

# APCUPSD Multimon zentraler USV-Server

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 🙏 . Gerne. Wir bedanken uns bei dir für deine Spende! ☐

[Spenden](#)

Zum frei verfügbaren [Apt-Repository](#)



GITLAB:

[Wechsle zum Englischsprachigem Artikel.](#)

## APC UPS Daemon

Der [APCupsd](#) kann wesentlich mehr als du immer für möglich gehalten hast. Nicht nur seriell oder USB, oder über SNMP Ethernetkarten. Nein, mit der Software ist es möglich rege USV-Cluster zu bauen, wobei Clients (slaves) nicht mal eine APCusv verwenden müssen. Jegliche USV ist erwünscht. Folgende Möglichkeiten beschreibe ich dir hier in diesem HowTo:

- APC USV per USB angeschlossen
- APC USV wird zwar als Stromquelle verwendet, ist aber am Client nicht per USB oder sonstigem Kabel angeschlossen
- Fremdmarken USV

### Was ist es nicht?

- Es ist kein zentrales Konfigurations, oder Steuerungsinterface.

### Was kann ich damit tun?

- Zentrale Übersicht aller angeschlossenen Geräte/USV's und deren Strom/Akku Status
- Shutdown auch von Geräten die keine Datenverbindung zu ihrer USV haben, oder eine Fremdmarken USV verwenden
- Eine einzige große Zentrale USV ansteuern

## Voraussetzung und Info zu den Systemen

Im Test hier wurden 4 APC USV's verwendet, aber 5 Server angeschlossen. Wobei der eine extra Server auf einer APC's dazu konfiguriert wurde, wo die USBschnittstelle der UPS bereits von dem anderen Server verwendet wird und daher keine Datenverbindung hat. Die verwendeten System sind hier [Proxmox 6.3](#), [Ubuntu 20.04](#) und [Raspbian 10 Buster](#). Auf allen Geräten muss der APCupsd installiert und konfiguriert sein.

## Installation und Grundkonfiguration

Das Paket ist bereits in allen Distribution enthalten und kann ganz einfach über das Paketmanagement installiert werden.

```
apt install apcupsd
```

Danach muss als erstes die Datei `/etc/default/apcupsd` bearbeitet werden und `ISCONFIGURED=no` auf `ISCONFIGURED=yes` gesetzt werden. Als nächstes begibst du dich in die Hauptkonfiguration von APCupsd:

```
nano /etc/apcupsd/apcupsd.conf
```

Hier sind dies die wichtigsten Punkte:

```
UPSNAME meinUSVname_den_ich_vergeben_möchte
UPSCABLE usb
UPSTYPE usb
ONBATTERYDELAY 6      #Zeitspanne in Sekunden, ab der der Systemdienst einen
Stromausfall annimmt. Nützlich, um sehr kurze Ausfälle ignorieren zu können.
BATTERYLEVEL 10      #Durch die USV gemeldete Restkapazität in Prozent, ab
der der Rechner heruntergefahren wird
MINUTES 5            #Zeitpuffer in Minuten, der für das Herunterfahren
einkalkuliert wird.
```

Die Werte wirst du vermutlich deinen Gegebenheiten anpassen möchten. Nach einem Neustart des APCupsd `systemctl restart apcupsd.service` kannst du sofort auf der CMD mit dem Befehle `apcaccess` die aktuellen Werte der USV ausgeben. Somit wäre die Grundkonfiguration einer APC USV abgeschlossen.

## Erweiterte Konfiguration APC-Cluster mit zentralem USV-Server

Nun möchten wir einen von unseren Servern zum USV-Master-Server machen. Welcher dies sein darf muss man entscheiden. Am besten ein Server/PC/Gerät der vom Akku am längsten hält, da Geräte die keine Datenverbindung zu einer APC USV haben, ohne diesen Server keinen Shutdown initiieren können. Hierbei musst du dir auch unbedingt Gedanken machen bei der Clientkonfiguration welche Geräte bei welchem Level herunter gefahren werden sollen. Der Level der Fremdbusv kann ja nicht gemessen werden, diese Werte legst du selbst auf diesem Client in der `apcupsd.conf` fest. Und ohne Verbindung zum Masterserver werden diese auch logischer Weise ignoriert.

### Konfiguration APC-Masterserver

```
nano /etc/apcupsd/apcupsd.conf
```

Hier ändern sich folgende Werte: (manche sind bereits Default gesetzt)

```
NETSERVER on
NISIP <erlaubtes Subnet> #Defaultwert ist hier 0.0.0.0
```

Installieren der Serverkomponente:

```
apt install apache2-utils apcupsd-cgi apache2
a2enmod cgi
systemctl restart apache2.service
```

Nun noch die default /var/www/html/index.html anpassen.

```
<html>
  <head>
    <meta http-equiv="refresh" content="0;
URL=http://FQDN-deines-Servers/cgi-bin/apcupsd/multimon.cgi">
  </head>
</html>
```

Ab jetzt ist dein Masterserver bereits im Netz erreichbar.

## Konfiguration APC-Client

Ein APC-Client hat selbst eine APC USV mit Datenverbindung und ist in der Lage den Shutdown selbst zu steuern. In diesen Fall lassen wir den Client in die zentrale Übersicht hinein wachsen. Wir editieren am unserem Client die apcupsd.conf. Folgende Teile der Config sind zu beachten:

```
UPSNAME meinUSVname_den_ich_vergeben_möchte
UPSCABLE usb
UPSTYPE usb
ONBATTERYDELAY 6    #Zeitspanne in Sekunden, ab der der Systemdienst einen
Stromausfall annimmt. Nützlich, um sehr kurze Ausfälle ignorieren zu können.
BATTERYLEVEL 10    #Durch die USV gemeldete Restkapazität in Prozent, ab
der der Rechner heruntergefahren wird
MINUTES 5          #Zeitpuffer in Minuten, der für das Herunterfahren
einkalkuliert wird.
NETSERVER on
NISIP #Hier wird die IP dieses Client eingetragen z.B. 192.168.4.88
NISPORT 3551
```

Ein `systemctl restart apcupsd.service` aktiviert die aktuelle Config. Nun ist der diese APC über das Netz ansprechbar und kann im übernächsten Schritt am Masterserver eingetragen werden.

## Konfiguration APC-Client (Slave)

Ein APC-Client (Slave) ist natürlich an einer USV angeschlossen. Dies muss keine APC sein. Der wichtigste Unterschied hier ist dass, dieser Client zu der USV wo er angeschlossen ist, keine

Datenverbindung besitzt. Sprich APCupsd ist der Stand des Akkus und des Stromnetzes nicht bekannt. Wird somit auch bei einem Stromausfall nicht getriggert. Um so einen Client trotzdem gesteuert herunter fahren zu können, lassen wir diesen auf dem Masterserver schauen. Wir editieren am unserem Client die `apcupsd.conf`. Folgende Teile der Config sind zu beachten:

```
UPSNAME meinUSVname_den_ich_vergeben_möchte
UPSCABLE usb
UPSTYPE net
DEVICE FQDN-MasterServer:3551
ONBATTERYDELAY 6    #Zeitspanne in Sekunden, ab der der Systemdienst einen
Stromausfall annimmt. Nützlich, um sehr kurze Ausfälle ignorieren zu können.
BATTERYLEVEL 10    #Durch die USV gemeldete Restkapazität in Prozent, ab
der der Rechner heruntergefahren wird
MINUTES 5          #Zeitpuffer in Minuten, der für das Herunterfahren
einkalkuliert wird.
NETSERVER on
NISIP #Hier wird die IP dieses Client eingetragen z.B. 192.168.4.88
NISPORT 3551
```

Ein `systemctl restart apcupsd.service` aktiviert die aktuelle Config. Nun ist der diese APC über das Netz ansprechbar und kann im nächsten Schritt am Masterserver eingetragen werden.



Fällt der Strom „NUR“ an dieser einen USV aus (APC-Client/Slave), bekommt das weder der Master noch der Client mit. Der Shutdown funktioniert hier nur wenn der Master auch auf Batteriestrom umschaltet. Schaltet der Master auf Batteriestrom um, und hätte der Client aber normal Strom, wird der Client trotzdem nach den Werten in der Config gesetzt wurden, heruntergefahren.



Der Grundgedanke ist hier definitiv eine große Zentralusv zu verwenden, und nicht jede Menge kleine USV's verstreut, oder gar APC Fremde zu verwenden. Funktionieren tut dies natürlich auch sofern man es gut ausarbeitet.

## APC-Client's zum APC Masterserver hinzufügen

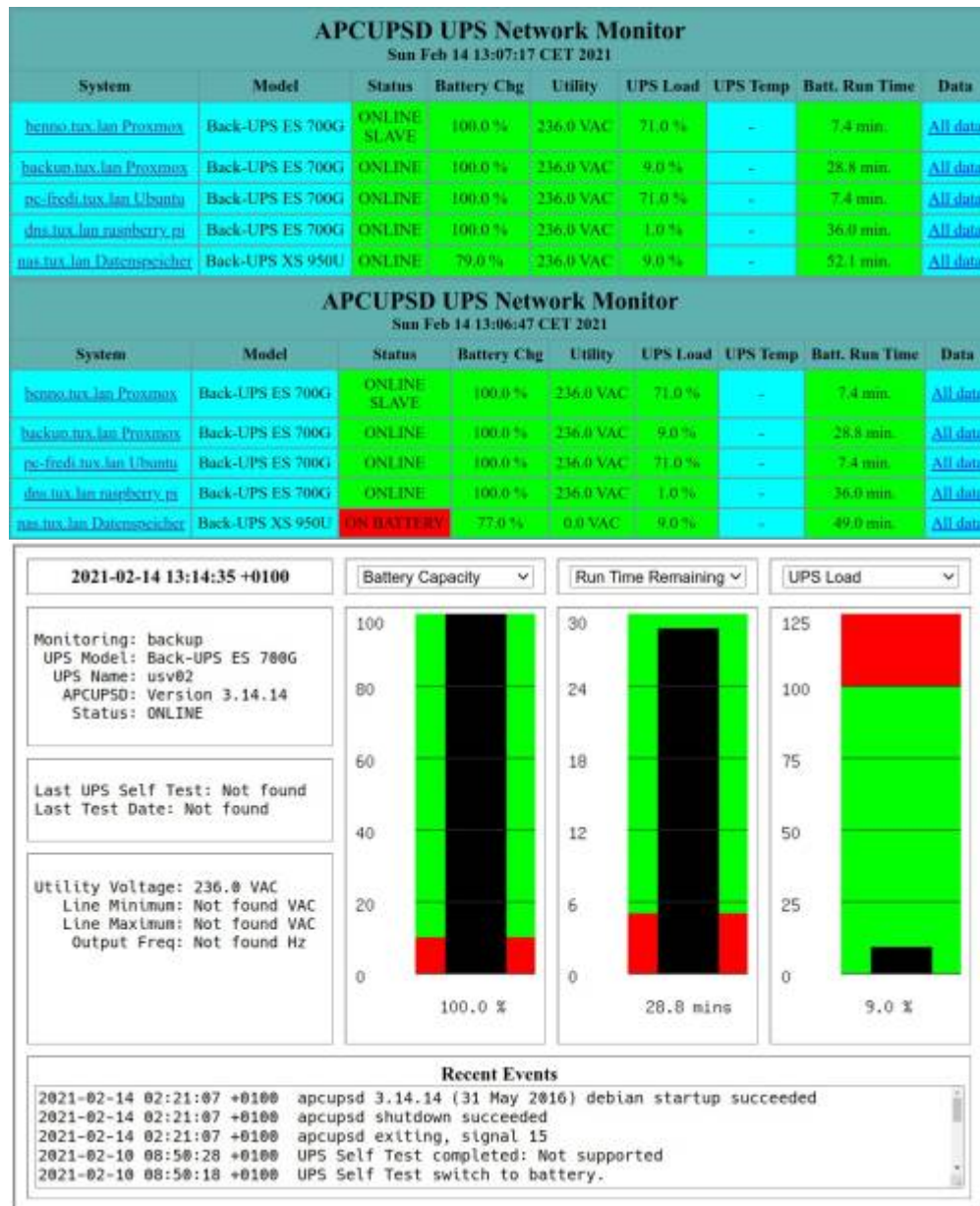
Dies ist einfach und schnell erledigt. `nano /etc/apcupsd/hosts.conf`

```
MONITOR benno.tux.lan "benno.tux.lan Proxmox"
MONITOR backup.tux.lan "backup.tux.lan Proxmox"
```

MONITOR pc-fredi.tux.lan "pc-fredi.tux.lan Ubuntu"  
 MONITOR dns.tux.lan "dns.tux.lan raspberry pi"  
 MONITOR nas.tux.lan "nas.tux.lan Datenspeicher"

In dem zentalem Webinterface sind nun sofort alle Geräte ersichtlich:

<http://FQDN-deines-Servers/cgi-bin/apcupsd/multimon.cgi> Zur originalen Ansicht ein Bild anklicken.



# Links

- <https://wiki.ubuntuusers.de/USV/Apcupsd/>
- <http://www.apcupsd.org>

From:  
<https://www.deepdoc.at/dokuwiki/> - **DEEPDOC.AT** - enjoy your brain

Permanent link:  
[https://www.deepdoc.at/dokuwiki/doku.php?id=server\\_und\\_serverdienste:apcupsd\\_multimon\\_zentraler\\_usv-server](https://www.deepdoc.at/dokuwiki/doku.php?id=server_und_serverdienste:apcupsd_multimon_zentraler_usv-server)

Last update: **2025/11/29 22:06**

