ersatzbestellungen E-Mail: computec@dpv.de Tel.: (0911) 99 39 90 98 Fax: (01805) 861 80 02* (* 0,14 €/min via Festnetz, max. 0,42 €/min via Mobilnetz)
Österreich, Schweiz	E-Mail: computec@dpv.de
und weitere Länder	Tel.: +49 911 99399098 Fax: +49 1805 8618002
Supportzeiten	Montag 07:00 – 20:00 Uhr, Dienstag – Freitag: 07:30 – 20:00 Uhr, Samstag 09:00 – 14:00 Uhr
Pressevertrieb	DMV Der Medienvertrieb GmbH & Co. KG Meßberg 1, 20086 Hamburg http://www.dermedienvertrieb.de
Druck	EDS Zrínyi Zrt., Nádas utca 8, 2600 Vác, Ungarn
ISSN	1615-4444

marquard group

Deutschland:

4PLAYERS, AREAMOBILE, BUFFED, GAMESWORLD, GAMESZONE, GOLEM, LINUX-COMMUNITY, LINUX-MAGAZIN, LINUXUSER, N-ZONE, GAMES AKTUELL, PC GAMES, PC GAMES HARDWARE, PC GAMES MMORE, PLAY 4, RASPBERRY PI GEEK, VIDEOGAMESZONE

Marquard Media Hungary:

JOY, JOY-NAPOK, INSTYLE, SHOPPIEGO, APA, ÉVA, GYEREKLÉLEK, FAMIILY.HU, RUNNER`S WORLD

ABONNEMENT

Probeabo (3 Ausgaben)	Deutschland	Österreich	Schweiz
No-Media-Ausgabe	15,00 €	15,00 €	15,00 €
DVD-Ausgabe	19,00 €	19,00 €	19,00 €
Jahres-Abo (12 Ausgaben)	**Deutschland**	**Österreich**	**Schweiz**
No-Media-Ausgabe	91,00 €	99,00 €	106,00 €
DVD-Ausgabe	112,00 €	120,00 €	127,00 €
Jahres-DVD zum Abo *	6,70 €	6,70 €	6,70 €
Preise Digital	**Deutschland**	**Österreich**	**Schweiz**
Heft-PDF Einzelausgaben Digital	8,50 €	8,50 €	8,50 €
Digital-Abo (12 Ausgaben)	84,99 €	84,99 €	84,99 €
Kombi Digital + Print (No-Media-Ausgabe, 12 Ausgaben)	103,00 €	111,00 €	118,00 €
Kombi Digital + Print (DVD-Ausgabe, 12 Ausgaben)	124,00 €	132,00 €	139,00 €

Die Probe-, Jahres- und Digital-Abos erhalten Sie in unserem Webshop unter https://shop.computec.de. Die Auslieferung erfolgt versandkostenfrei.

(*) Nur erhältlich in Verbindung mit einem Jahresabonnement der Printausgabe von LinuxUser.

Internet	https://www.linux-user.de
News und Archiv	https://www.linux-community.de
Facebook	https://www.facebook.com/linuxuser.de

Schüler- und Studentenermäßigung: 20 Prozent gegen Vorlage eines Schülerausweises oder einer aktuellen Immatrikulationsbescheinigung. Der aktuelle Nachweis ist bei Verlängerung neu zu erbringen. Andere Abo-Formen, Ermäßigungen im Ausland etc. auf Anfrage. Adressänderungen bitte umgehend beim Kundenservice mitteilen, da Nachsendeaufträge bei der Post nicht für Zeitschriften gelten.

Rechtliche Informationen

LinuxUser Community Edition

LinuxUser gibt es auch als Community-Edition: Dabei handelt es sich um eine rund 30-seitige PDF-Datei mit ausgewählten Artikeln aus der aktuellen Ausgabe, die parallel zur Veröffentlichung des gedruckten Hefts erscheint.

Die kostenlose Community-Edition steht unter einer Creative-Commons-Lizenz, die es erlaubt, „das Werk zu vervielfältigen, zu verbreiten und öffentlich zugänglich machen". Sie dürfen die LinuxUser Community-Edition also beliebig kopieren, gedruckt oder als Datei an Freunde und Bekannte weitergeben, auf Ihre Website stellen – oder was immer Ihnen sonst dazu einfällt. Lediglich bearbeiten, verändern oder kommerziell nutzen dürfen Sie sie nicht. Darum bitten wir Sie im Sinn des „fair use". Weitere Informationen finden Sie unter: http://linux-user.de/CE

Probleme mit den Datenträgern

Falls es bei der Nutzung der Heft-DVDs zu Problemen kommt, die auf einen defekten Datenträger schließen lassen, dann schicken Sie bitte eine E-Mail mit einer genauen Fehlerbeschreibung an die Adresse computec@dpv.de. Wir senden Ihnen dann umgehend kostenfrei einen Ersatzdatenträger zu.

README

In jedem Artikel in diesem Heft liefern spezielle Auszeichnungen und grafische Elemente wichtige Zusatzinformationen zum Text.

Der Mensch lebt nicht vom Text allein: Zu jedem Artikel in diesem Heft gehören eine Reihe von Zusatzinformationen, die das bloße Narrativ um weiterführende Inhalte ergänzen. Manche davon integrieren sich direkt in den Textfluss, andere stehen als gesonderte grafische Elemente in der sogenannten Marginalspalte, also dem teilweise freien Bereich an der rechten beziehungsweise linken Seitenkante.

Typografische Konventionen

Eine blaue Einfärbung hebt Verweise auf Tabellen und Kästen hervor: siehe Kasten *Kastentitel*. Die Kursivierung signalisiert hier wie in vielen anderen Fällen eine symbolische Bezeichnung; in einem Codebrocken könnte das etwa so aussehen:

```
$ cat "EinLängererTextbrocken" >> ⮐
Ausgabe.txt
```

Der „Umbruchhaken“ am Ende der ersten Zeile des Codes verweist hier darauf, dass es sich in diesem Fall eigentlich um eine einzige Eingabezeile handelt, die nur aus Platzgründen im Druck umgebrochen werden musste.

Die Kursivierung kann neben Platzhaltern auch andere Elemente bezeichnen, wie Paketnamen und Benutzerkonten, beispielsweise *build-essential* und *root*. Aber auch Menüpunkte drucken wir kursiv ab, wobei in Menüfolgen eine Pipe die einzelnen Elemente trennt: *Sonstiges* | *Textkodierung* | *Unicode*.

Dateien zum Artikel herunterladen unter

www.linux-user.de/dl/52037

Gelegentlich begegnen Ihnen in den Artikeln auch orangefarbig hinterlegte Textstellen. Sie verweisen auf ein Glossar, das den markierten Begriff kurz erläutert. Sie finden den Glossartext dann in einer der Marginalspalten.

Tasten und Tastenfolgen

Ein Buchstabe oder eine Buchstabenfolge in eckigen Klammern, wie [Esc], steht symbolisch für einen Tastendruck. Dabei dient als Schreibweise grundsätzlich die Beschriftung der Tasten einer deutschen Tastatur. Ein Druck auf [T] erzeugt also ein kleines „t“, die Kombination [Umschalt]+[T] ein großes „T“.

Dabei signalisiert das Pluszeichen zwischen Tasten, dass man sie gleichzeitig drücken muss, ein Komma dagegen, dass sie nacheinander betätigt werden müssen. Das allseits beliebte Copy & Paste gelingt also mit [Strg]+[C], [Strg]+[V].

Lesen Sie etwas von der Super-Taste, handelt es sich dabei um die eigentlich korrekte Bezeichnung der Taste, die in Microsoft-Umgebungen „Windows-Taste“ heißt und auf der bei vielen Tastaturen das entsprechende Logo prangt.

Infos und Downloads

An einzelnen Stellen im Text finden Sie das Zeichen , das auf eine weiterführende Information verweist. Um an die Links zum Artikel zu gelangen, blättern Sie ans Ende des Artikels, wo Sie einen Kasten Weitere Infos und interessante Links finden. Entweder tippen Sie die dort angegebene URL `www.linux-user.de/qr/Nummer` in einen Webbrowser ein – das führt Sie auf eine Webseite mit allen Links zum Artikel –, oder Sie scannen mit dem Smartphone oder Tablet den im Kasten abgedruckten QR-Code ein und surfen so direkt zur Seite mit den Links.

Beispiel für Heft-DVD-Inhalt
`LU/Ordner/`

Glossar: Nähere Definition zum Verständnis eines Begriffs oder einer Abkürzung.

Analog funktioniert der Kasten Dateien zum Artikel herunterladen unter mit der URL `www.linux-user.de/dl/Nummer`. Er bringt Sie auf eine Webseite, die auf interessante Downloads zum Artikel verweist. (Das Exemplar unten links dient nur als Beispiel und führt ins Nirgendwo.)

Heft-DVD

Die preisgünstigere No-Media-Edition von LinuxUser kommt ohne Datenträger, doch die meisten Leser bevorzugen die am Kiosk erhältliche Ausgabe mit Heft-DVD. Bei Artikeln, zu denen Inhalte auf der DVD gehören, finden Sie auf der ersten Doppelseite einen grauen „Halbkreis mit Loch“ (siehe oben), der eine optische Disk symbolisiert. Der Text darunter bezeichnet den zugehörigen DVD-Inhalt und nennt gegebenenfalls auch das Verzeichnis, in dem sich dieser auf dem Datenträger befindet. (*jlu*)

Weitere Infos und interessante Links

www.linux-user.de/52037

Vorschau auf 12/2025

Die nächste Ausgabe erscheint am 21.11.2025

© Pavel Shlykov / 123RF.com

Linux der Zukunft

Linux befindet sich derzeit im Wandel, was den Unterbau moderner Distributionen und die technische Basis für ihren Betrieb angeht. Dabei lassen sich ganz klar zwei Schlüsseltechnologien identifizieren: immutable Systeme und Containerisierung. Beide sind dem Verlangen nach mehr Sicherheit und dem leichteren distributionsübergreifenden Einsatz von Software geschuldet. Wir sehen uns deswegen die XDG Desktop Portals samt Backends näher an, die den Containereinsatz unter Linux erst alltagstauglich machen. Außerdem vergleichen wir mit Aurora, BlendOS, NixOS und Vanilla OS vier topaktuelle Immutable-Distros.

Tonemapper Darktable AgX

Darktable besitzt bei FOSS-affinen Fotografen Kultstatus. Obwohl nicht mehr ganz neu, wartet das Tool regelmäßig mit Neuerungen auf und integriert die fortschrittlichsten Bearbeitungstechniken der Filmindustrie. Der neueste Streich der Darktable-Entwickler ist das professionelle Tonemapper-Modul AgX.

Netzdiagnose mit Iftop

Nicht alles, was von außen auf das Netzwerk-Interface zugreift oder Daten von innen nach draußen befördern will, führt Gutes im Schilde. Mit Iftop gehen Sie den Dingen auf den Grund und erhalten Klarheit. Das Tool spürt obendrein sogar fehlerhafte Switch-Ports, defekte LAN-Kabel und gestörte Funkstrecken auf.

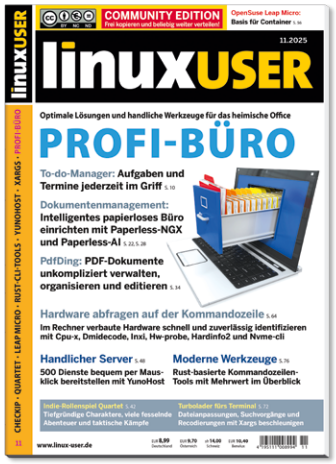
Die Redaktion behält sich vor, Themen zu ändern oder zu streichen.

Heft als DVD-Edition

- 108 Seiten Tests und Workshops zu Soft- und Hardware
- 2 DVDs mit Top-Distributionen sowie der Software zu den Artikeln. Mit bis zu 18 GByte Software das Komplettpaket, das Unmengen an Downloads spart

Heft als No-Media-Edition

- Preisgünstige Heftvariante ohne Datenträger für Leser mit Breitband-Internet-Anschluss
- Artikelumfang identisch mit der DVD-Edition: 108 Seiten Tests und Workshops zu aktueller Soft- und Hardware



Community-Edition-PDF

- Über 30 Seiten ausgewählte Artikel und Inhaltsverzeichnis als PDF-Datei
- Unter CC-Lizenz: Frei kopieren und beliebig weiter verteilen
- Jeden Monat kostenlos per E-Mail oder zum Download

DVD-Edition (10,99 Euro) oder No-Media-Edition (8,99 Euro) Einfach und bequem versandkostenfrei bestellen unter:

http://www.linux-user.de/bestellen

Jederzeit gratis herunterladen unter:

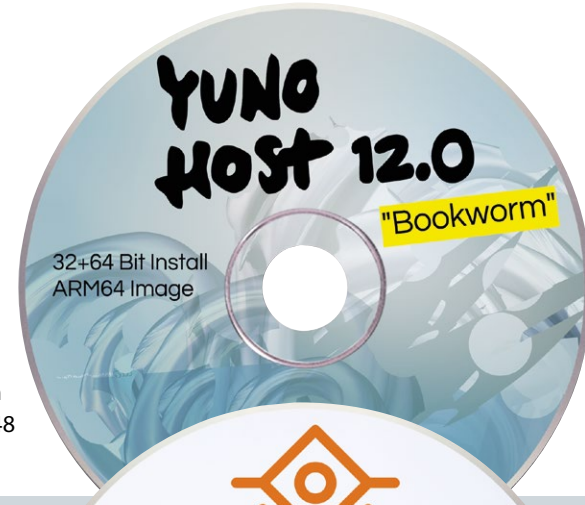
http://www.linux-user.de/CE

Neues auf der Heft-DVD

Die Heft-DVD liegt ausschließlich der LinuxUser DVD-Edition bei.

YunoHost 12.0

Die einfach zu bedienende, auf Debian 12 aka „Bookworm“ basierende Server-Distribution YunoHost bietet auf einem gut vorkonfigurierten Server derzeit rund 550 mit wenigen Mausklicks installierbare Dienste und Anwendungen an. Eine grafische Weboberfläche hilft beim Konfigurieren des Systems und der Installation von Apps. Dazu zählen unter anderem Dokuwiki, Gitlab, Immich, Jitsi Meet, Mastodon, Nextcloud, Peertube, Pixelfed, Syncthing, Tiddlywiki, Wordpress, Vaultwarden und viele weitere. Sie booten die Installations-Distribution vom Datenträger. Im Ordner `isos/` finden Sie die ISO-Images der ARM- und 64-Bit-Version, die Sie auf einen USB-Stick kopieren oder eine DVD brennen können. ➥ S. 48



OpenSuse Leap Micro 6.2

Leap Micro ist als reine Laufzeitumgebung für Container und virtuelle Maschinen ausgelegt. Jegliche Nutzsoftware läuft isoliert vom Grundsystem in Containern oder virtuellen Maschinen ab, als zusätzlicher Schutz ist das Root-Dateisystem nur lesbar eingehängt. Die remote im Browser aufrufbare Benutzeroberfläche Cockpit gestattet eine einfache, grafische Handhabung der Container, VMs und des gesamten Systems im Browser. Wollen Sie experimentelle Software testen, andere Distributionen ausprobieren oder Anwendungen von einem zentralen Server im Heimnetz aus bereitstellen, bietet Leap Micro dazu eine ideale Basis. ➥ S. 56

SparkyLinux 2025.09 XFCE

Das auf Debian basierende SparkyLinux richtet sich vornehmlich an Gamer. Neben Standard-Tools wie einem Webbrowser und dem VLC-Player wartet es mit 80 Spielen auf. Das Angebot reicht dabei von kleinen Klassikern wie Minesweeper, Sudoku und Mahjongg bis hin zu Langzeitmotivatoren wie 0 A.D. und OpenArena. Darüber hinaus bringt die Distribution noch einen Steam-Client mit, der es erlaubt, zusätzliche Spiele einzurichten. Den Unterbau stellt der Linux-Kernel in Version 6.16.7, die Installation übernimmt Calamares 3.4. Sie booten die Live-Distribution über den Datenträger, das ISO-Image finden Sie unter `isos/`.

Tails 7.0

Das Live-System zum anonymen Surfen über das Tor-Netzwerk liegt in einer neuen Major-Version vor. Sie aktualisiert den Unterbau auf Debian 13, komprimiert die Pakete mit dem Zstd-Algorithmus und erfordert mindestens 3 GByte Hauptspeicher. Mit dem Umstieg auf „Trixie“ kommt auch der neuere Kernel 6.12.43 zum Zug, der vor allem moderne Hardwarekomponenten unterstützt. Gestrichen haben die Entwickler unter anderem Unrar, Sq, das Werkzeug Power Statistics und Aircrack-ng. Letzteres lässt sich aber nachinstallieren. Sie booten die Live-Distribution vom Datenträger, das ISO-Image finden Sie unter `isos/`. (*tle/jlu*)

Darstellung symbolhaft