

USV Zweiten Ubuntu-Server als NUT-Client einrichten

Ziel

Der zweite Ubuntu-Server wird als NUT-Netzwerkclient eingerichtet.

Er besitzt keine direkte USB-Verbindung zur USV. Stattdessen verbindet er sich über das Netzwerk mit dem ersten Ubuntu-Server, der als NUT-Master beziehungsweise NUT-Primary arbeitet.

Der NUT-Client übernimmt folgende Aufgaben:

- * USV-Status vom NUT-Master abrufen
- * Stromausfälle erkennen
- * Kritischen Batteriestatus erkennen
- * Den zweiten Server kontrolliert herunterfahren
- * Shutdown-Anweisungen des NUT-Masters verarbeiten

Beispielkonfiguration

System	IP-Adresse	Aufgabe
NUT-Master	192 . 168 . 1 . 20	Direkt per USB mit der USV verbunden
NUT-Client	192 . 168 . 1 . 21	Zweiter geschützter Ubuntu-Server
USV-Name	apc	Name der USV in /etc/nut/ups.conf

<note important>

Die Beispiel-IP-Adressen müssen durch die tatsächlichen IP-Adressen der Server ersetzt werden.

</note>

Voraussetzungen

- * NUT-Master ist bereits eingerichtet
- * USV wird auf dem Master korrekt erkannt
- * NUT-Server läuft auf dem Master
- * TCP-Port 3493 ist vom Client erreichbar
- * Der Client besitzt eine feste IP-Adresse
- * Root- oder sudo-Berechtigung auf beiden Servern

1. USV-Name auf dem Master prüfen

Auf dem NUT-Master den Namen der konfigurierten USV anzeigen:

```
\\  
upsc -l localhost\\
```

Erwartete Ausgabe:

```
\\apc
```

Der Name `apc` stammt aus dem Abschnitt in der Datei `/etc/nut/ups.conf`:

```
[apc]  
driver = usbhid-\\ups  
port = \\auto  
vendorid = \\051d  
desc = "APC Smart-UPS SMC1500IC"
```

Der Name in den eckigen Klammern muss später auch auf dem Client verwendet werden.

2. Clientbenutzer auf dem Master anlegen

Auf dem NUT-Master die Datei öffnen:

```
sudo nano /etc/nut/upsd.users
```

Ubuntu 22.04 beziehungsweise NUT 2.7

Folgenden Benutzer eintragen:

```
[monclient]  
password = \\CLIENT_PASSWORT_HIER_EINTRAGEN  
upsmon \\slave
```

Ubuntu 24.04 beziehungsweise NUT 2.8

Folgenden Benutzer eintragen:

```
[monclient]  
password = \\CLIENT_PASSWORT_HIER_EINTRAGEN  
upsmon \\secondary
```

Ein sicheres Passwort kann mit folgendem Befehl erzeugt werden:

```
openssl rand -hex \\24
```

<note warning> Das Passwort muss später auf dem Client exakt gleich in `/etc/nut/upsmon.conf` eingetragen werden.

Das Passwort sollte nicht in einer öffentlich zugänglichen Dokumentation gespeichert werden.
</note>

Nach der Änderung den NUT-Server neu starten:

```
sudo systemctl restart nut-\server
```

Status prüfen:

```
sudo systemctl status nut-server --no-pager
```

3. Netzwerkzugriff auf dem Master konfigurieren

Auf dem Master die Datei öffnen:

```
sudo nano /etc/nut/upsd.conf
```

Beispielkonfiguration:

```
LISTEN 127.0.0.1 \3493  
LISTEN 192.168.1.20 \3493
```

Dabei ist 192.168.1.20 die LAN-IP-Adresse des NUT-Masters.

Nach der Änderung den Dienst neu starten:

```
sudo systemctl restart nut-\server
```

Prüfen, ob NUT auf Port 3493 lauscht:

```
sudo ss -ltnp | grep \3493
```

Eine mögliche Ausgabe lautet:

```
LISTEN 0 16 127.0.0.1:\3493  
LISTEN 0 16 192.168.1.20:\3493
```

4. Firewall auf dem Master konfigurieren

Falls UFW aktiviert ist, sollte nur der zweite Server Zugriff auf den NUT-Port erhalten.

Beispiel:

```
sudo ufw allow from 192.168.1.21 to any port 3493 proto \tcp
```

Firewallstatus prüfen:

```
sudo ufw \status
```

Eine mögliche Regel lautet:

```
3493/tcp ALLOW 192.168.1.\21
```

<note> Die Firewallregel wird auf dem NUT-Master eingerichtet, nicht auf dem Client. </note>

5. Netzwerkverbindung vom Client zum Master prüfen

Auf dem zweiten Server prüfen, ob der Master erreichbar ist:

```
ping -c 4 192.168.1.\20
```

Danach prüfen, ob der NUT-Port erreichbar ist:

```
nc -zv 192.168.1.20 \3493
```

Falls nc nicht installiert ist:

```
sudo apt install netcat-\openbsd
```

Bei erfolgreicher Verbindung erscheint beispielsweise:

```
Connection to 192.168.1.20 3493 port [tcp/nut] succeeded!
```

6. NUT-Client installieren

Auf dem zweiten Ubuntu-Server die Paketquellen aktualisieren:

```
sudo apt \update
```

NUT-Client installieren:

```
sudo apt install nut-\client
```

Installierte Version anzeigen:

```
dpkg-query -W -f='${Version}\n' nut-\client
```

Alternativ:

```
apt-cache policy nut-\client
```

7. Verbindung zum NUT-Master testen

Auf dem Client den USV-Status vom Master abfragen:

```
upsc apc@192.168.1.\20
```

Bei erfolgreicher Verbindung werden zahlreiche Werte angezeigt.

Beispiel:

```
battery.charge: \100
battery.runtime: \2400
device.mfr: American Power \Conversion
device.model: Smart-UPS C \1500
ups.load: \25
ups.status: \OL
```

Wichtige Werte:

Wert	Bedeutung
battery.charge	Ladezustand der Batterie in Prozent
battery.runtime	Geschätzte Restlaufzeit in Sekunden
ups.load	Auslastung der USV in Prozent
ups.status	Aktueller Status der USV
ups.model	Erkanntes USV-Modell

8. Client-Betriebsart konfigurieren

Auf dem zweiten Server die Datei öffnen:

```
\\
sudo nano /etc/nut/nut.conf\\
```

Folgenden Inhalt eintragen:

```
MODE=\netclient
```

<note> Bei `MODE=netclient` wird kein lokaler USV-Treiber gestartet.

Der Server überwacht ausschließlich eine entfernte USV über das Netzwerk. </note>

Beim Eintrag dürfen keine Leerzeichen um das Gleichheitszeichen stehen.

Richtig:

```
MODE=\netclient
```

Falsch:

```
MODE = \netclient
```

9. Vorhandene Clientkonfiguration sichern

Vor der Änderung sollte die Datei `upsmon.conf` gesichert werden:

```
sudo cp -a /etc/nut/upsmon.conf "/etc/nut/upsmon.conf.backup.$(date +%F-%H%M%S)"
```

Eine mögliche Sicherungsdatei lautet:

```
/etc/nut/upsmon.conf.backup.2026-07-19-120000
```

10. NUT-Monitor auf dem Client konfigurieren

Die Datei öffnen:

```
sudo nano /etc/nut/upsmon.conf
```

Vorhandene aktive MONITOR-Zeilen auskommentieren oder entfernen.

Ubuntu 22.04 beziehungsweise NUT 2.7

Folgende Konfiguration verwenden:

```
MONITOR apc@192.168.1.20 1 monclient CLIENT_PASSWORT_HIER_EINTRAGEN slave
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
FINALDELAY 5
```

Ubuntu 24.04 beziehungsweise NUT 2.8

Folgende Konfiguration verwenden:

```
MONITOR apc@192.168.1.20 1 monclient CLIENT_PASSWORT_HIER_EINTRAGEN
secondary
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
```

FINALDELAY \5

Das Passwort muss exakt mit dem Passwort des Benutzers `monclient` auf dem Master übereinstimmen.

11. Bedeutung der Clientkonfiguration

Einstellung	Bedeutung
<code>apc@192.168.1.20</code>	USV <code>apc</code> auf dem NUT-Master
<code>1</code>	Power-Wert des überwachten Stromsystems
<code>monclient</code>	Benutzername aus <code>upsd.users</code> auf dem Master
<code>CLIENT_PASSWORT_HIER_EINTRAGEN</code>	Passwort des Clientbenutzers
<code>slave</code>	Bezeichnung für einen Client unter NUT 2.7
<code>secondary</code>	Bezeichnung für einen Client unter NUT 2.8
<code>MINSUPPLIES 1</code>	Mindestens eine Stromversorgung muss verfügbar sein
<code>SHUTDOWNCMD</code>	Befehl zum Herunterfahren des Clients
<code>FINALDELAY 5</code>	Wartezeit vor dem lokalen Herunterfahren

12. POWERDOWNFLAG auf einem Client

Auf einem reinen Netzwerkclient ist normalerweise kein Eintrag erforderlich:

```
\\  
POWERDOWNFLAG /etc/killpower\\
```

Der `POWERDOWNFLAG` wird vor allem auf dem NUT-Master beziehungsweise Primary verwendet.

Der Master kann damit nach dem Herunterfahren der Server das Abschalten des USV-Ausgangs veranlassen.

Der Client besitzt keine direkte USB-Verbindung zur USV und kann die USV daher normalerweise nicht selbst ausschalten.

13. Dateirechte absichern

Die Datei `upsmon.conf` enthält das NUT-Passwort und sollte geschützt werden.

Eigentümer setzen:

```
sudo chown root:nut /etc/nut/upsmon.conf
```

Dateirechte setzen:

```
sudo chmod 640 /etc/nut/upsmon.conf
```

Dateirechte prüfen:

```
ls -l /etc/nut/upsmon.conf
```

Erwartete Ausgabe:

```
-rw-r----- 1 root nut /etc/nut/upsmon.conf
```

14. NUT-Monitor aktivieren und starten

Dienst beim Systemstart aktivieren:

```
sudo systemctl enable nut-monitor
```

Dienst neu starten:

```
sudo systemctl restart nut-monitor
```

Status anzeigen:

```
sudo systemctl status nut-monitor --no-pager
```

Der Dienst sollte folgenden Zustand anzeigen:

```
Active: active (running)
```

15. Verbindungsstatus prüfen

USV-Status vom Client aus abfragen:

```
upsc apc@192.168.1.20 ups.status
```

Bei normaler Netzversorgung sollte erscheinen:

```
\0L
```

Weitere Werte abfragen:

```
upsc apc@192.168.1.20 battery.charge
```

```
upsc apc@192.168.1.20 battery.runtime
```

```
upsc apc@192.168.1.20 ups.load
```

16. Wichtige USV-Statuswerte

Status	Bedeutung
OL	Online: Netzversorgung vorhanden
OB	On Battery: USV läuft auf Batterie
LB	Low Battery: Batterie fast leer
RB	Replace Battery: Batterie sollte ersetzt werden
CHRG	Batterie wird geladen
DISCHRG	Batterie wird entladen
OVER	USV ist überlastet
OFF	USV-Ausgang ist ausgeschaltet
FSD	Forced Shutdown wurde ausgelöst

Mehrere Statuswerte können gleichzeitig angezeigt werden.

Beispiel:

```
\\  
OB DISCHRG\\
```

Das bedeutet:

- * USV läuft auf Batterie
- * Batterie wird entladen

17. Protokoll des Clients kontrollieren

Die letzten Meldungen des NUT-Monitors anzeigen:

```
sudo journalctl -u nut-monitor -n 100 --no-pager\
```

Protokoll laufend beobachten:

```
sudo journalctl -u nut-monitor -f\
```

Bei einer erfolgreichen Verbindung sollten keine wiederkehrenden Fehlermeldungen erscheinen.

Mögliche Fehlermeldungen:

```
\  
Login on UPS [apc] failed\
```

```
\  
Communications with UPS apc@192.168.1.20 lost\
```

```
\  
UPS apc@192.168.1.20 is unavailable\
```

18. Anmeldung auf dem Master kontrollieren

Auf dem Master können die NUT-Protokolle geprüft werden:

```
sudo journalctl -u nut-server -n 100 --no-pager\
```

Bei einer erfolgreichen Anmeldung kann beispielsweise folgende Meldung erscheinen:

```
Client monclient@192.168.1.21 logged into UPS [apc]\
```

Je nach NUT-Version und Protokollkonfiguration kann die Meldung anders aussehen oder fehlen.

19. Sicheren Stromausfalltest durchführen

Auf dem Client die Statuswerte laufend anzeigen:

```
watch -n 2 'upsc apc@192.168.1.20 ups.status; upsc apc@192.168.1.20  
battery.charge; upsc apc@192.168.1.20 battery.runtime'\
```

Danach nur den Netzeingang der USV kurz vom Stromnetz trennen.

Erwarteter Status bei vorhandener Netzspannung:

```
OL
```

Erwarteter Status bei Batteriebetrieb:

```
OB
```

Nach Wiederherstellung der Netzspannung:

```
OL
```

<note important>

Bei diesem kurzen Test darf der Client noch nicht herunterfahren.

Der Server fährt normalerweise erst bei einem kritischen Batteriestatus oder einem Forced Shutdown herunter.

</note>

20. Verhalten bei einem längeren Stromausfall

Bei einem Stromausfall läuft die USV zunächst auf Batterie.

Der typische Ablauf lautet:

- USV wechselt von OL auf OB
- Batterie wird entladen
- Client und Master bleiben zunächst eingeschaltet
- USV meldet bei niedrigem Batteriestand zusätzlich LB
- NUT-Master löst die Abschaltsequenz aus
- NUT-Client fährt kontrolliert herunter
- NUT-Master fährt anschließend herunter
- USV-Ausgang kann danach abgeschaltet werden

21. Vollständigen Shutdown nur im Wartungsfenster testen

Ein vollständiger Test kann auf dem NUT-Master mit folgendem Befehl ausgelöst werden:

```
sudo upsmon -c fsd\
```

<note warning>

Dieser Befehl löst eine echte Abschaltsequenz aus.

Dabei können der Client, der Master und anschließend der USV-Ausgang abgeschaltet werden.

Der Befehl sollte nur in einem geplanten Wartungsfenster verwendet werden.

</note>

Vor einem vollständigen Test sollten folgende Punkte geprüft werden:

- * Laufende Datenbanken können sauber herunterfahren
- * Docker-Container besitzen geeignete Stop-Timeouts
- * Virtuelle Maschinen werden kontrolliert beendet
- * Dateisysteme sind fehlerfrei
- * Netzwerkdienste können beendet werden
- * Beide Server können nach Rückkehr der Netzspannung wieder starten

22. Fehler: Connection refused

Fehlermeldung:

```
Error: Connection failure: Connection refused\
```

Auf dem Master prüfen:

```
sudo systemctl status nut-server --no-pager\
```

Port prüfen:

```
sudo ss -ltnp | grep 3493\
```

Firewall prüfen:

```
sudo ufw status\
```

Konfiguration prüfen:

```
grep -v '^[[:space:]]*#' /etc/nut/upsd.conf\
```

23. Fehler: Unknown UPS

Fehlermeldung:

```
Error: Unknown UPS\
```

Auf dem Master den USV-Namen prüfen:

```
upsc -l localhost\
```

Falls die Ausgabe beispielsweise lautet:

```
smc1500
```

muss auf dem Client statt apc folgender Name verwendet werden:

```
MONITOR smc1500@192.168.1.20 1 monclient CLIENT_PASSWORT_HIER_EINTRAGEN  
secondary\
```

24. Fehler: Login failed

Fehlermeldung im Protokoll:

```
Login on UPS [apc] failed\
```

Folgende Punkte prüfen:

- * Benutzername auf Client und Master identisch
- * Passwort auf Client und Master identisch
- * Rolle korrekt eingetragen
- * Datei upsd.users korrekt gespeichert
- * NUT-Server nach Änderung neu gestartet

Master neu starten:

```
sudo systemctl restart nut-server\
```

Client neu starten:

```
\  
sudo systemctl restart nut-monitor\
```

25. Fehler: Communications lost

Fehlermeldung:

```
Communications with UPS apc@192.168.1.20 lost\
```

Netzwerkverbindung prüfen:

```
ping -c 4 192.168.1.20\
```

NUT-Port prüfen:

```
nc -zv 192.168.1.20 3493\
```

Direkte Statusabfrage versuchen:

```
upsc apc@192.168.1.20\
```

Protokoll auf dem Master prüfen:

```
sudo journalctl -u nut-server -n 100 --no-pager\
```

26. Fehler: NUT-Monitor startet nicht

Dienststatus anzeigen:

```
sudo systemctl status nut-monitor --no-pager\
```

Protokoll anzeigen:

```
sudo journalctl -u nut-monitor -n 100 --no-pager\
```

Konfiguration auf aktive Einträge prüfen:

```
grep -v '^[[:space:]]*#' /etc/nut/upsmon.conf\
```

Betriebsart prüfen:

```
cat /etc/nut/nut.conf\
```

Erwartete Ausgabe:

```
MODE=netclient\
```

27. Komplette Clientkonfiguration

/etc/nut/nut.conf

```
MODE=netclient\
```

/etc/nut/upsmon.conf für Ubuntu 22.04

```
MONITOR apc@192.168.1.20 1 monclient CLIENT_PASSWORT_HIER_EINTRAGEN slave
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
FINALDELAY 5
```

/etc/nut/upsmon.conf für Ubuntu 24.04

```
MONITOR apc@192.168.1.20 1 monclient CLIENT_PASSWORT_HIER_EINTRAGEN
secondary
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
FINALDELAY 5
```

28. Abschlusskontrolle

Auf dem Client folgende Befehle ausführen:

```
cat /etc/nut/nut.conf
```

```
sudo systemctl status nut-monitor --no-pager\
```

```
upsc apc@192.168.1.20 ups.status\
```

```
upsc apc@192.168.1.20 battery.charge\
```

```
sudojournalctl -unut-monitor -n50-no-pager\
```

Bei normalem Betrieb sollte der Status lauten:

```
0
```

Ergebnis

Der zweite Ubuntu-Server ist nun als NUT-Client eingerichtet.

Er kann:

- * die APC-USV über den ersten Server überwachen
- * Stromausfälle erkennen
- * den Batteriestatus auslesen
- * einen kritischen Batteriezustand erkennen
- * bei einer NUT-Abschaltmeldung kontrolliert herunterfahren

From:

<http://wiki.waldhofer.at/> - **Wiki von Franz**

Permanent link:

<http://wiki.waldhofer.at/doku.php?id=ubuntu:usv2&rev=1784470465>

Last update: **2026/07/19 16:14**

