Network UPS Tools (NUT)

This guide explains how to install NUT on a client / server architecture with several clients accessing the UPS.

NUT Server

Installation

```
# apt-get install nut
```

Installing the nut package, should add the nut user and group. If it doesn't, you have to add those manually. Next, create or edit the necessary configuration files, which are located in /etc/nut.

ups.conf

```
[ups]
  driver = usbhid-ups
  port = auto
  pollonly
  desc = "ups"
```

The label in the square brackets can be anything. It identifies the UPS to upsd and upsmon. Find your UPS model on the NUT hardware compatibility list and substitute the corresponding driver for usbhid-ups above. If your UPS isn't listed, find similar ones. They probably use the same driver. If you have a USB-based UPS, you can leave the port as auto; it's ignored.

Note: if you are connecting a Synology NAS to your NUT setup, then the label in square brackets *must be named* **ups**.

Make sure that nut properly detects the UPS:

```
# upsdrvctl start
```

You should see something like the following:

```
Network UPS Tools - UPS driver controller 2.2.2
Network UPS Tools: 0.29 USB communication driver - core 0.33 (2.2.2)
Using subdriver: APC HID 0.92
```

Now, we need to configure upsd and upsmon. upsd communicates with the UPS driver that we just started. upsmon communicates with upsd and actually shuts down the machine in the event of a power failure. Multiple instances of upsmon can be started on different machines. They can all share the same physical UPS. Create the file /etc/nut/upsd.conf as follows:

upsd.conf

```
LISTEN 127.0.0.1 [<port>] # default, default port is 3493
LISTEN 192.168.x.x [<port>] # server IP address to access from other
machines
```

Don't forget to open the port in a zone rule, if you access upsd from another machine through a firewall.

upsd.users

```
[local_mon]
  password = <pwd>
  allowfrom = localhost # optional
  upsmon master
```

You should add multiple users if you are monitoring from multiple machines. See the man page for upsd.users for more information. Now we need to configure upsmon.

Note: if you are connecting a Synology NAS to your NUT setup, then you *must add* the Synology NAS user like so:

```
[monuser]
  password = secret
  upsmon slave
```

upsmon.conf

```
MONITOR apc@localhost 1 local_mon <pwd> master
POWERDOWNFLAG /etc/killpower
SHUTDOWNCMD "/sbin/shutdown -h +0"
```

Replace apc with the name you have your UPS in /etc/nut/ups.conf and <pwd> with the password you set in /etc/nut/upsd.users. Since these files contain sensitive data, we should fix the permissions:

```
# chown root:nut /etc/nut/*
# chmod 640 /etc/nut/*
```

nut.conf

```
MODE=netserver
```

Starting

Finally, edit /etc/default/nut so that upsd and upsmon will start at system boot:

```
# vim /etc/default/nut
START_UPSD=yes
START_UPSMON=yes
```

Start NUT:

```
# /etc/init.d/nut-server start
```

Check the end of /var/log/daemon.log to make sure everything started up. Also, you can run the following command which should print out some statistics:

```
$ upsc apc
```

Replace apc with whatever you named your UPS.

Testing

If you rely on the UPS to shutdown your systems in case of mains failure and to restart them when the power returns, you must test this. You can do so by running

```
upsmon -c fsd
```

With the mains present, this should bring your systems down and then cycle the power to restart them again. If you do the same without mains present, it should do the same, but in this case, the outputs shall remain off until mains power is applied again.

NUT Client

Installation

```
# apt-get install nut
```

Installing the nut package, should add the nut user and group. If it doesn’t, you have to add those manually. Next, create or edit the necessary configuration files, which are located in /etc/nut.

upsmon.conf

```
MONITOR apc@<server> 1 remote_mon <pwd> slave
POWERDOWNFLAG /etc/killpower
SHUTDOWNCMD "/sbin/shutdown -h +0"
```

Replace apc with the name you have your UPS in /etc/nut/ups.conf, <server> with the server name or IP address upsd is running on, and <pwd> with the password you set in /etc/nut/upsd.users, on the server.

nut.conf

```
MODE=netclient
```

Starting

Finally, edit /etc/default/nut so that upsmon will start at system boot:

```
# vim /etc/default/nut
START_UPSMON=yes
```

Start NUT:

```
# /etc/init.d/nut-client start
```

Check the end of /var/log/daemon.log to make sure everything started up. Also, you can run the following command which should print out some statistics:

```
$ upsc apc@<server>
```

Replace apc with whatever you named your UPS, and <server> with the server name or IP address upsd is running on.

NUT Client CGI access on a web server

Installation

```
# apt-get install nut nut-cgi
```

Install the Client as described above in NUT Client.

Configuration for web server

- Add the MONITOR command for hosts to be accessible in /etc/nut/hosts.conf
- Remove the ### coments in the last line of /etc/nut/upsset.conf after securing access to the webserver page
- Add alias to the apache configuration in /etc/apache2/conf.d:

  ```
  Alias /nut /usr/share/nut/www
  ```

Synology DSM Client

Connect UPS

Make sure you have named your ups as described in section ups.conf, and added the Synology NAS user to upsd.users.

Login to DSM and navigate to **Control Panel -> Hardware & Power -> UPS**. Enable UPS support and enter the *Network UPS server IP*.

You can then click [Device Information] to verify the UPS is connected. Should Synology ever change the defaults, log into your Diskstation via SSH and check the configuration files at /usr/syno/etc/ups.

Full shutdown and reboot

The Synology DSM and NAS hardware cannot start automatically when power is restored. That's why the default beviour is for the NAS to go into safe mode, which means it switches off disk access and other services to protect the NAS from taking damage if the power goes out. When the battery is fully drained and the power goes out, the NAS will completely switch off, from which it can recover when the power comes back on.

One other way would be to completely shut down the NAS, and then reboot it once the power comes back. Note that once the power has been disconnected form the NAS, then wake-on-lan would not work anymore. For a complete shut down and recovery, you would need to power the shut down NAS independently, so the NICs never loose complete power. This could be done with a secondary battery which powers only the NAS and which maintains enough battery power to supply a shut down NAS for hours or days… not an easy setup.

If you want to completely switch off the NAS, shut it down from the machine where the UPS is connected to (e.g. the “server”) via SSH, and restart it with a wake-on-lan command.

1. Install etherwake on the server
2. Put the following command with the NAS' MAC address into your startup script of the server

   ```
   wakeonlan <mac>
   ```

3. Replace the SHUTDOWNCMD in upsmon.conf of the server with the following script

   ```
   ssh root@<nas> "shutdown -h +0"
   /sbin/shutdow -h +0
   ```

4. Enable wake-on-lan in the NAS, in **Control Panel -> Hardware & Power -> General**.

UPS Hardware

You also must configure the UPS hardware correctly for all of this to work. For an APC Smart-UPS 2200VA, for example, you would need to set the following in advanced settings to make sure it cooperates with your NUT setup:

1. Check the **Maximum Required Delay** after having connected all your clients and after configuration of NUT

2. Set **Low Battery Duration** to a value equal or higher than that
3. Set **Shutdown Delay** which is the time the UPS will keep running after the clients have shut down. This will power remaining network devices.

In an unstable power environment, I would recommend the following additional measures:

1. Set **Low Battery Duration** with an additional buffer. The buffer should allow to reboot, but power down again gracefully after reboot if the power goes out again. Say, power comes back on for just 5 minutes, then you need approximately 10 minutes of remaining battery power for full load to shut down gracefully a second time.
2. Set **Reboot Duration** to a higher value, for example 5 minutes, which means the UPS waits 5 minutes after power is restored before it supplies power to it's clients. This will avoid rebooting the clients if the restored power goes out within 5 minutes.

Links

NUT Installation

- Powerwalker UPS unter Debian 9 Stretch
- Debian / Ubuntu Linux: APC Controlling and Monitoring UPSes with apcupsd
- Monitoring a UPS with nut on Debian or Ubuntu Linux
- Network UPS Tools User Manual
- Connecting a Synology DiskStation to a NUT server
- upsc - example lightweight UPS client

Other resources

- HowTo: Configure Low Battery Duration and PCNS to Shut Down with an APC SmartUPS
- Smart-UPS Operation Manual
- Monitoring UPS Power Status Using Network UPS Tools (NUT) 2.2.0 on Multiple OpenSuSE 10.3 Servers
- Debian 7 - Install & Monitor EATON UPS
- Web monitoring of an UPS with NUT
- APC order in Thailand

From:
https://wiki.condrau.com/ - **Bernard's Wiki**

Permanent link:
https://wiki.condrau.com/deb720:ups

Last update: **2019/04/11 15:48**