

# Editorial

---

## Mehr ausprobieren, weniger schwitzen

---

Früher, als Server noch im eigenen Haus liefen und dem Admin ihre Abwärme ins Gesicht bliesen, war ihr Betrieb harte und schweißtreibende Arbeit. Das ist jetzt vorbei, das Zeitalter der Cloud hat begonnen. Statt Server ins Rack zu wuppen, mietet man sie heute bequem vom Schreibtisch aus und statt des Admins schwitzt nur noch die Kreditkarte, die beim Vermieter hinterlegt ist.

Wer allerdings glaubt, in diesem Zeitalter mit weniger IT-Personal auszukommen, hat nur die Werbeprospekte der Cloudprovider gelesen und noch nie einen gemieteten Server eingerichtet. Daher keine Sorge: Arbeit gibt es noch immer genug, anstrengend ist sie weiterhin, nur anders. Statt im Kabelsalat im Netzwerkschrank verstrickt man sich jetzt in Konfigurationsdateien im YAML-Format.

Dieses Heft soll Ihnen die Arbeit erleichtern und zum Ausprobieren anregen. Sie finden Hintergrundwissen zu Docker, Kubernetes und Ansible sowie Praxistipps und Rezepte, um populäre Software wie Nextcloud oder Rocket.Chat auf eigenen oder gemieteten Servern zu betreiben. Sie lesen, wie ein fiktives Start-up seine Probleme mit Servern aus der Cloud löst und wie Sie einen geeigneten Anbieter für Mietserver finden. Wenn Sie eigene Cloud-Dienste programmieren wollen, finden Sie einen Einstieg in die noch junge, aber in der Cloudszene populäre Programmiersprache Go.

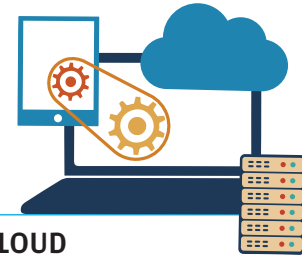
In erster Linie macht das Hantieren mit flexiblen Servern Spaß und viel, viel schneller als damals, als Server noch laut und schwer waren.



Jan Mahn



# Inhalt



## WEGE IN DIE CLOUD

Mit dem einfachen Anmieten von Servern allein ist es nicht getan. Erst mit dem passenden Anbieter und einer geeigneten Strategie gelingt ein echter Umzug in die Cloud.

- 6 Serverinfrastruktur mieten in der Cloud
- 14 Cloudprovider aus Europa
- 22 Cloudserver bestellen und einrichten
- 26 Automatisierung mit Ansible
- 36 Schnellstart mit Ansible
- 42 Cloud-Dienste mit Terraform
- 46 Kubernetes für Docker-Kenner

## CLOUD AUF EIGENER HARDWARE

Nicht immer brauchen Sie gemietete Server, um Ihre Dienste zu betreiben. Sie können Ihre Cloud auch auf eigenen Maschinen einrichten.

- 54 Starthilfe für den eigenen Server
- 62 Dienste ins Internet bringen
- 68 Einstiegsfreundlicher Heimserver

## DIENSTE FÜR DIE CLOUD

Viele nützliche Anwendungen sind dank Containern schnell auf Cloudservern oder eigener Hardware zum Laufen gebracht.

- 72 Wegweiser im Heimnetz
- 74 CI/CD mit GitHub Actions
- 80 Online-Dokumentationen mit MkDocs
- 86 Statusseiten für IT-Infrastruktur
- 90 Passwortmanager in Eigenregie hosten
- 98 Mitgliederversammlungen virtuell

## EIGENE DATEN IN NEXTCLOUD

Die Open-Source-Komplettlösung Nextcloud unterstützt mit Dateispeicher, Kalenderserver und vielen Erweiterungen die Gruppenarbeit in der eigenen Cloud.

- 104 Nextcloud als Bausatz
- 108 Chat-Server hinter Nextcloud
- 112 Nextcloud: Tipps für die Gruppenarbeit
- 118 Videokonferenzen mit Nextcloud Talk



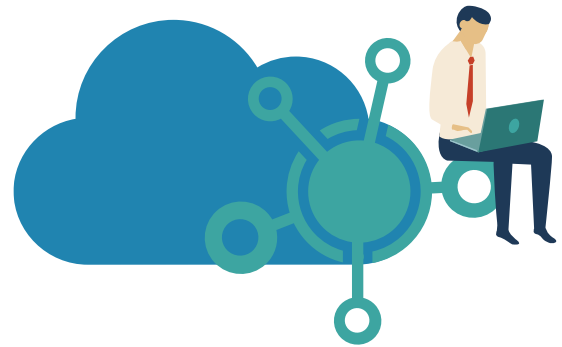
---

## E-MAIL-SERVER SELBER HOSTEN

---

Ein eigener Mailserver ist Ihre Unabhängigkeitserklärung von Mail Providern. Grundlagen und eine praktische Anleitung helfen bei der Einrichtung.

- 126 E-Mail: Begriffe und Protokolle
- 134 Mailserver einrichten mit Mailcow



---

## CLOUD-PROGRAMMIERSPRACHE GO

---

Für eigene Dienste, die bereit für den Einsatz in der Cloud sind, ist die Programmiersprache Go die erste Wahl. Ein praktischer Einstieg für Programmiererfahrene.

- 142 Entwickeln für die Cloud
- 148 Kommandozeile
- 154 Unit Tests
- 160 Server-Dienste
- 166 Zugangsdaten

---

## ZUM HEFT

---

- 3 Editorial
- 53 **Aktion:**  
Videokurs Continuous Delivery von A bis Z
- 159 Impressum

The image shows the cover of the book 'c't Cloud'. The title 'c't CLOUD' is prominently displayed in blue and black. Below it, the subtitle reads 'Einstieg und Praxis für Admins und Entwickler'. A red banner in the top right corner says '7. Jahr Videokurs mit Leserrabatt (€ 59)'. The cover features an illustration of a person standing on a cloud, surrounded by various icons like a laptop, a smartphone, and a server rack. A table of contents is listed on the left side of the cover, and a barcode with the price '€ 14,90' is at the bottom right.

**c't CLOUD**  
Einstieg und Praxis für Admins und Entwickler

7. Jahr Videokurs mit Leserrabatt (€ 59)

**Nützliche Cloud-Dienste betreiben**

- 126 Eigenen E-Mail-Server installieren & administrieren
- 98 Mitgliederversammlungen selbst hosten
- 90 Passwort-Server sicher und zuverlässig betreiben

**Teamwork mit Nextcloud**

- 104 Gruppenarbeit organisieren
- 112 Office im Browser nutzen
- 118 Online-Konferenzen per Chat und Video

**Cloud-Programmiersprache Go**

- 142 Effizient programmieren für viele Plattformen
- 160 Mit Go eigene Cloud-Dienste entwickeln

**Profi-Know-how für Cloud-Admins & Entwickler**

- 6, 14 Cloud: Konzepte und Software, die Sie kennen sollten
- 22, 36 Strategien für den Einstieg und den Umzug in die Cloud

facebook.com/ctspecials

€ 14,90  
OUP € 27,90  
HÖ € 14,90  
LBB € 17,90

4 196982 314900 05



# Cloudserver bestellen und einrichten

**Wer heute eine gute Idee für ein Online-Angebot hat, braucht keine eigenen Server und kein Rechenzentrum, wohl aber einen Schlachtplan für den Weg in die Cloud. Viele kleine und größere Probleme müssen Start-ups und große Unternehmen gleichermaßen lösen, bevor der eigene Code in gemieteter Umgebung läuft.**

Von **Jan Mahn**

**A**m Anfang steht die Idee: Eine App und eine Website sollen entstehen, über die man Fotos hochladen und mit Freunden teilen kann. Das Entwicklungsteam ist klein: Einer kümmert sich um die mobilen Apps, einer um das Web-Frontend und ein Dritter baut das Backend, das Bilder und Benutzerinformationen speichert und wieder anzeigt. So oder so ähnlich sind viele IT-Start-ups personell aufgestellt, für eine ganze Abteilung, die sich um den Betrieb eigener Server kümmert, ist weder Geld noch Platz im Co-Working-Bürogebäude da. Außerdem soll sich die App bestenfalls gegen Instagram, Twitter und Facebook durchsetzen und zum weltweiten Erfolg werden – dann wäre ein einziges Rechenzentrum in Deutschland zu wenig und zu unflexibel.

Also entscheiden sich die Start-up-Gründer dafür, die Anwendung von Anfang an auf gemieteter Infrastruktur zu betreiben. Das Start-up, das sich diese Anwendung ausgedacht hat, heißt Piquip Ventures und ist rein fiktiv. Dieser Artikel vermittelt einen Eindruck, welche Probleme die Entwickler auf dem Weg zu einer skalierbaren Cloud-Anwendung lösen müssen: Von DNS über Containerisierung, automatische Updates, skalierbaren Speicherplatz bis hin zu Monitoring gibt es viel einzurichten. Auf den nächsten Seiten geht es zunächst um ein Rezept für einen einzelnen Server – wie man es schafft, dass beliebig viele gemietete Server im Verbund arbeiten, erfahren Sie in einer der nächsten Ausgaben.

Dieser Überblick sollte nicht als Schritt-für-Schritt-Anleitung zum Nachbauen verstanden werden – ei-

nige der vorgestellten Werkzeuge erfordern durchaus eine längere Einarbeitungszeit. Wie immer in der IT führen bei jedem Teilproblem zahlreiche Wege zum Ziel und es gibt immer gute Gründe, andere als die hier beschrittenen Pfade zu wählen. Wenn Sie Ideen aus diesem Artikel für die eigene Cloud-Strategie übernehmen wollen, finden Sie das beschriebene Rezept über [ct.de/w8ma](https://ct.de/w8ma).

## Reproduzierbar und flexibel

Wichtig ist den fiktiven Unternehmern von Piquip, dass die gesamte Einrichtung ihrer Infrastruktur ohne Handarbeit funktioniert. Die Entwickler träumen davon, mit nur einem Befehl neben der produktiven Umgebung [app.piquip.de](https://app.piquip.de) schnell eine baugleiche Testumgebung unter der Adresse [testing.piquip.de](https://testing.piquip.de) hochzufahren und der Geschäftsführer würde gern auf Knopfdruck eine Demo-Umgebung für Produktpräsentationen starten können. Damit das klappt, sollen alle Schritte reproduzierbar sein und in einem Rezepte-Ordner liegen. Im einfachsten Fall könnte das eine Sammlung von Bash-Skripten sein.

Bequemer wird die Arbeit mit dem populären Werkzeug Ansible. Eine ausführliche Einführung in das Open-Source-Programm lesen Sie auf Seite 26. Die Grundidee: In sogenannten Playbooks im YAML-Format definiert man, welche Konfigurationsschritte Ansible erledigen soll. Über Plug-ins kann Ansible zum Beispiel mit den APIs von Cloudanbietern sprechen und Ressourcen einkaufen, es kann sich aber auch per SSH mit Servern verbinden und dort Einrichtungsschritte vornehmen und so auch komplexeste Anwendungen einrichten. Der große Vorteil: Die Einrichtungsschritte sind idempotent – das bedeutet, dass man das Rezept ohne Schaden immer wieder ausführen kann. Ansible achtet darauf, dass nur Änderungen umgesetzt werden und die mehrfache Ausführung eines Rezepts keine negativen Folgen hat. Vereinfacht gesagt: Wenn man schon drei Server bestellt hat und gerne noch einen vierten

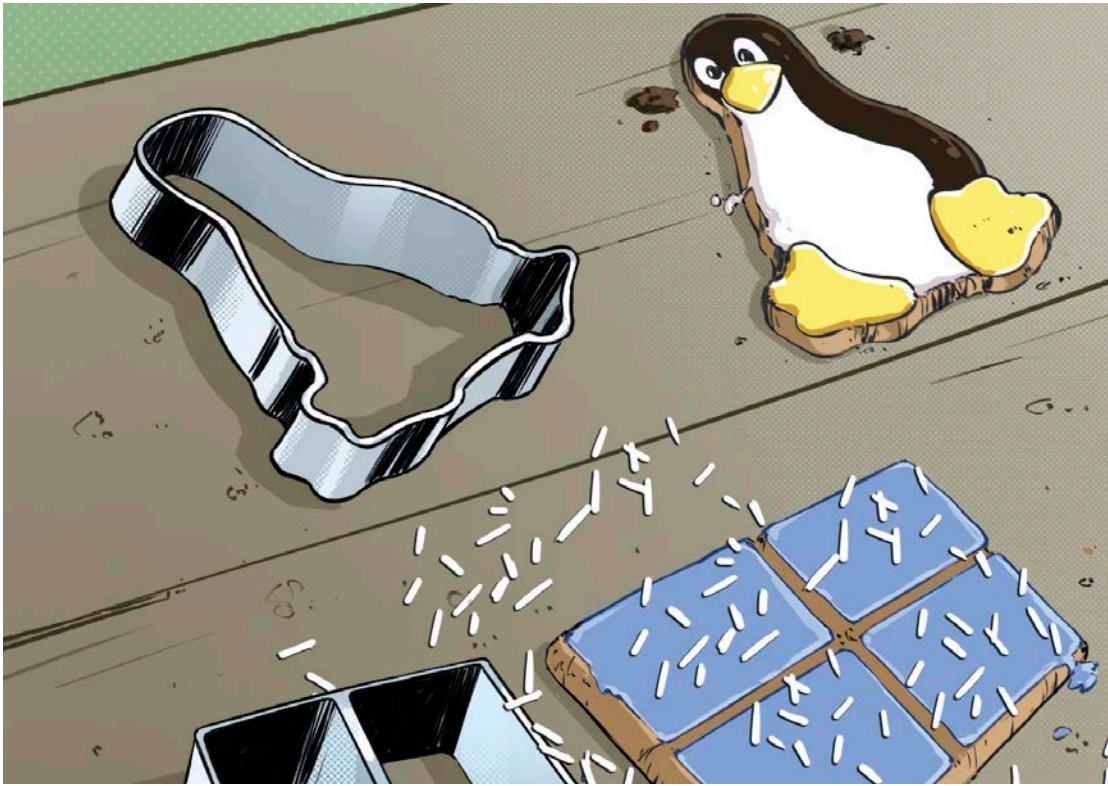
sogenannten Provider, der die Domain [piquip.de](https://piquip.de) bei der deutschen Registrierungsstelle Denic registriert und dort die Adresse zu einem Nameserver hinterlegt. Viele Hoster bieten ihren Kunden diese Dienste im Paket mit anderen Dienstleistungen an. Gerade in klassischen Webhoster-Paketen oder bei virtuellen Servern im langfristigen Tarif ist eine Domain oft im Preis enthalten und der Hoster stellt auch den DNS-Server bereit.

Es gibt aber gute Gründe, die Aufgaben von Provider und DNS-Anbieter von Anfang an zu trennen und die eigene Domain nicht an ein vermeintlich günstiges Paket vom Hoster zu binden. Solche in Webhosting integrierten DNS-Verwaltungen kann man meist nur über eine Weboberfläche steuern – in der schnelllebigen Cloud-Welt möchte man aber nicht jede IP-Adresse jedes Mietserver per Hand dort eintippen. Wenn man sich für einen unabhängigen Dritten als reinen Domainverwalter entscheidet, erspart man sich außerdem Probleme, die entstehen können, wenn man sich irgendwann einem anderen Anbieter für die Server zuwendet.

Die Start-up-Gründer sind daher gut beraten, sich einen Provider zu suchen, der Domains als lose Ware zum Stückpreis anbietet. Der Betreiber des DNS-Servers kann davon völlig unabhängig sein. Viele setzen für die DNS-Verwaltung ihrer Domains mittlerweile auf das US-Schwergewicht Cloudflare: Dessen Schwerpunkt ist die Abwehr von DDoS-Attacks, ganz nebenbei betreiben sie aber auch DNS-Server – wenn man keine Extras braucht, ist das Angebot kostenlos.

Bei einem deutschen Domainhändler bestellen die Piquip-Admins daher nur die Domain und hinterlegen dort die Adressen zu zwei Nameservern von Cloudflare. Der große Vorteil: Alle Funktionen der Weboberfläche von Cloudflare kann man auch per API fernsteuern und damit gut skripten. Bestenfalls nutzt man die Weboberfläche nur einmalig, um das API-Token zu generieren, und arbeitet dann mit reproduzierbaren Rezepten per API. Besonders komfortabel wird das mit Ansible – weil Cloudflare weit

**Lesen Sie mehr in c't Cloud 2021**



Bilder: Albert Hulim

# Schnellstart mit Ansible

Die Pakete für Ansible installieren, SSH-Schlüssel generieren und verteilen, sudo einrichten – alles für sich genommen ganz einfach, doch unterm Strich Aufwand. Unser Bootstrap-Skript und einige weitere Beispiele kürzen Ihren Weg zum ersten eigenen Playbook ab.

Von **Peter Siering**

**W**enn nach der Lektüre des vorangehenden Artikels der Tatendrang groß ist, mit Ansible loszulegen, sind Sie hier richtig: Das Folgende liefert den Grundstock dafür, um nach dem Einrichten von Ansible auf dem Control-Host sofort

einen Ziel-Host damit zu konfigurieren. Control-Host kann Ihr Arbeitsplatz-PC, ein Notebook oder eine Linux-VM sein. Auf dem Gerät müssen Sie Ansible installieren und es empfiehlt sich, eine einfache Verzeichnisstruktur anzulegen, um darin Dateien für

Ansible abzulegen; idealerweise verwalten Sie die Dateien mit Git.

## SSH

Erzeugen Sie ein Verzeichnis „ansible“ und erstellen Sie darin das Verzeichnis .ssh. Wechseln Sie in das Verzeichnis und erzeugen Sie in diesem Verzeichnis ein SSH-Schlüsselpaar:

```
mkdir -p ansible/.ssh
cd ansible/.ssh
ssh-keygen -f ansible -t ecdsa -b 521
```

Dieses Schlüsselpaar in den Dateien ansible und ansible.pub soll Ansible zukünftig nutzen, um sich mit Ziel-Hosts per SSH zu verbinden. Ansible sieht keine Möglichkeit vor, eine Passphrase für den Schlüssel zu hinterlegen. Wenn Sie sich vorerst Tricksereien mit ssh-agent ersparen wollen, geben Sie keine Passphrase ein, wenn ssh-keygen danach fragt.

Für den administrativen Zugriff auf die aus dem Verzeichnis „ansible“ heraus verwalteten Ziel-Hosts empfiehlt es sich, zusätzlich ein SSH-Schlüsselpaar für den Root-Zugriff zu erzeugen:

```
ssh-keygen -f root -t ecdsa -b 521
```

Für diesen Key sollten Sie eine Passphrase vergeben. Die im späteren Verlauf vorgestellten Playbooks werden den öffentlichen Schlüssel auf den verwalteten Ziel-Hosts verteilen. Wenn Sie weitere Schlüssel für den Zugriff auf die Ziel-Hosts brauchen, etwa fürs Monitoring, können Sie die jetzt ebenfalls mit ssh-keygen erzeugen und in dem Verzeichnis ansible/.ssh ablegen.

## Ziel-Host

Außer dem Control-Host brauchen Sie für die ersten Schritte einen Ziel-Host, um Ihre Automatisierungs-

```
#!/bin/bash
# enable host for ansible
set -e
PUBKEY="AAAAE2VjZh...f1B2A== ps@"
id -u ansible /dev/null 2>&1 || \
adduser ansible --disabled-password \
--gecos "" --quiet
mkdir -p /home/ansible/.ssh
echo "$PUBKEY" \
> /home/ansible/.ssh/authorized_keys
chown -R ansible /home/ansible/.ssh
apt-get update
apt-get install sudo
grep -q ansible /etc/sudoers || \
echo "ansible ALL = (ALL) \
NOPASSWD: ALL" >> /etc/sudoers
```

**Ein Shellskript (debansstrap.sh) hilft Ihnen, ein frisches Debian-System mal eben für die weitere Konfiguration durch Ansible vorzubereiten.**

den Nutzer ansible und deaktiviert seine Anmeldung per Passwort. Es legt in der nutzerspezifischen SSH-Konfiguration (in ~/.ssh/authorized\_keys) auf dem Ziel-Host den öffentlichen Teil des zuvor erstellten Schlüsselpaars ab. Dadurch kann sich der Nutzer ansible mit dem privaten Schlüssel mit dem Host verbinden. Den Schlüssel müssen Sie in Ihr Skript übertragen (Variable PUBKEY, im Skript gekürzt).

Als Eigentümer für die SSH-Konfigurationsdatei setzt das Skript den Nutzer ansible ein. Anschließend holt es aktuelle Paketlisten, richtet sudo ein und erlaubt dem Nutzer ansible ohne Eingabe eines Pass-

Lesen Sie mehr in c't Cloud 2021

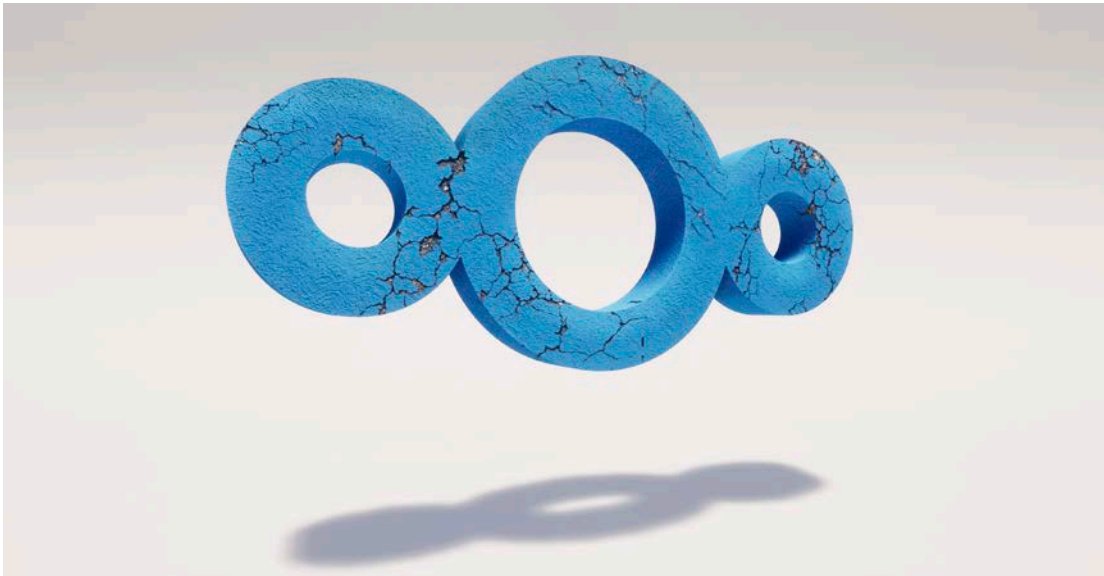


Bild: Hagen Kober

# Nextcloud: Tipps für die Gruppenarbeit

**Nextcloud bringt verteilt zusammenarbeitende Computer und Menschen einander näher. Doch die alternative Cloud ist manchmal spröde: Unsere Hinweise machen die neue Heimat schöner.**

Von **Peter Siering**

**N**extcloud kann beim Hoster, auf einem Mietserver oder auf dem hauseigenen NAS laufen. Dieser Beitrag ergänzt unsere Installationsanleitung um Tipps für frisch gebackene Nextcloud-Eigener und -Admins. Er hilft, die überbordenden Erwartungen zu justieren und gibt pragmatische Hilfestellungen zur geeigneten Nutzung.

Dreh- und Angelpunkt von Nextcloud sind Dateien: Nextcloud-Nutzer haben im Wesentlichen drei Möglichkeiten, Dateien darauf abzulegen. Sie können sie per Webbrowser dort hoch- und runterladen. Das Gleiche gelingt über das WebDAV-Protokoll ähnlich wie eine Freigabe im Netzwerk.

Außerdem kann spezielle Client-Software Dateien speichern und abrufen.

Der Zugriff per Browser ist umständlich. Die zweite Methode, WebDAV, funktioniert wegen Unzulänglichkeiten des Protokolls [1] nur selten gut. Es möchte Dateien stets vollständig laden. Eine aktive Bearbeitung, wie man sie von Dateifreigaben kennt, erlaubt WebDAV nicht. Außerdem variieren die Fähigkeiten von WebDAV-Clients stark. Kurzum: Nutzen Sie die beiden Methoden nur, wenn sich der Zugang nicht anders realisieren lässt.

Die deutlich bessere Methode stellt die Nextcloud-eigene Client-Software dar. Die gleicht auto-



matisch Dateien zwischen lokalen Verzeichnissen und Nextcloud ab. Wenn sich die Datei auf einer Seite ändert, überträgt der Client sie auf die andere. Er nutzt dafür Funktionen, die im Hintergrund dafür sorgen, dass nicht nur die aktuelle Fassung der Datei erhalten bleibt, sondern auch die vorherigen Versionen – Nextcloud versioniert automatisch.

Wenn der Benutzer eine alte Fassung seiner Datei braucht, etwa weil er sich beim Bearbeiten eines Textes vergaloppiert hat, kann er über die Web-Oberfläche von Nextcloud alte Versionsstände der Datei abrufen. In die Web-Oberfläche bringt ihn ein Klick aus der Bedienoberfläche der Client-Software („Nextcloud im Browser öffnen“). Dort kann er im Kontextmenü einer Datei, das unter anderem über die drei Punkte erreichbar ist, unter „Details“ die „Versionen“ einsehen und alte wiederherstellen.

## Dateiaustausch

Das klappt gut, wenn ein Benutzer die Ablage für seine eigenen Dateien verwendet, kriegt aber eine eigene Dynamik, wenn weitere Benutzer ins Spiel kommen und an den gleichen Dateien werkeln. Nextcloud erlaubt es nämlich, Dateien oder Verzeichnisse mit anderen Nutzern zu teilen. Die sehen die dann auch in ihrem Nextcloud-Client beziehungsweise ihrem synchronisierten Verzeichnis. Ändern die dort eine geteilte Datei, verpasst Nextcloud ihr ebenfalls eine neue Version.

Spannend wird es, wenn eine geteilte Datei gleichzeitig auf mehreren Clients geändert wird: Nextcloud legt dann von sich aus eine Kopie der Datei mit den lokalen Änderungen an und verpasst ihr einen Zeitstempel. Der Benutzer muss dann die Dateien vergleichen und die Daten von Hand zusammensetzen – die Synchronisierung kann da also keine Wunder vollbringen. Für das gleichzeitige Arbeiten an ein und derselben Datei bieten sich

Funktionen zur Online-Zusammenarbeit an – dazu gleich mehr.

Zum Nutzungsalltag mit Nextcloud gehören allerdings auch hartnäckigere Synchronisierungsfehler, die damit enden, dass der Server eine Datei verriegelt („Locked“). Der Nextcloud-Client hat dann beim Versuch kapituliert, eine Datei zu synchronisieren. In einem solchen Fall kann sich der Nutzer allein nicht mehr helfen, sondern muss den Nextcloud-Admin um Hilfe bitten: Der versetzt das System in den Wartungsmodus und leert mit speziellen Befehlen eine Datenbanktabelle. Die einzelnen Schritte zeigt der Artikel später am Beispiel einer Container-Installation.

## Gemeinsame Struktur

Subtilen Unterhaltungswert hat Nextcloud, wenn Mitglieder einer Arbeitsgruppe Dateien in geteilten Ordnern mit besten Absichten aufräumen. Sobald sie diese in einen privaten, nicht geteilten Ordner verschieben, verschwinden die Dateien aus dem Blickfeld der anderen – Nextcloud verhindert das nicht. Solange die Dateien aber in geteilten Ordnern verbleiben, besteht keine Gefahr.

Soll eine Nextcloud-Instanz Benutzern eine einheitliche, statische Verzeichnisstruktur vorsehen, so muss der Admin zunächst mithelfen: Er aktiviert in der Installation die Gruppenordner mit der App „Group Folders“. Anschließend tauchen in den „Einstellungen“ unter „Verwaltung“ die „Gruppen-Ordner“ auf. Dort lässt sich dann für jede benötigte gemeinsame Verzeichnishierarchie ein Gruppenordner anlegen. Die Ordner erhalten in der Web-Oberfläche ein für die Nutzer gut sichtbar um Personenprofile ergänztes Symbol.

Knifflig wird es, wenn innerhalb einer solchen Struktur unterschiedliche Zugriffsrechte gewünscht sind. Dann kann man einander überdeckende Grup-

Lesen Sie mehr in c't Cloud 2021

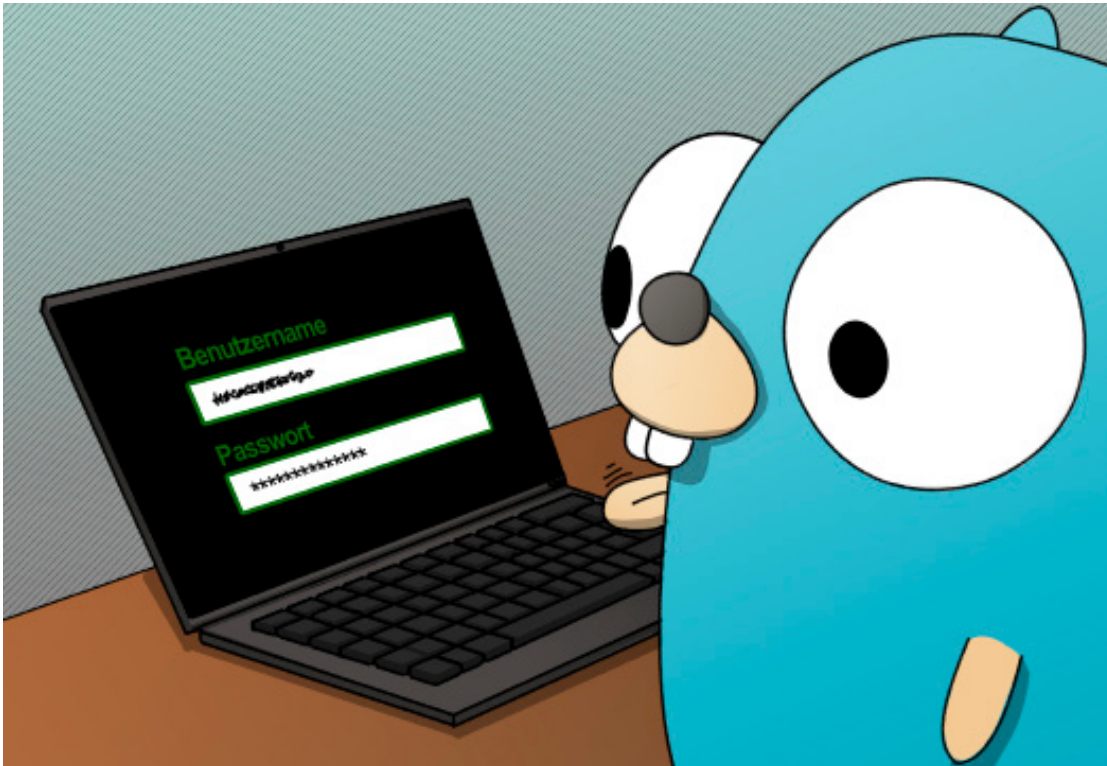


Bild: Thorsten Hübner

# Go-Einstieg: Zugangsdaten

Viele Programme erfragen Anmeldedaten vom Benutzer, um sich an einem Server anzumelden. Oft landet das Kennwort dann im Klartext in einer Datei auf der Festplatte – dabei ist es kein Hexenwerk, den sicheren Kennwortspeicher des Betriebssystems zu nutzen. Mit Go gelingt das plattformübergreifend.

Von **Jan Mahn**

**D**ie „schönsten“ Probleme beim Programmieren sind solche, die man vorm Programmieren noch gar nicht hatte. Eigentlich hatte man sich mit einem ganz anderen Ziel an den Computer gesetzt und verbringt dann die meiste Zeit mit einer

ganz neuen Baustelle. Für manche Probleme gibt es nämlich ästhetische Lösungen, deren Schönheit darin besteht, dass der Nutzer von ihnen nichts mitbekommt. Die Gewissheit, eine wirklich schöne Lösung gefunden und umgesetzt zu haben, muss als

Lohn für die Arbeit dann ausreichen. Wenn man die Welt dabei noch ein Stückchen sicherer gemacht hat, ist das befriedigend genug.

Das sichere Speichern von Zugangsdaten in einer Anwendung ist ein gutes Beispiel, das eine schöne Lösung verlangt. Auf die Existenz dieses Problems stieß ich natürlich nebenbei: Ein kleines Kommandozeilenwerkzeug sollte her, mit dem Kollegen und ich per HTTPS auf ein internes API im Verlag zugreifen können. Eigentlich keine große Sache: Meine neue Lieblingssprache Go bringt alles mit, um eine Kommandozeilenanwendung zu bauen und für alle gängigen Betriebssysteme zu kompilieren. Unter Windows, Linux und macOS sollte das Programm laufen, damit möglichst viele Kollegen in den Genuss kommen können. Die ersten Schritte waren auch schnell erledigt, die Hülle einer Kommandozeilenanwendung mit Go und dem CLI-Framework Cobra finden Sie ab Seite 142. Das Cross-Compiling erledigt Go voller Eleganz und schnell hatte ich auf dem Mac drei Binärdateien für die drei Betriebssysteme. Auch das Abrufen von Informationen aus einem HTTP-API ist mit den Bordmitteln von Go keine große Herausforderung.

Doch dann musste ich mich am API-Server anmelden, ganz klassisch per Benutzername und Kennwort. Irgendwie widerstrebte es mir, auch zu Testzwecken, mein Kennwort mal kurz in den Code einzubauen. Allzu schnell landen solche Experimente versehentlich in der Versionsverwaltung und dann schlimmstenfalls bei GitHub. Ein vermeidbarer Fehler. Eine

ästhetische Login-Routine musste her, bevor ich den Rest der eigentlichen Anwendung bauen konnte.

## Was andere machen

Das Rad neu zu erfinden, war in der Geschichte der Softwareentwicklung nur selten eine gute Idee. Also sah ich mir an, was andere getan haben. Das Docker-CLI zum Beispiel ist ausreichend weit verbreitet, um als Anhaltspunkt zu dienen. Ich dachte mir: „Was die gebaut haben, kann nicht ganz falsch sein.“ Die Anmeldung an einer Container-Registry funktioniert dort mit der Syntax:

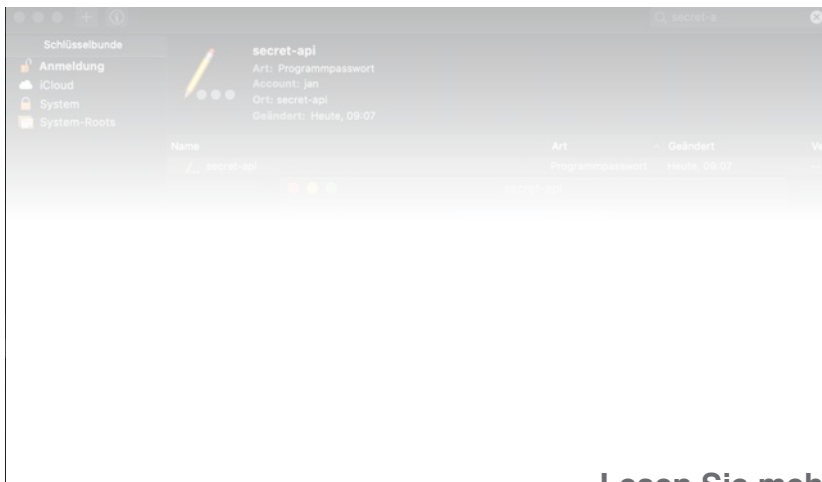
```
docker login <Adresse des Servers>
```

Danach wird man nach Benutzername und Kennwort gefragt. Hat die Anmeldung am Server funktioniert, muss man diesen Schritt auf der Maschine auch nicht mehr erledigen, die Daten werden also irgendwo gespeichert - doch wo eigentlich genau?

Um das herauszufinden, schaute ich mir die drei gängigsten Betriebssysteme einmal an, zunächst Linux mit Gnome Desktop und einen Ubuntu-Server ohne GUI. Fündig wurde ich jeweils im Ordner `.docker` im Home-Verzeichnis. Dort legt Docker die Datei `config.json` an:

```
"auths": {  
  "docker.pkg.github.com": {  
    "auth": <Name & Passwort in Base64>  
  }  
}
```

Für jede Registry, an der ich mich anmeldete, landete das Kennwort in dieser JSON-Datei, konvertiert als Base64-String. Base64 ist natürlich keine Verschlüsselung, sondern nur ein anderer Zeichensatz und leicht rückkonvertierbar. Nach dem sicherheitstechnisch letzten Schrei sah mir das noch nicht aus. Wer an die Datei kommt, hat das Kennwort. Und nicht



Lesen Sie mehr in c't Cloud 2021