

Alles backupen mit Rsnapshot

Folien + Skripte:

gitlab.com/dboehmer/clt2025

Daniel Böhmer ·  post@daniel-boehmer.de ·  [@dboehmer@ieji.de](https://twitter.com/dboehmer)

Ziel

- 1 Backup-Rechner
- Backup von sich selbst
- Backup von Servern/Geräten
- rotierende Backups

Wo beginnt „Backup“?

- RAID
- Logisches Backup
- Physisches Backup
 - Online lokal
 - Online woanders
 - offline

Installation

```
root@backup-rechner:/# apt install rsnapshot
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
The following additional packages will be installed:
```

```
bsd-mailx ca-certificates cron cron-daemon-common exim4-base exim4-config exim4-daemon-light krb5-locales  
libbsd0 libcbor0.8 libedit2 libevent-2.1-7 libfido2-1 libgdbm-compat4 libgdbm6 libgnutls-dane0 libgssapi-krb5-2  
libidn12 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblchown-perl liblockfile-bin liblockfile1  
libnsl2  
libperl5.36 libpopt0 libssl3 libtirpc-common libtirpc3 libunbound8 libx11-6 libx11-data libxau6 libxcb1  
libxdmcp6 libxext6 libxmuu1 logrotate netbase openssh-client openssl perl perl-modules-5.36 psmisc rsync  
sensible-utils xauth
```

```
Suggested packages:
```

```
anacron checksecurity exim4-doc-html | exim4-doc-info eximon4 file spf-tools-perl swaks gdbm-l10n dns-root-data  
krb5-doc krb5-user keychain libpam-ssh monkeysphere ssh-askpass perl-doc libterm-readline-gnu-perl  
| libterm-readline-perl-perl make libtap-harness-archive-perl openssl-server python3 python3-braceexpand
```

```
The following NEW packages will be installed:
```

```
bsd-mailx ca-certificates cron cron-daemon-common exim4-base exim4-config exim4-daemon-light krb5-locales  
libbsd0 libcbor0.8 libedit2 libevent-2.1-7 libfido2-1 libgdbm-compat4 libgdbm6 libgnutls-dane0 libgssapi-krb5-2  
libidn12 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblchown-perl liblockfile-bin liblockfile1  
libnsl2  
libperl5.36 libpopt0 libssl3 libtirpc-common libtirpc3 libunbound8 libx11-6 libx11-data libxau6 libxcb1  
libxdmcp6 libxext6 libxmuu1 logrotate netbase openssh-client openssl perl perl-modules-5.36 psmisc rsnapshot  
rsync sensible-utils xauth
```

```
0 upgraded, 50 newly installed, 0 to remove and 0 not upgraded.
```

```
Need to get 18.8 MB of archives.
```

```
After this operation, 80.8 MB of additional disk space will be used.
```

```
Do you want to continue? [Y/n]
```

Installation

```
root@backup-rechner:~# apt install rsnapshot
```

```
Reading package lists... Done
```

```
Building dependency tree... Done
```

```
Reading state information... Done
```

```
The following additional packages will be installed:
```

```
bsd-mailx ca-certificates cron cron-daemon-common exim4-base exim4-config exim4-daemon-light krb5-locales  
libbsd0 libcbor0.8 libedit2 libevent-2.1-7 libfido2-1 libgdbm-compat4 libgdbm6 libgnutls-dane0 libgssapi-krb5-2  
libidn12 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblchown-perl liblockfile-bin liblockfile1  
libnsl2  
libperl5.36 libpopt0 libssl3 libtirpc-common libtirpc3 libunbound8 libx11-6 libx11-data libxau6 libxcb1  
libxdmcp6 libxext6 libxmuu1 logrotate netbase openssh-client openssl perl perl-modules-5.36 psmisc rsync  
sensible-utils xauth
```

```
Suggested packages:
```

```
anacron checksecurity exim4-doc-html | exim4-doc-info eximon4 file spf-tools-perl swaks gdbm-l10n dns-root-data  
krb5-doc krb5-user keychain libpam-ssh monkeysphere ssh-askpass perl-doc libterm-readline-gnu-perl  
| libterm-readline-perl-perl make libtap-harness-archive-perl openssl-server python3 python3-braceexpand
```

```
The following NEW packages will be installed:
```

```
bsd-mailx ca-certificates cron cron-daemon-common exim4-base exim4-config exim4-daemon-light krb5-locales  
libbsd0 libcbor0.8 libedit2 libevent-2.1-7 libfido2-1 libgdbm-compat4 libgdbm6 libgnutls-dane0 libgssapi-krb5-2  
libidn12 libk5crypto3 libkeyutils1 libkrb5-3 libkrb5support0 liblchown-perl liblockfile-bin liblockfile1  
libnsl2  
libperl5.36 libpopt0 libssl3 libtirpc-common libtirpc3 libunbound8 libx11-6 libx11-data libxau6 libxcb1  
libxdmcp6 libxext6 libxmuu1 logrotate netbase openssh-client openssl perl perl-modules-5.36 psmisc rsnapshot  
rsync sensible-utils xauth
```

```
0 upgraded, 50 newly installed, 0 to remove and 0 not upgraded.
```

```
Need to get 18.8 MB of archives.
```

```
After this operation, 80.8 MB of additional disk space will be used.
```

```
Do you want to continue? [Y/n]
```

/etc/cron.d/rsnapshot

```
# This is a sample cron file for rsnapshot.
# The values used correspond to the examples in /etc/rsnapshot.conf.
# There you can also set the backup points and many other things.
#
# To activate this cron file you have to uncomment the lines below.
# Feel free to adapt it to your needs.

# 0 */4 * * * root /usr/bin/rsnapshot alpha
# 30 3 * * * root /usr/bin/rsnapshot beta
# 0 3 * * 1 root /usr/bin/rsnapshot gamma
# 30 2 1 * * root /usr/bin/rsnapshot delta
```

/etc/cron.d/rsnapshot

```
# This is a sample cron file for rsnapshot.  
# The values used correspond to the examples in /etc/rsnapshot.conf.  
# There you can also set the backup points and many other things.  
#  
# To activate this cron file you have to uncomment the lines below.  
# Feel free to adapt it to your needs.
```

```
0 */4 * * * root /usr/bin/rsnapshot hourly  
30 3 * * * root /usr/bin/rsnapshot daily  
0 3 * * 1 root /usr/bin/rsnapshot weekly  
30 2 1 * * root /usr/bin/rsnapshot monthly
```

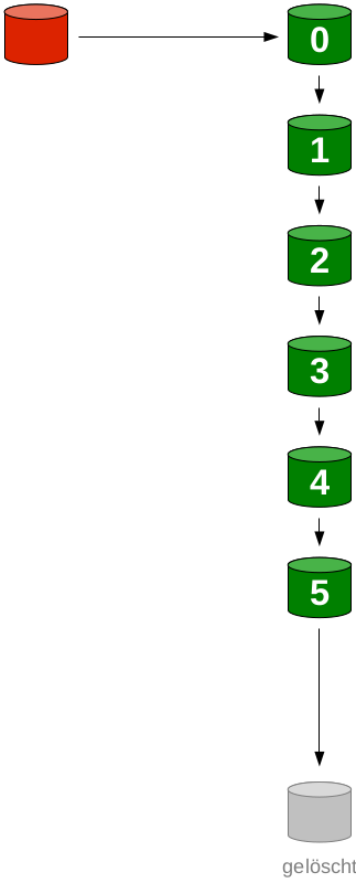
Original



Original

hourly

aller 4 Stunden



Original



hourly

aller 4 Stunden



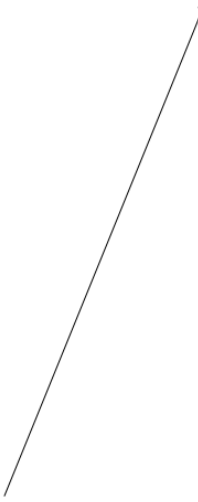
gelöscht

daily

jede Nacht



gelöscht



Original

hourly

aller 4 Stunden

daily

jede Nacht

weekly

jeden Sonntag



gelöscht



gelöscht



gelöscht

Original



hourly

aller 4 Stunden



gelöscht

daily

jede Nacht



gelöscht

weekly

jeden Sonntag



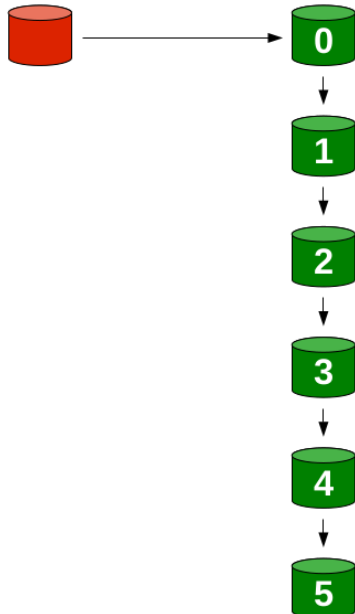
gelöscht

monthly

am Monatsanfang



gelöscht



Original



hourly

aller 4 Stunden



gelöscht

daily

jede Nacht



gelöscht

weekly

jeden Sonntag



gelöscht

monthly

am Monatsanfang



gelöscht

~125 Tage
>4 Monate

=

24 Stunden

+

7 Tage

+

4 Wochen

+

3 Monate

/etc/rsnapshot.conf

```
#####
# rsnapshot.conf - rsnapshot configuration file #
#####
#
# PLEASE BE AWARE OF THE FOLLOWING RULE:
#
# This file requires tabs between elements
#
#####

#####
# CONFIG FILE VERSION #
#####

config_version    1.2

#####
# SNAPSHOT ROOT DIRECTORY #
#####

# All snapshots will be stored under this root directory.
#
snapshot_root      /var/cache/rsnapshot/
```

/etc/rsnapshot.conf

```
#####  
# rsnapshot.conf - rsnapshot configuration file #  
#####  
#  
# PLEASE BE AWARE OF THE FOLLOWING RULE: #  
#  
# This file requires tabs between elements #  
#  
#####
```

```
#####  
# CONFIG FILE VERSION #  
#####
```

```
config_version 1.2
```

```
#####  
# SNAPSHOT ROOT DIRECTORY #  
#####
```

```
# All snapshots will be stored under this root directory.
```

```
#
```

```
snapshot_root /var/cache/rsnapshot/
```

/etc/rsnapshot.conf

Werkszustand in Debian:

```
retain    alpha 6  
retain    beta  7  
retain    gamma 4  
#retain    delta 3
```

Empfehlung:

```
retain    hourly    6  
retain    daily      7  
retain    weekly    4  
retain    monthly   3
```

Lokale Ordner

backup	/etc/	backup-rechner/
backup	/home/	backup-rechner/
backup	/root/	backup-rechner/
backup	/var/log/	backup-rechner/

```
root@backup-server:~# ls -dt /mnt/rsnapshot/*/backup-rechner/*
/mnt/rsnapshot/hourly.0/backup-rechner/etc
/mnt/rsnapshot/hourly.1/backup-rechner/etc
/mnt/rsnapshot/hourly.2/backup-rechner/etc
/mnt/rsnapshot/hourly.3/backup-rechner/etc
/mnt/rsnapshot/hourly.4/backup-rechner/etc
/mnt/rsnapshot/hourly.5/backup-rechner/etc
/mnt/rsnapshot/hourly.0/backup-rechner/home
/mnt/rsnapshot/hourly.1/backup-rechner/home
/mnt/rsnapshot/hourly.2/backup-rechner/home
...
```

Andere Server via SSH

backup	<code>root@server.example:/etc/</code>	<code>server/</code>
backup	<code>root@server.example:/home/</code>	<code>server/</code>
backup	<code>root@server.example:/root/</code>	<code>server/</code>
backup	<code>root@server.example:/var/log/</code>	<code>server/</code>

backup	<code>root@server.example:/srv/</code>	<code>server/</code>
backup	<code>root@server.example:/var/ftp/</code>	<code>server/</code>
backup	<code>root@server.example:/var/mail/</code>	<code>server/</code>
backup	<code>root@server.example:/var/www/</code>	<code>server/</code>

<code>backup_script</code>	<code>/usr/bin/ssh root@server '...'</code>	<code>server_skript/</code>
----------------------------	---	-----------------------------

Andere Server via SSH

```
root@backup-rechner:/# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zdflcs0qknxHZUfdneZi5VH7xkRoYMDFGgd0dC/G0wE root@3101f53ef777
The key's randomart image is:
+---[RSA 3072]---+
|                oE*B^|
|                . o+&O|
|                .@.X|
|                o  = &+|
|               S o o = O|
|                . o . = |
|               + o o  |
|                o o   |
|                   |
+-----[SHA256]-----+
```

Andere Server via SSH

```
ich@backup-rechner:~$ ssh root@server.example
root@server:/# mkdir /root/.ssh/
root@server:/# cat >> /root/.ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCqAUt3b21/veyvzyJ062CNWoj80CBfmFs/s7PBvR
KzLG0YVvnCKKzHS1WYGanxjWgRJ0hSXb0W+mo/955xjKQ6RgR6l1b1pHKKZZTNAztgkK4g4GCQJZyT
Uan5CR6iSoBAKkgdR3lju1AG25Ly+qAaD9d8XCjIUbMgYwQXuB0pwGqlQKm1t/EXw2ryPe3sT22d9T
QTEFZ1PQwLqFvz669/GEu4nljmtZcIxvkF+pVBtnE3bu7rT02zXqI6lFnZ0jdlcDkrRwsUlQjHY
juUGft4u4omyV+UdBtPuagERVL59tIYfliIDoB6+E7Q5FwhqlE4hfSUcFMW39stuxDcjF905LCUdu
Z4p805Bdj7fzCEAZX8u+vC/PpAk6CB+tZowbVlrZ95gKlaes3aTqNErtwV8r+0DAh0c60ozfUH8qh0
bWP52cmuDgcrdfISwSqmi1lf0mZVygyRYkL1d0oX0rfoqGiKFIMecDpcLhIsIDXX8r8MKnd9SbNCak5
WF9FU= root@backup-rechner
[Inhalt von id_rsa.pub einfügen und Strg+D drücken]
```

```
root@backup-rechner:/# ssh root@server.example rsync --version
rsync version 3.2.7 protocol version 32
```

Unwichtige Dateien ignorieren

temporäre Dateien

```
exclude      .cache
exclude      .directory
exclude      cache
exclude      temp
exclude      tmp
```

Foto-Vorschaubilder

```
exclude      /home/*/thumbnails/
```

individuelle Perl-Installationen mit `perlbrew`

```
exclude      /home/*/perl5/
```

PHP-Module in Projektordnern

```
exclude      /home/*/workspace/*/vendor/
```

Lösungen für Systeme
ohne SSH + rsync

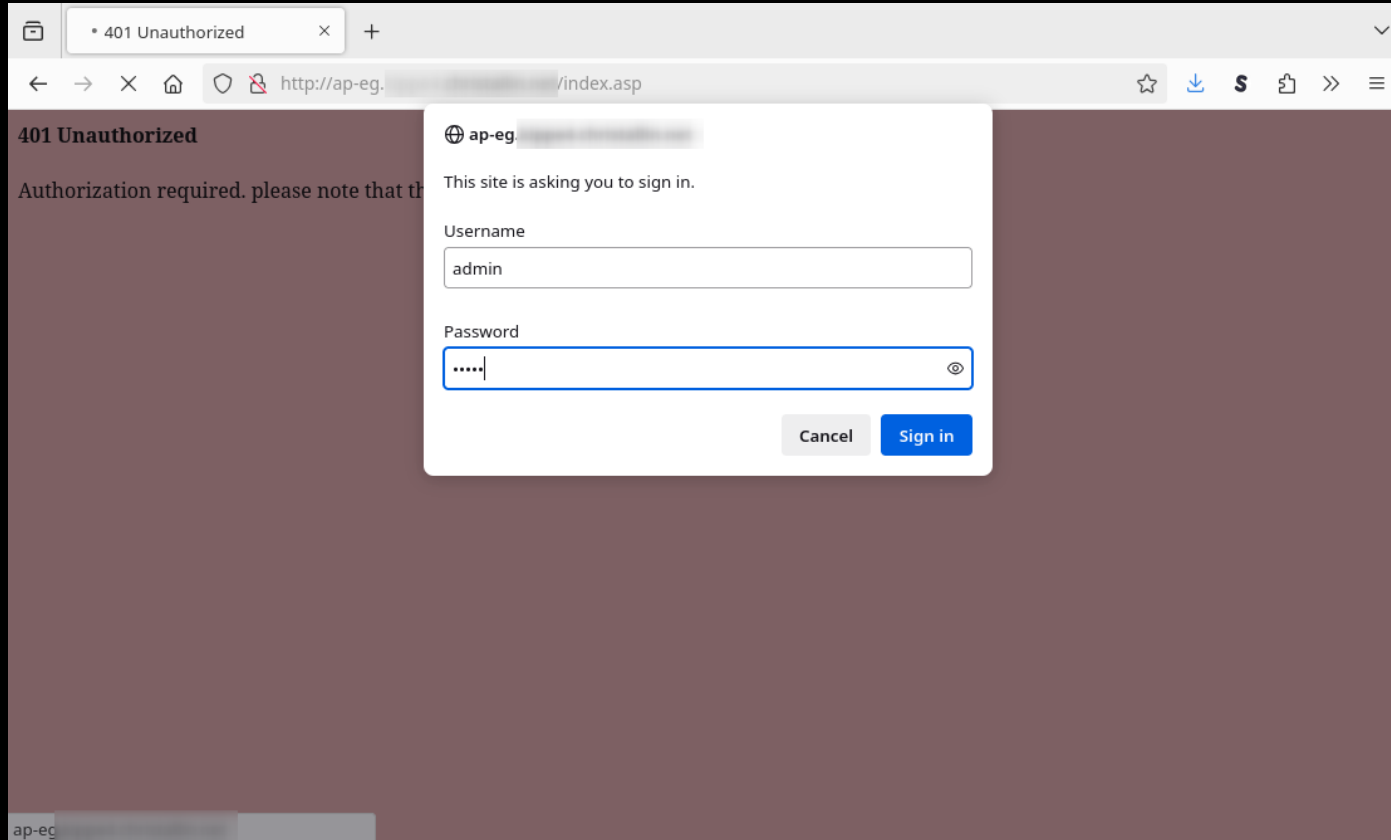
Backup von Open-WRT

```
# OpenWRT hat standardmäßig kein `rsync`
```

```
backup_script
```

```
/usr/bin/ssh root@openwrt.example 'cd / && tar c etc' | /usr/bin/tar x  
openwrt/
```

Backup von DD-WRT mit HTTP Basic Auth



Backup von DD-WRT mit HTTP Basic Auth

```
backup_script
```

```
/usr/bin/wget --netrc --quiet http://ddwrt.example/nvrambak.bin  
ddwrt/
```

```
root@server:~# cat /root/.netrc
```

```
machine ddwrt.example login admin password geheim
```

Backup von PostgreSQL als root

```
backup_script
```

```
/usr/bin/ssh root@server.example ↵
```

```
'cd / && su postgres -c "pg_dumpall --no-password"' > postgresql.sql  
server_postgresql/
```

Backup von MySQL/MariaDB als root

```
backup_script
```

```
/usr/bin/ssh root@server.example ↵
```

```
'mysqldump --all-databases --events --force --single-transaction' > mysql.sql  
server_mysql/
```

```
backup_script
```

```
/usr/bin/ssh root@server.example ↵
```

```
'mysqldump --all-databases --events --force --single-transaction | gzip' > mysql.sql.gz  
server_mysql/
```

Datenbank-Backup via SSH-Tunnel

Beispiel Webhosting bei Manitu.de:

```
backup_script
```

```
/opt/ssh_mysqldump.sh ssh12345@ngcobalt89.manitu.net mysql03.manitu.net:3306 db321 db456 ...  
webhosting_mysql/
```

Skript `ssh_mysqldump.sh` auf gitlab.com/dboehmer/clt2025

```
root@backup-rechner:~# cat /root/.my.cnf
```

```
[mysqldump_db321]
```

```
user=u12345
```

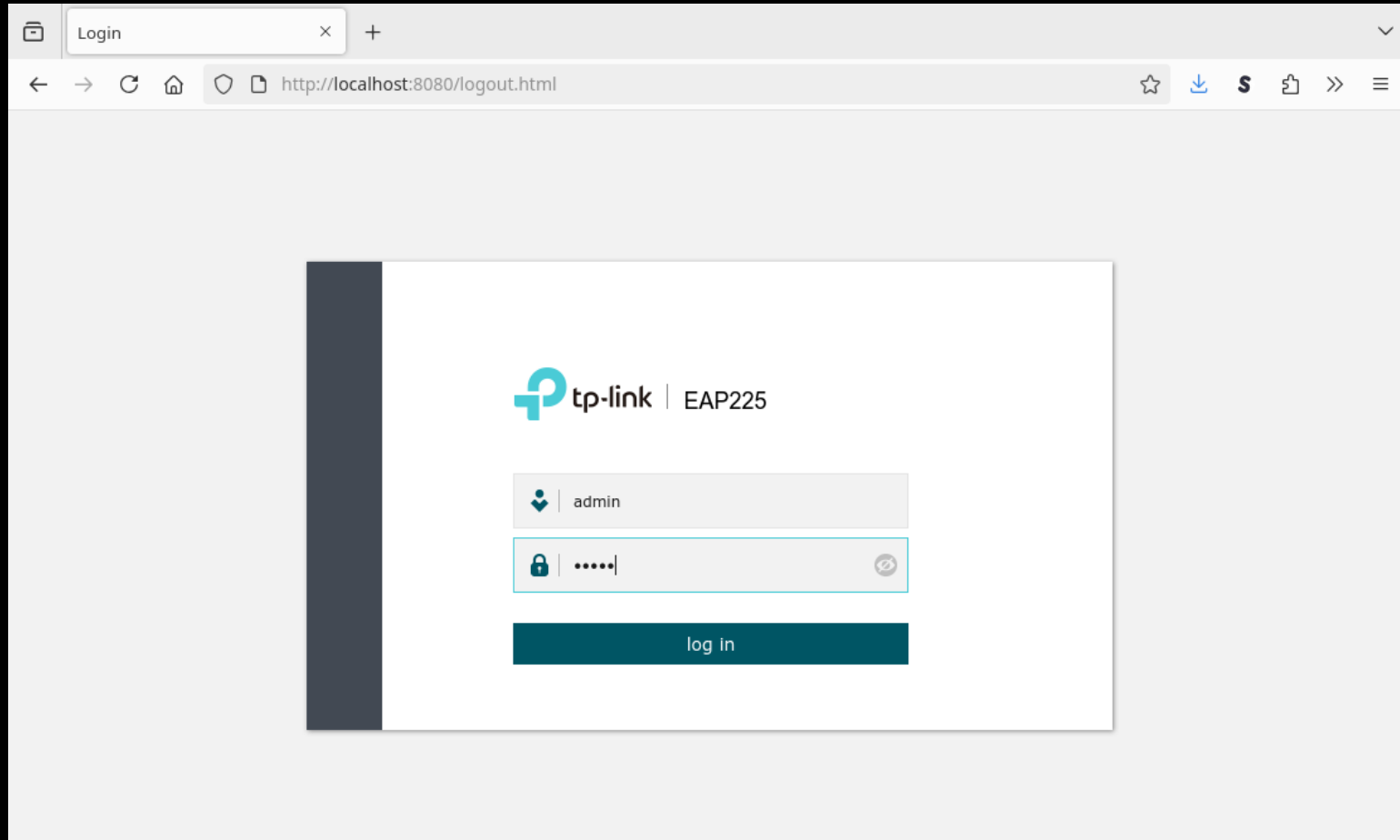
```
password=geheim
```

```
[mysqldump_db456]
```

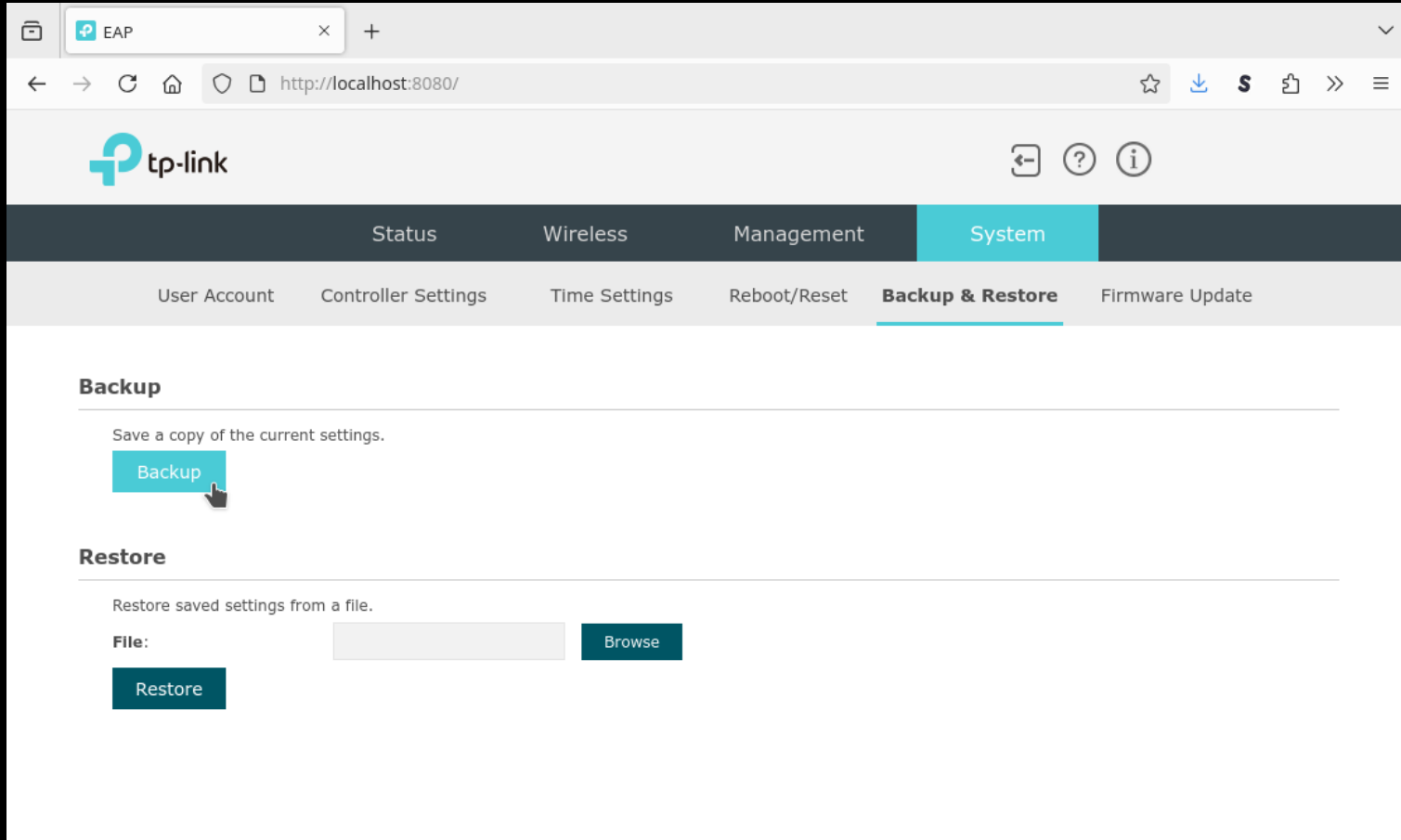
```
user=u67890
```

```
password=vertraulich
```

Backup von TP-Link EAP225



Backup von TP-Link EAP225



The screenshot shows the TP-Link EAP225 web interface in a browser window. The address bar shows `http://localhost:8080/`. The interface has a top navigation bar with the TP-Link logo and icons for help, search, and information. Below this is a main navigation menu with tabs: Status, Wireless, Management, System (highlighted), and a dropdown menu. The dropdown menu includes: User Account, Controller Settings, Time Settings, Reboot/Reset, Backup & Restore (highlighted), and Firmware Update.

Backup

Save a copy of the current settings.

[Backup](#)

Restore

Restore saved settings from a file.

File: [Browse](#)

[Restore](#)

Backup von TP-Link EAP225

```
root@backup-rechner:~# /opt/backup_tplink_eap225.pl  
USAGE: /opt/backup_tplink_eap225.pl [[BASE_URL] USERNAME] PASSWORD
```

Skript `backup_tplink_eap225.pl` auf gitlab.com/dboehmer/clt2025

Einzige Abhängigkeit: HTTP-Framework *Mojolicious* mit `Mojo::UserAgent`:

```
root@backup-rechner:~# apt install libmojolicious-perl
```

```

1  #!/usr/bin/env perl
2
3  use 5.36.0;
4
5  use Digest::MD5 'md5_hex';
6  use Mojo::UserAgent;
7
8  1 <= @ARGV <= 3
9    or die "USAGE: $0 [[BASE_URL] USERNAME] PASSWORD\n";
10
11 my $password = pop @ARGV;
12 my $username = pop @ARGV || 'admin';          # default username
13 my $base_url = pop @ARGV || 'http://192.168.0.254:80/'; # default IP
14
15 my $ua = Mojo::UserAgent->new;
16
17 my $tx1 = $ua->post(
18     $base_url,
19     form => {
20         username => $username,
21         password => uc( md5_hex($password) ),
22     },
23 );
24
25 # responds with HTTP 200 OK for wrong credentials, too
26 # -> instead check if session cookie is set
27 ( $tx1->res->headers->set_cookie // ' ' ) =~ m/ JSESSIONID= [0-9a-f]{30} ; /x
28 or die "Login failed: ", $tx1->res->to_string;
29
30 my $tx2 = $ua->get(
31     $base_url . 'data/configBackup.json?operation=backup',
32     { Referer => $base_url },      # actually checked by router UI
33 );
34
35 $tx2->res->code eq 200
36 and $tx2->res->headers->content_type eq 'x-bin/octet-stream; charset=utf-8'
37 or die "Download failed: ", $tx2->res->to_string;
38
39 print $tx2->res->body;

```

Backup von TP-Link EAP225

```
backup_script
```

```
/opt/backup_tplink_eap225.pl http://192.168.0.42/ admin geheim > config.bin  
tplink_eap225/
```

Backup von FRITZ!Box OS 6



FRITZ!Box 7330

Willkommen bei Ihrer FRITZ!Box

Melden Sie sich mit Ihrem Kennwort an.

Kennwort

[Kennwort vergessen?](#)

Anmelden

Backup von FRITZ!Box OS 6



FRITZ!Box 7330

FRITZ!NASMyFRITZ!

?

System > Sicherung

Sichern

Wiederherstellen

Neustart

Werkseinstellungen

Hier können Sie alle Einstellungen, die Sie in der FRITZ!Box vorgenommen haben, in einer Sicherungsdatei speichern. Die Einstellungen können in dieser FRITZ!Box oder in einer FRITZ!Box des gleichen Typs vollständig wiederhergestellt werden. In eine FRITZ!Box anderen Typs können ausgewählte Einstellungen aus der Sicherungsdatei übernommen werden.

Einstellungen sichern

Schützen Sie die Sicherungsdatei mit einem Kennwort.

Kennwort

geheim

kurz

Hinweis:
Bewahren Sie das Kennwort gut auf! Die Sicherungsdatei kann nur nach Eingabe des Kennwortes verwendet werden.

Sichern

Übersicht

Internet

Telefonie

Heimnetz

WLAN

DECT

Diagnose

System

Ereignisse

Energiemonitor

Push Service

Tasten und LEDs

FRITZ!Box-Benutzer

Sicherung

Update

Assistenten

Backup von FRITZ!Box OS 6

```
root@backup-rechner:~# /opt/backup_fritzbox_os6.pl  
USAGE: /opt/backup_fritzbox_os6.pl [BASE_URL] LOGIN_PASSWORD ENCRYPTION_PASSWORD
```

Skript `backup_fritzbox_os6.pl` auf gitlab.com/dboehmer/clt2025

Einzige Abhängigkeit: HTTP-Framework *Mojolicious* mit `Mojo::UserAgent`:

```
root@backup-rechner:~# apt install libmojolicious-perl
```

```

18 my $tx1 = $ua->get( $base_url . '/login_sid.lua' );
19
20 my $challenge = $tx1->res->dom->at('SessionInfo > Challenge')->text;
21 my $response = $challenge . '-'
22     . md5_hex( encode( 'UTF-16le', $challenge . '-' . $login_password ) );
23
24 my $tx2 = $ua->get( $base_url . '/login_sid.lua?user=&response=' . $response );
25
26 my $sid = $tx2->res->dom->at('SessionInfo > SID')->text;
27 $sid eq '0000000000000000' and die "Login failed: ", $tx2->res->to_string;
28
29 my $tx3 = $ua->post(
30     $base_url . '/cgi-bin/firmwarecfg',
31     multipart => [
32         {
33             'Content-Disposition' => 'form-data; name="sid"',
34             content                 => $sid,
35         },
36         {
37             'Content-Disposition' => 'form-data; name="ImportExportPassword"',
38             content                 => $encryption_password,
39         },
40         {
41             'Content-Disposition' => 'form-data; name="ConfigExport"',
42             content                 => '',
43         },
44     ],
45 );
46
47 $tx3->res->code eq 200
48     and $tx3->res->headers->content_type eq 'application/octet-stream;'
49     or die "Download failed: ", $tx3->req->to_string, $tx3->res->to_string;
50
51 print $tx3->res->body;

```

Backup von FRITZ!Box OS 6

```
backup_script
```

```
/opt/backup_fritzbox_os6.pl http://fritz.box/ geheim test > config.bin  
fritzbox/
```

Übertragungsrate limitieren

```
backup root@server:/mnt/data data/ +rsync_long_args=--bwlimit=1MiB
```

```
rsync_long_args --bwlimit=1MB
```

```
backup root@server1:/mnt/data data1/
backup root@server2:/mnt/data data2/
backup root@server3:/mnt/data data3/
```

```
ich@rechner:~$ pv /dev/cdrom0 > CD-Abbild.iso
443MiB 0:04:27 [1.67MiB/s] [=====] 63% ETA 0:02:34
```

```
root@backup-rechner:~# apt install pv
```

```
backup_script
/usr/bin/ssh root@server '...' | /usr/bin/pv --quiet --rate-limit 1m > data
server_data/
```

Backup-Prozess mit geringer Priorität

- `nice`: CPU-Priorität
- `ionice`: Lese-/Schreib-Priorität

```
0 */4 * * * root nice -n 19 ionice -c idle /usr/bin/rsnapshot hourly
30 3 * * * root nice -n 19 ionice -c idle /usr/bin/rsnapshot daily
0 3 * * 1 root nice -n 19 ionice -c idle /usr/bin/rsnapshot weekly
30 2 1 * * root nice -n 19 ionice -c idle /usr/bin/rsnapshot monthly
```

Fehlermeldungen

Cron <root@b[redacted]> nice -n 19 ionice -c idle /usr/bin/rsnapshot hourly 



From root@[redacted]

To root@[redacted]

Date 2024-09-19 8:07

 Summary  Headers

rsnapshot encountered an error! The program was invoked with these options:
/usr/bin/rsnapshot hourly

```
ERROR: /usr/bin/rsync returned 255 while processing root@[redacted]: /etc/
ERROR: /usr/bin/rsync returned 255 while processing root@[redacted]: /home/
ERROR: /usr/bin/rsync returned 255 while processing root@[redacted]: /root/
ERROR: /usr/bin/rsync returned 255 while processing root@[redacted]: /var/log/
ssh: Could not resolve hostname [redacted]: Name or service not known
kex_exchange_identification: Connection closed by remote host
Connection closed by UNKNOWN port 65535
ERROR: backup_script /usr/bin/ssh rsnapshot@[redacted] export > export && /usr/bin/ssh
rsnapshot@[redacted] export compact > export_compact returned 255
WARNING: Rolling back "l[redacted]_router/"
```

Weitere Ideen

- andere Systeme?
 - mobile Systeme
- SSH-Kommando in `authorized_keys` festnageln
- Btrfs-Snapshots
 - lokal
 - andere Server mit `send/receive`

Video-Empfehlungen vom 22. März 2025

- Universelles Offsite-Backup mit iSCSI
- Backups mit Restic und resticprofile

Weitere Ressourcen

- rsnapshot.org – kompakt
- github.com/rsnapshot/rsnapshot – Issues, PRs
- wiki.ubuntuusers.de/rsnapshot/ – ausführlich
- wiki.archlinux.de/title/Rsnapshot – kreativ
- it.werther-web.de/2011/10/23/migrate-rsnapshot-based-backup-to-btrfs-snapshots/ – btrfs
(nur via Web Archive)

Alles backupen mit Rsnapshot

Folien + Skripte:

gitlab.com/dboehmer/clt2025

Daniel Böhmer

 post@daniel-boehmer.de

 [@dboehmer@ieji.de](https://matrix.to/#/!dboehmer@ieji.de)