

6. FEBRUAR 2025

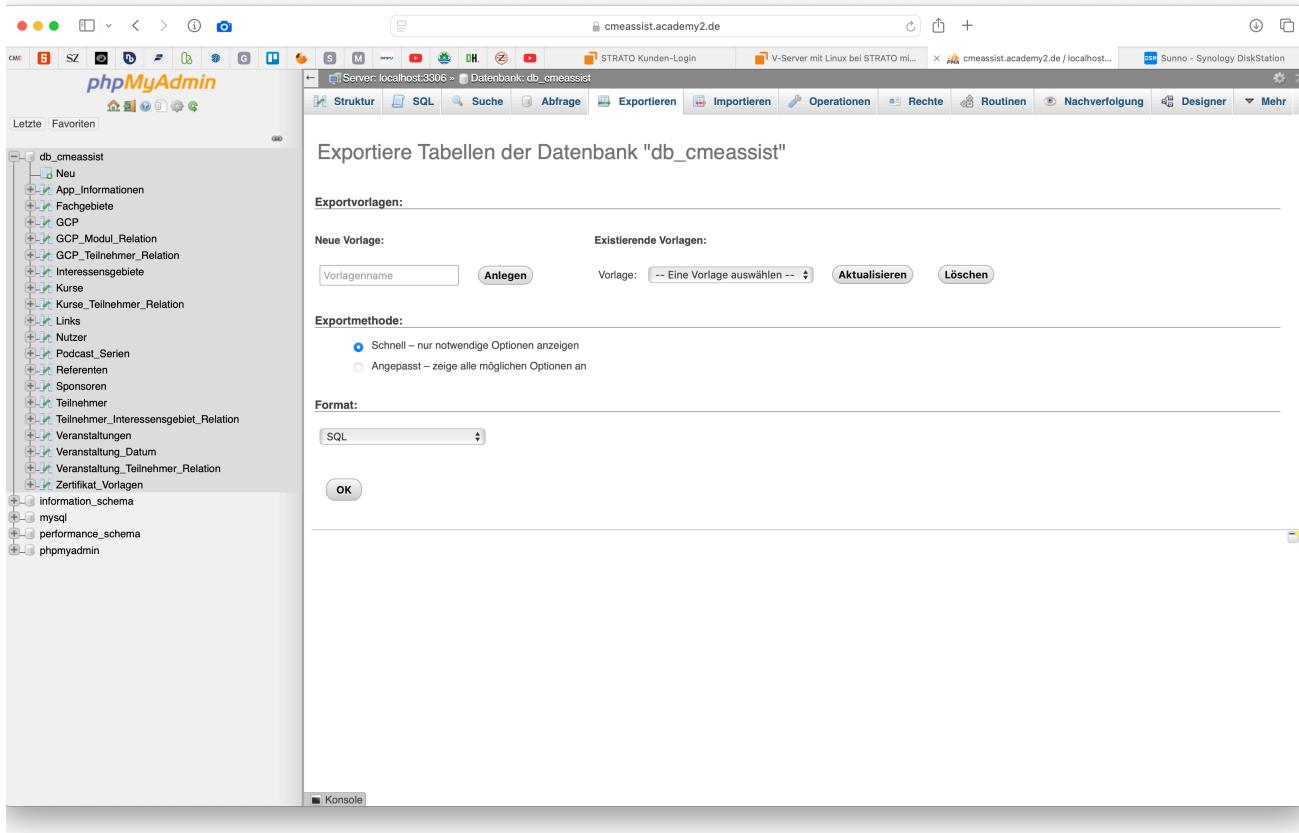
CMEASSIST

DATENBANK BACKUP AUTOMATISIEREN

Autor:in

DAVID MALINOWSKI

MYSQL DATENBANK BACKUP



BACKUP MANUELL

Ein Datenbank Backup kann manuell auf cmeassist mit PHP myAdmin erstellt werden.

Anmelden:

URL: <https://cmeassist.academy2.de/phpmyadmin/>

Benutzer: academy2

Passwort: admin@Z1rna

Datenbank db_cmeassist auswählen und den Reiter Exportieren drücken. Dann auf ok.

Fertig

BACKUP AUTOMATISIERT

CMEASSIST BACKUP

Eine automatisierte Datensicherung der Datenbank erfolgt in zwei Teilen.

1. Automatisierter Back-up auf cmeassist
2. Automatisches kopieren der Datenbanksicherung auf unsere NAS Sunno.

```
[david@h2979634:~$ cd /var/backups/db_backups/
[david@h2979634:/var/backups/db_backups$ ls -l
total 5856
-rw-r--r-- 1 root root 5993024 Feb  6 15:21 backup_db_cmeassist_06-02-2025.zip
[david@h2979634:/var/backups/db_backups$ sudo su
[lsudo] password for david:
[root@h2979634:/var/backups/db_backups# cd
[root@h2979634:/var/backups/db_backups# ls
[backupscript
[root@h2979634:/var/backups/db_backups# cd backupscript/
[root@h2979634:/var/backups/db_backups# ls -l
total 4
-rwxrwxrwx 1 root root 759 Feb  6 15:21 backup.sh
root@h2979634:/var/backups/db_backups# ]]
```

1. CMEASSIST

Auf der Konsole:

```
ssh
david@cmeassist.academy2.de
```

Passwort: admin@Z1rna

```
Last login: Thu Feb  6 16:45:09 on ttys000
[david@Mac ~ % ssh david@cmeassist.academy2.de
[david@cmeassist.academy2.de's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-1160.105.1.vz7.214.3 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
Last login: Thu Feb  6 14:45:51 2025 from 94.31.75.81
david@h2979634:~$ ]]
```

Mit cd + Enter kommt man automatisch zum root Verzeichnis.

Die Backups werden im folgenden Verzeichnis erstellt:

/var/backups/db_backups

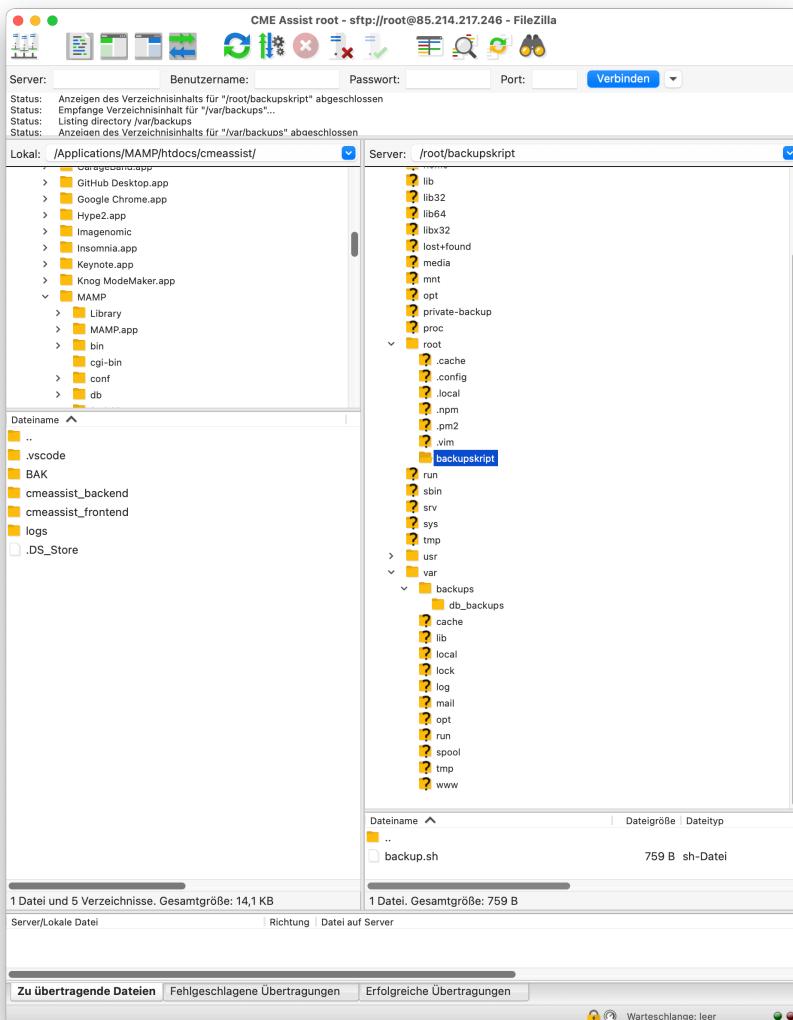
Hier schon ein erstes Backup gezippt (Screenshot):

```
[david@h2979634:~$ cd /var/backups/db_backups — ssh david@cmeassist.ac...
[david@h2979634:/var/backups/db_backups$ ls -l
total 5856
-rw-r--r-- 1 root root 5993024 Feb  6 15:21 backup_db_cmeassist_06-02-2025.zip
david@h2979634:/var/backups/db_backups$ ]]
```

Das Script zum Ausführen und zippen des Backups liegt hier:

/root/baskupscript/backup.sh

CMEASSIST BACKUP



Mit

```
cat backup.sh
```

Kann man einen Blick drauf werfen:

```
david — root@h2979634: ~/backupskript — ssh david@cmeassist.academy2.de — 104x36
[root@h2979634:~# ls
backupskript
[root@h2979634:~/backupskript# cd backupskript/
[root@h2979634:~/backupskript# ls -l
total 4
-rwxrwxrwx 1 root root 759 Feb  6 15:21 backup.sh
[root@h2979634:~/backupskript# cat backup.sh
#!/bin/bash

# MySQL Zugangsdaten
MYSQL_USER="root"
MYSQL_PASS="praktikant117@Z1rna"
MYSQL_DB="db_cmeassist"
BACKUP_DIR="/var/backups/db_backups"
DATE=$(date +"%d-%m-%Y")
BACKUP_FILE="$BACKUP_DIR/$MYSQL_DB_backup_$DATE.sql"
ZIP_FILE="$BACKUP_DIR/backup_db_cmeassist_$DATE.zip"
# Erstelle das Backup
mysqldump -u $MYSQL_USER -p$MYSQL_PASS $MYSQL_DB > $BACKUP_FILE

# Prüfen, ob das Backup erfolgreich war
if [ $? -eq 0 ]; then
    echo "Backup erfolgreich: $BACKUP_FILE"
    zip -j $ZIP_FILE $BACKUP_FILE
    if [ $? -eq 0 ]; then
        echo "Backup komprimiert: $ZIP_FILE"
        rm $BACKUP_FILE # Lösche die unkomprimierte Datei
    else
        echo "Fehler beim Komprimieren des Backups"
    fi
else
    echo "Fehler beim Erstellen des Backups"
fi
root@h2979634:~/backupskript#
```

Zur Erklärung des Skripts:

Zunächst werden diverse Variablen (GROSSBUCHSTABEN) definiert, die zum Ausführen des mysql Befehls notwendig sind. Zugriff auf die Variablen erfolgt mit \$NAME_DER_VARIABLEN

Die ersten drei Einträge sind der Zugang zur Datenbank. Eigentlich sollten es die gleichen Daten sein wie in der .env im cmeassist_backend. Aber der User academy2 hat nicht die Rechte für den Datenbank dump. Zumindest nicht über die Konsole.

Muss also root machen. Sollte vielleicht geändert werden...

DATE - ein Datum von heute wird generiert. Wird für den Namen des Backups benötigt, damit die Datei eindeutig zuzuordnen ist.

BACKUP_DIR - Verzeichnis in dem das Backup erstellt wird.

ZIP_FILE - Pfad und Name des gezippten Backups

Mit mysqldump + Parameter wird ein Datenbank-Dump angestoßen.

Die if-Schleife (von Chat GPT generiert) prüft ob ein Backup vorhanden ist. Falls ja, wird es gezippt und das Backup (backup_db_cmeassist_06-02-2025.sql) wird gelöscht.

```
# MySQL Zugangsdaten
MYSQL_USER="root"
MYSQL_PASS="praktikant117@Z1rna"
MYSQL_DB="db_cmeassist"
BACKUP_DIR="/var/backups/db_backups"
DATE=$(date +"%d-%m-%Y")
BACKUP_FILE="$BACKUP_DIR/$MYSQL_DB_backup_$DATE.sql"
ZIP_FILE="$BACKUP_DIR/backup_db_cmeassist_$DATE.zip"
# Erstelle das Backup
mysqldump -u $MYSQL_USER -p$MYSQL_PASS $MYSQL_DB > $BACKUP_FILE

# Prüfen, ob das Backup erfolgreich war

if [ $? -eq 0 ]; then
    echo "Backup erfolgreich: $BACKUP_FILE"
    zip -j $ZIP_FILE $BACKUP_FILE
    if [ $? -eq 0 ]; then
        echo "Backup komprimiert: $ZIP_FILE"
        rm $BACKUP_FILE # Lösche die unkomprimierte Datei
    else
        echo "Fehler beim Komprimieren des Backups"
    fi
else
    echo "Fehler beim Erstellen des Backups"
fi
```

Diese Datei backup.sh kann mit nano im Terminal editiert werden:

```
nano backup.sh
```



The screenshot shows a terminal window titled "david — root@h2979634: ~/backupskript — ssh david@cmeassist.academy2.de — 104x36". The title bar also displays "GNU nano 4.8" and the file name "backup.sh". The terminal content is a bash script for creating a MySQL backup. It defines variables for MySQL user, password, database, and backup directory, then uses mysql dump to create a SQL backup file. It then compresses this file into a ZIP file and removes the original SQL file if successful. The script ends with an if statement to check for errors. At the bottom of the screen, there is a status bar with various keyboard shortcuts for nano.

```
# MySQL Zugangsdaten
MYSQL_USER="root"
MYSQL_PASS="praktikant117@Zirna"
MYSQL_DB="db_cmeassist"
BACKUP_DIR="/var/backups/db_backups"
DATE=$(date +"%d-%m-%Y")
BACKUP_FILE="$BACKUP_DIR/$MYSQL_DB_backup_$DATE.sql"
ZIP_FILE="$BACKUP_DIR/backup_db_cmeassist_$DATE.zip"
# Erstelle das Backup
mysqldump -u $MYSQL_USER -p$MYSQL_PASS $MYSQL_DB > $BACKUP_FILE

# Prüfen, ob das Backup erfolgreich war

if [ $? -eq 0 ]; then
    echo "Backup erfolgreich: $BACKUP_FILE"
    zip -j $ZIP_FILE $BACKUP_FILE
    if [ $? -eq 0 ]; then
        echo "Backup komprimiert: $ZIP_FILE"
        rm $BACKUP_FILE # Lösche die unkomprimierte Datei
    else
        echo "Fehler beim Komprimieren des Backups"
    fi
else
    echo "Fehler beim Erstellen des Backups"
fi

[ Read 28 lines ]
^G Get Help      ^O Write Out   ^W Where Is   ^K Cut Text   ^J Justify   ^C Cur Pos   M-U Undo
^X Exit          ^R Read File   ^\ Replace    ^U Paste Text  ^T To Spell   ^_ Go To Line M-E Redo
```

Mit Strg+x wird nano wieder geschlossen. Mit y (yes) gespeichert und ENTER.

Mit ./ und der Name des Skripts (ohne Leerzeichen!) wird das Script ausgeführt:

```
./backup.sh
```

```
david — root@h2979634: ~/backupskript — ssh david@cmeassist.academy2.de —...  
[root@h2979634:~/backupskript# ./backup.sh  
Backup erfolgreich: /var/backups/db_backups/backup_db_cmeassist_06-02-2025.sql  
updating: backup_db_cmeassist_06-02-2025.sql (deflated 75%)  
Backup komprimiert: /var/backups/db_backups/backup_db_cmeassist_06-02-2025.zip  
root@h2979634:~/backupskript# ]
```

Jetzt muss das Skript täglich ausgeführt werden.

Das wird im sog. cronjob eingetragen:

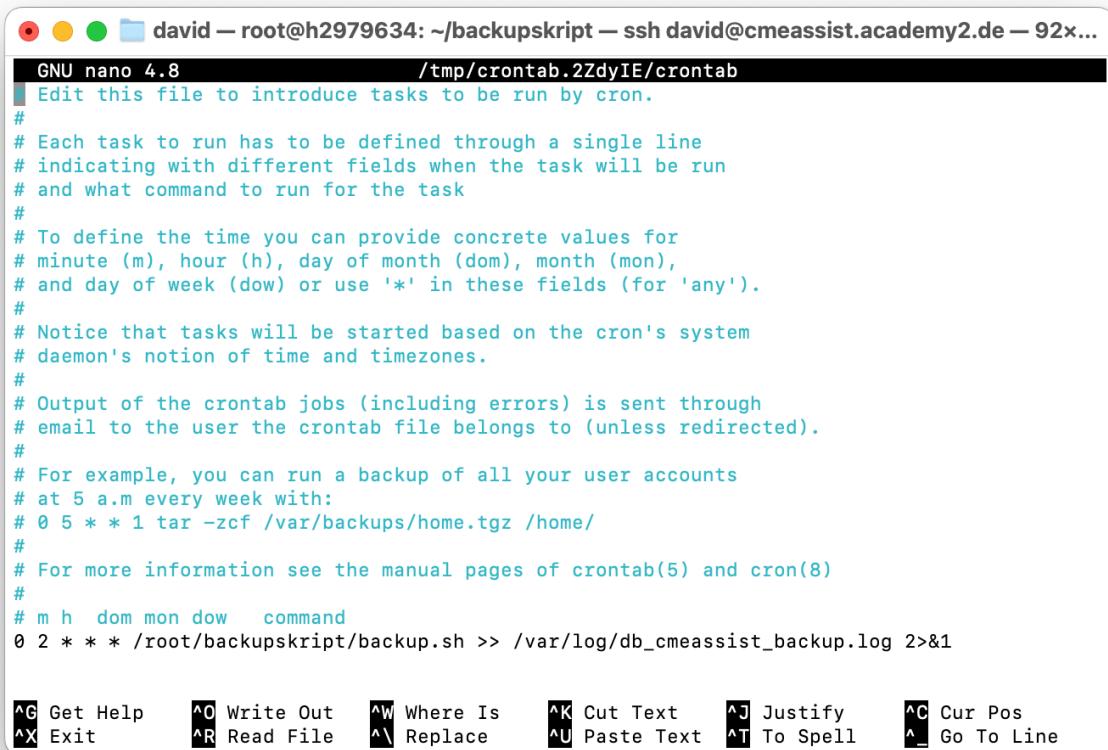
Mit

```
crontab -e
```

Kann ein Datum und das auszuführende Skript eingetragen werden:

```
0 2 * * * /root/backupskript/backup.sh >> /var/log/
db_cmeassist_backup.log 2>&1
```

02 *** - 2:00Uhr + Pfad zum Script >> Pfad zum Logfile db_cmeassist_backup.log das nach dem Task erzeugt wird.



The screenshot shows a terminal window titled "david — root@h2979634: ~/backupskript — ssh david@cmeassist.academy2.de — 92x...". The window displays the contents of the crontab file. The text is in light blue, and the terminal interface includes standard nano keybindings at the bottom.

```
GNU nano 4.8          /tmp/crontab.2ZdyIE/crontab
# Edit this file to introduce tasks to be run by cron.

#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
0 2 * * * /root/backupskript/backup.sh >> /var/log/db_cmeassist_backup.log 2>&1
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

Damit ist der Part auf cmeassist abgeschlossen. Jetzt muss das gezippte Backup auf Sunno kopiert werden.

SUNNO

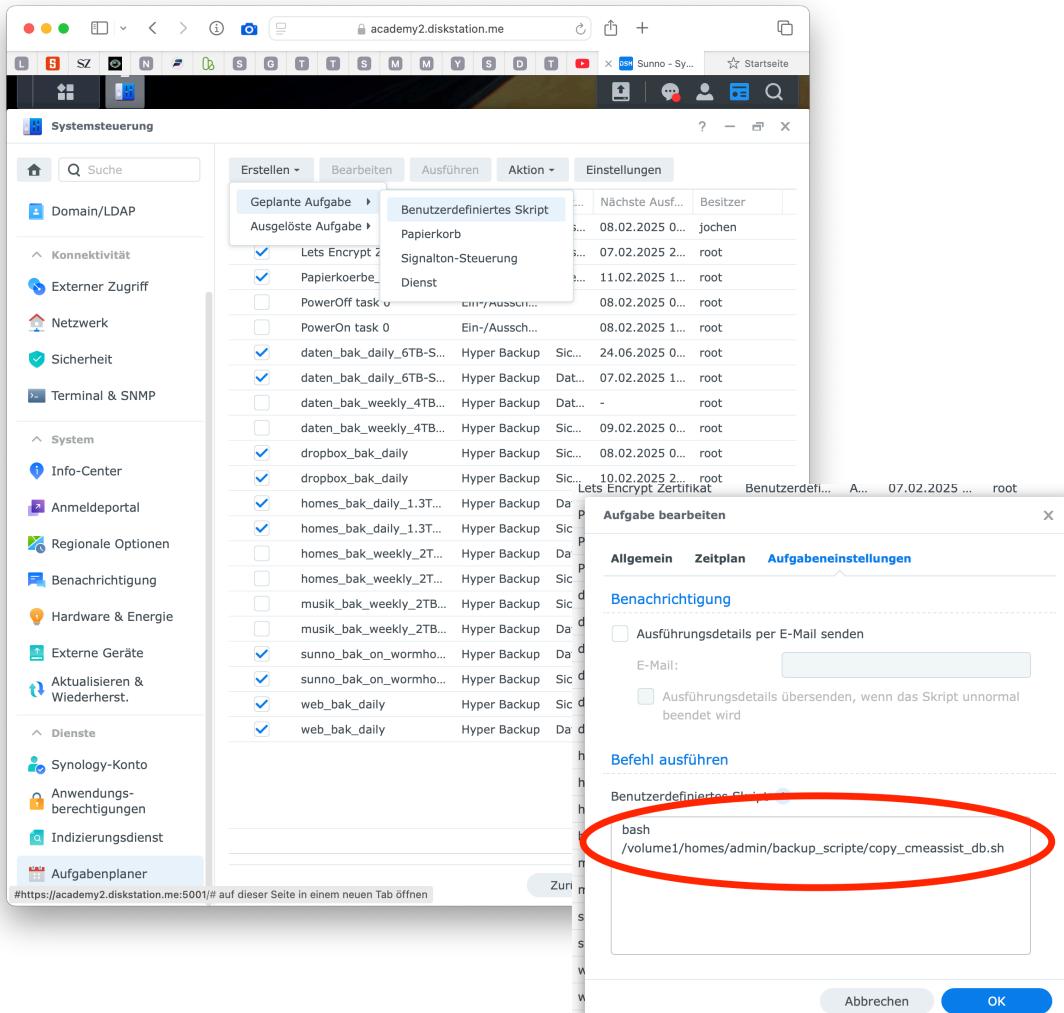
Auf Sunno muss eine **Aufgabe** erstellt werden. Diese Aufgabe führt terminiert ein Skript aus.

Auf Sunno anmelden: <https://academy2.diskstation.me:5001/>

Systemsteuerung->Aufgabenplaner->Erstellen->Geplante Aufgabe->Benutzerdefiniertes Script/Aufgabeneinstellungen

Hier wird bash und der Pfad zum Script eingetragen:

```
bash /volume1/homes/admin/backup_scripte/copy_cmeassist_db.sh
```



Das Skript sieht wie folgt aus:

```
david@BlackPro ~ % ping cmeassist.academy2.de
PING cmeassist.academy2.de (85.214.217.246): 56 data bytes
64 bytes from 85.214.217.246: icmp_seq=0 ttl=51 time=29.377 ms
64 bytes from 85.214.217.246: icmp_seq=1 ttl=51 time=28.284 ms
64 bytes from 85.214.217.246: icmp_seq=2 ttl=51 time=28.548 ms
64 bytes from 85.214.217.246: icmp_seq=3 ttl=51 time=28.739 ms
^C
--- cmeassist.academy2.de ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 28.284/28.737/29.377/0.403 ms
david@BlackPro ~ %

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

```
#!/bin/bash

# Variablen
REMOTE_USER="jochen"
REMOTE_HOST="85.214.217.246"
DATE=$(date +"%d-%m-%Y")
REMOTE_PATH="/var/backups/db_backups/backup_db_cmeassist_$DATE.zip"
LOCAL_PATH="/volume1/homes/admin/backup_db/backup_db_cmeassist_$DATE.zip"
PRIVATE_KEY="/volume1/homes/jochen/.ssh/id_rsa" # Falls du einen privaten
SSH-Schlüssel verwendest

# Archiv mit SCP kopieren
scp -i $PRIVATE_KEY $REMOTE_USER@$REMOTE_HOST:$REMOTE_PATH $LOCAL_PATH

# Überprüfen, ob das Kopieren erfolgreich war
if [ $? -eq 0 ]; then
    echo "Archiv erfolgreich kopiert!"
else
    echo "Fehler beim Kopieren des Archivs!"
fi
```

Skripterklärung:

Zunächst werden Variablen für den Zugriff auf den Server definiert:

REMOTE_USER: jochen oder david

REMOTE_HOST: die IP von cmeassist

Um die IP-Adresse des Zielservers herauszufinden, muss der Server mit Domainnamen angepingt werden. Die IP wird dann angezeigt.

DATE: Datum von heute. Das Format ist sehr wichtig, denn das identische Format wurde bei der Erstellung des gezippten Datenbankbackups verwendet! Nur so kann das Skript auf die richtige Datei auf cmeassist zugreifen.

REMOTE_PATH: Pfad zum gezippten Datenbankbackup auf cmeassist

LOCAL_PATH: Pfad auf sunno. Hier wird das gezippte Datenbankbackup kopiert

PRIVATE_KEY: Pfad auf Sunno zum PRIVATE_KEY

"/volume1/homes/jochen/.ssh/id_rsa"

Wie ein KEY erzeugt wird, wird im nächsten Kapitel detailliert beschrieben

Mit der folgenden Zeile wird mit scp das das gezippte Datenbankbackup von cmeassist auf Sunno kopiert:

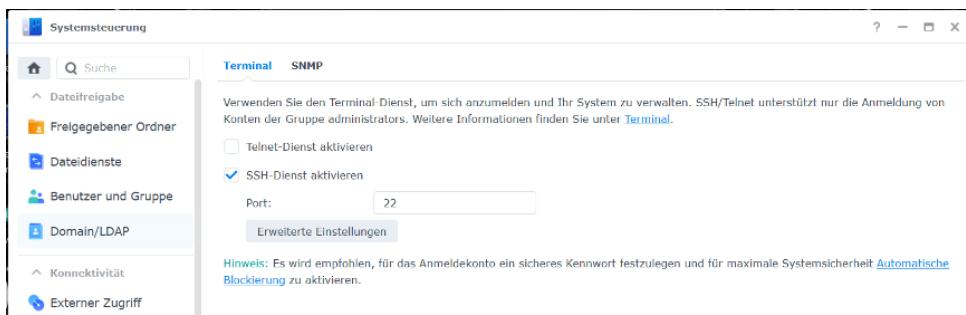
```
scp -i $PRIVATE_KEY $REMOTE_USER@$REMOTE_HOST:$REMOTE_PATH $LOCAL_PATH
```

Anschliessend wird eine Meldung über Vollzug oder Fehler rausgegeben.

PRIVATE_KEY ERZEUGEN

- Auf Sunno anmelden
- Systemsteuerung – Terminal & SMTP
- SSH-Dienst aktivieren

ACHTUNG! NACH DER PROZEDUR SSH WIEDER AUSSCHALTEN! SICHERHEIT!



Danach per ssh am Terminal auf Sunno anmelden

```
C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools>ssh jochen@academy2.diskstation.me
jochen@academy2.diskstation.me's password:

Synology strongly advises you not to run commands as the root user, who has
the highest privileges on the system. Doing so may cause major damages
to the system. Please note that if you choose to proceed, all consequences are
at your own risk.

jochen@Sunno:~$ |
```

Um einen privaten und einen öffentlichen Schlüssel über das Terminal zu erzeugen, kannst du das ssh-keygen-Werkzeug verwenden, das üblicherweise in Unix-basierten Betriebssystemen wie Linux und macOS vorinstalliert ist.

Schritt-für-Schritt Anleitung:

1. Terminal öffnen
2. Befehl zum Erzeugen des Schlüssels ausführen

```
ssh-keygen -t rsa -b 4096 -C "j.heilmann@academy2.de"
```

Hier die Erklärung zu den Parametern:

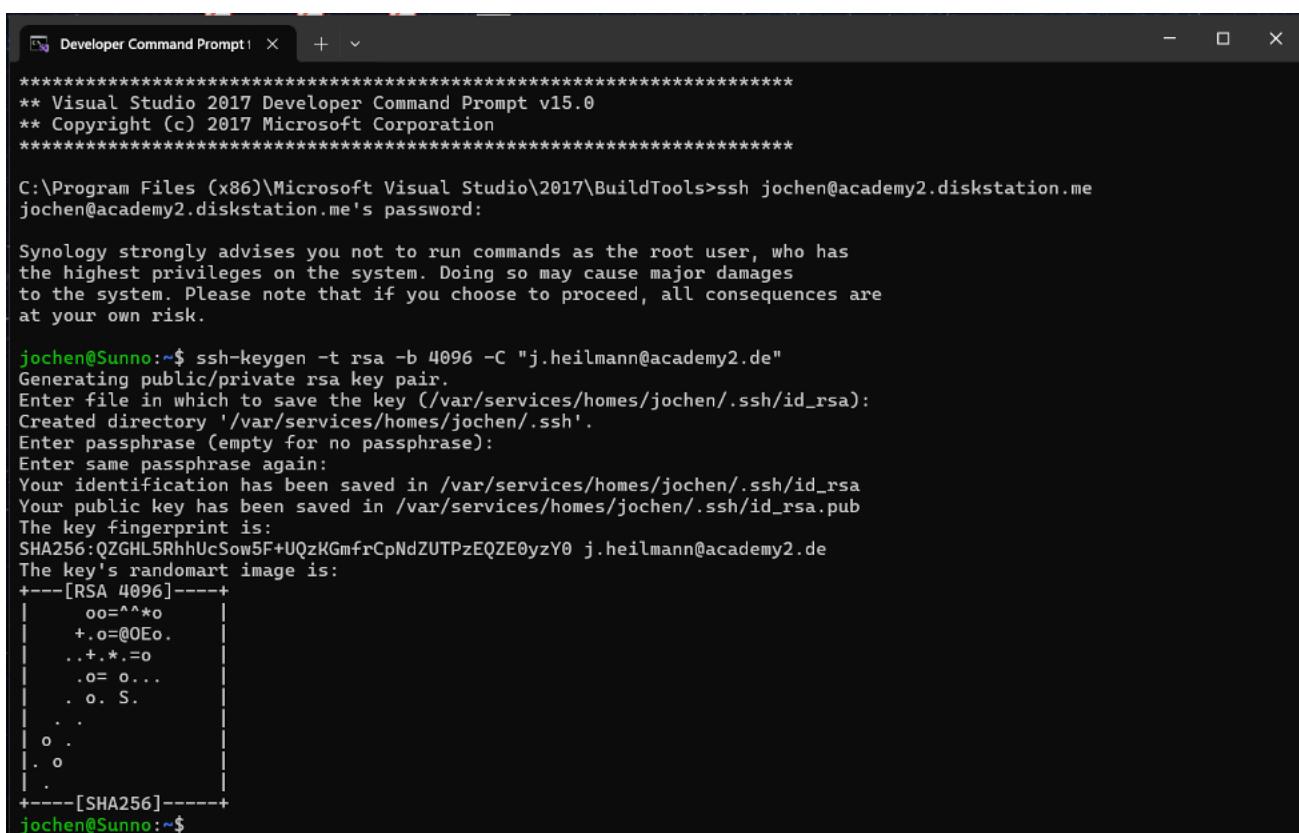
- -t rsa: Gibt an, dass der RSA-Algorithmus verwendet werden soll.
 - -b 4096: Setzt die Länge des Schlüssels auf 4096 Bit, was als sicher gilt.
 - -C "deine_email@example.com": Ein Kommentar (oft die eigene E-Mail-Adresse), der dem Schlüssel hinzugefügt wird, um ihn später besser identifizieren zu können.
3. Speicherort auswählen: Du wirst gefragt, wo der Schlüssel gespeichert werden soll. In der Regel kannst du einfach mit der Eingabetaste bestätigen, um den Standardpfad zu verwenden:
4. Enter file in which to save the key (/home/jochen/.ssh/id_rsa):

Wenn du den Standardpfad verwenden möchtest, drücke einfach Enter. Ansonsten gib einen neuen Pfad an.

Der Key sollte besser im admin Verzeichnis liegen und entsprechend der Pfad zum PRIVATE_KEY im Skript

Nach dem Erstellen des Schlüssels sollten diese Bestätigungen erscheinen:

- Your identification has been saved in /home/username/.ssh/id_rsa.
- Your public key has been saved in /home/username/.ssh/id_rsa.pub.



```

Developer Command Prompt x + v
=====
** Visual Studio 2017 Developer Command Prompt v15.0
** Copyright (c) 2017 Microsoft Corporation
=====

C:\Program Files (x86)\Microsoft Visual Studio\2017\BuildTools>ssh jochen@academy2.diskstation.me
jochen@academy2.diskstation.me's password:

Synology strongly advises you not to run commands as the root user, who has
the highest privileges on the system. Doing so may cause major damages
to the system. Please note that if you choose to proceed, all consequences are
at your own risk.

jochen@Sunno:~$ ssh-keygen -t rsa -b 4096 -C "j.heilmann@academy2.de"
Generating public/private rsa key pair.
Enter file in which to save the key (/var/services/homes/jochen/.ssh/id_rsa):
Created directory '/var/services/homes/jochen/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/services/homes/jochen/.ssh/id_rsa
Your public key has been saved in /var/services/homes/jochen/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:QZGHL5RhhUcSow5F+UQzKGmfrCpNdZUTPzEQZE0yzY0 j.heilmann@academy2.de
The key's randomart image is:
+---[RSA 4096]---+
| oo=^^*o |
| +..o=@0Eo. |
| ..+.*.=o |
| .o= o... |
| . o. S. |
| . . . |
| . o . |
+---[SHA256]---+
jochen@Sunno:~$
```

Der private Schlüssel befindet sich nun unter /home/jochen/.ssh/id_rsa und der öffentliche Schlüssel unter /home/jochen/.ssh/id_rsa.pub.

Die Schlüsseldateien:

- Privater Schlüssel (id_rsa): Dieser Schlüssel sollte niemals weitergegeben werden und ist zum Authentifizieren deiner Identität auf entfernten Systemen.
- Öffentlicher Schlüssel (id_rsa.pub): Dieser kann auf Servern hinterlegt werden, bei denen du dich mit deinem privaten Schlüssel authentifizieren möchtest.

DEN ÖFFENTLICHEN SCHLÜSSEL AUF EINEN SERVER KOPIEREN

Den öffentlichen Schlüssel zu einem entfernten Server per ssh-copy-id oder manuell hinzufügen.

MIT SSH-COPY-ID:

```
ssh-copy-id username@hostname
```

Mit dem Befehl, wird er public Key auf den Server (hier cmeassist.academy2.de) im Userverzeichnis des Angemeldeten Users ins `./.ssh/authorized_keys` kopiert. Bzw. in die Datei `authorized_keys` eingefügt.

MANUELL

Den Inhalt der id_rsa.pub in die Datei `~/.ssh/authorized_keys` auf dem Server einfügen:

Die erzeugte Schlüsseldatei id_rsa.pub mit dem cat `~/.ssh/id_rsa.pub` Befehl anzeigen



```
Git CMD - ssh jochen@academy2.diskstation.me
jochen@Sunno:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQCrSB+vgqy11D75E0YZIFH0mfNmpucs dW0+vHYIQNX0hWu9ncdh1v
Z01v1k0wx13JkNRGS138oMG0XDHuSUfhz849R/hcVuTSLw2AhXPvSI8N1o/n1HE/a0J+PhfSE+drLxcIzEgadH1wKJ
wjDRZZNL24oCJyW8baIWNFnjDbQcgK9HqUiOd7dAYoVUkYRK8x07L1GxuHBTjsBtC+y7J7dHv2oAf3CAJ0Luu5H/5g
taXBXnmO8njv31pPASSpEe96u1xBae/5fZ2hRJBz7hg5SJU5TFSh0piTIRhgsi2t/g1TdtFQ43XXZD7oM4e1Zpk01y
S3HkXH+3N1LE9EUk+Qb7wgvU49ngIg15cz+rRznUtfSd0gE97QvbgzKLLIFW7dKYIr8wKxqQk2N81RdqFNNuAZDVC
B2HDE9N8dqdhXVcgmv0w5ZwTx1VSKKPAjNT0A1fNz7dzQXpn4cQcFiXNmj0FQT644rIBjbUX1FUEhd1WRD69LZV8
Jp3MB0TZPUxPo15ZM1F0wQRX1Y21gs1uj+oFSwfGkAEpYRSGhGJe1Xsh6wvyQD8Bw+x3L1o43GaHrr3jpNuhWaN14L
Tw1zzQW13AAWvdfk7tD2ts+z8PzENCw5jeWKxbv1ogKCjbmNeaXZ04rjv7CxAC8tdzihycsfKwsXddqruMnPrFvQE2
sw== j.heilmann@academy2.de
jochen@Sunno:~$
```

```
Auswählen Git CMD - ssh jochen@academy2.diskstation.me
jochen@Sunno:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACrSB+vgqy1ID75E0YZIFH0mfNmPucsdw0+vHYIQNX0hWu9ncdh1v
Z01v1kOwx13JkNRGS138oMG0XDHuUFhz849R/hcVuTSLw2AhXPvSI8N1o/n1HE/a0J+PhfSE+drLxcIzEgadHlwKJ
wjDRzzNL24oCJy8baIWNFnjDbQcgk9HqUiOd7dAYoVUKYRK8x07LiGxuHBTjsBtC+y7J7dHv2oAf3CAJ0Luu5H/5g
taXBXnmO8njv3IpPASSpEe96u1xBae/5fZ2hRBz7hg5SJu5TFSh0piTirhgsi2t/g1TdtFQ43XxD7oM4e1Zpk01y
S3HkXH+3N1LE9Euk+Qb7wgvU49ngIg15cz+rRznUtfSd0gE97QvbzgkLLIFW7dKYIR8w8KxqQk2N81RdqFNNUAZDVC
B2HDE9N8dqdhXVcmgv0w5ZWTx1VSKKPAjNt0A1fNz7dzQXpn4cQcFiXNmj0FQT644rIBjbUXIFUEhdIWRD69LZV8
Jp3MB0TZPUxPo15ZM1FOWQRX1Y21gs1uj+oFSwfGkAEpYRSGhGJe1Xsh6wvyQD8Bw+x3L1043GaHrr3jpNuhWaN14L
Tw1zzQW13AAWvd fk7td2ts+z8PzENCw5jeWKxbv10gKCjbmNeaXZ04rjv7CxAc8tdzihycsfKwsXddqruMnPrFvQE2
sw== j.heilmann@academy2.de
jochen@Sunno:~$
```

Den Schlüssel mit der linken Maustaste markieren (der Schlüssel wird dadurch auch gleich kopiert)

```
Neues Textdokument.txt - Editor
Datei Bearbeiten Format Ansicht Hilfe
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACrSB+vgqy1ID75E0YZIFH0mfNmPucsdw0+vHYIQNX0hWu9ncdh1v
+N8tqqcbimGwFlST297vx901saDowVbUuWZ7SSca+e1+IROHOURdaevenjJR3Rc2Vu0cmBLTK+iV5C8csDSHBNsUo0A5JPZ/lzZKwcq1KeuoTNc0LD+cXmXVF
+3Coi28IoPwahe3zY79on91WE6L5n31sMI+c9Y2ck0fbxKLd17rAVq8fYXvGkZvgRvX4m/QkrZ4o8+MyYQ9iGRKxDku8Nb3KUj/3YH1marZel1sm1z+CytjcOGmbp4KB
+7/xmBp9p2y0bu0/sy7ep1mxpBjkwqoy79s0Ws88/YjPrzawTKwl7lcyX1fqXCAThPxjwNqatKdTzS68d1u47+vm7Q0Cn5Cnqz806r0Vd6iTwxt
+lj/8wB6rvB3iahntaHVF2Tg05/gu/0mZr60MeMYH11QWNUmhVJojqjBH825hvWevJILRWwuhLG9zc3FG1cALU+IN66Ez1txQ1vv0rZkq96LS4+d67M
+ICBooC8e5V84MdeTJFgtZ50qSzsnc0txNvlyQbw2nX4+5CxWymUwkAEGBflimiGPxyQYwgDQ00jo6jh00M66NM1NYXnw== j.heilmann@academy2.de

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
Last login: Fri Feb 7 12:44:29 2025 from 109.250.179.181
jochen@h2979634:~$
```

Vorsichtshalber den Schlüssel auf einen txt Datei zwischenspeichern

```
iochen@h2979634:~/ssh
jochen@h2979634:~/ssh
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACrSB+vgqy1ID75E0YZIFH0mfNmPucsdw0+vHYIQNX0hWu9ncdh1v>
```

Danach auf cmeassist.academy2.de per ssh am Terminal anmelden

Mit cd ~/ssh/ ins ssh Verzeichnis wechseln und mit nano authorized_keys die Datei öffnen

Den zuvor kopierten Key einfügen und speichern.

Der private und der public key liegen zur Zeit jeweils im Homeverzeichnis von Jochen!

