Setting up NUT server

This documents how to set up a *nut* server. It is for monitoring one or more UPS' and providing the monitoring to one or more devices.

This document was written using the following configuration.

- Cyberpower PR1000LCDRT2U UPS with an RMCARD205 installed, giving LAN access to the UPS control, snmp monitoring and other things
- One Devuan (Debian) workstation will be set up as the master. It will communicate with the UPS, then other machines will communicate with it
- One opnSense firewall/router with the UPS module installed
- One FreeBSD NAS

Note: a Raspberry Pi works very well as a stand alone NUT server. See https://www.jeffgeerling.com/blog/2025/nut-on-my-pi-so-my-servers-dont-die for full instructions.

Set up UPS

Log into the UPS via web browser, ssh or telnet and get/set the following:

- snmp community name
- snmp version (I'm using v1 in a secured environment)
    - If using v2, get username and password
- record the IP address/DNS Name
- choose the name you want to give to this UPS

Set up master

Installation

first, install the necessary packages. I'm not sure if all of these are needed, but I think they are I chose not to install the webui as I monitor the UPS with Zabbix, and manage through the RMCARD

```
apt -y install snmp snmp-mibs-downloader nut nut-client nut-server nut-snmp
```

Configure

On Devuan/Debian systems, the NUT configuration files are in /etc/nut. There are defaults in there which have excellent internal documentation, so you don't want to destroy them. I tend to rename all of them to have a .original suffix, then build the configs from scratch.

Set up the master to be a network server in nut.conf

nut.conf

```
MODE=netserver
```

This is the nasty part. We need to have ups.conf set up with the correct stanza to read the snmp UPS. Normally we would just point nut-scanner at it, but nut-scanner has some issues, at least on 2.7.4, under Debian where it can not find the snmp libraries. So, autodetect does not work.

```
nut-scanner --snmp_scan --mask_cidr 192.168.100.0/24 --disp_nut_conf
```

returns “SNMP library not found. SNMP search disabled.” even though the snmp libraries are there. It is supposed to be fixed with later versions. If it is working on your system, it will output a stanza that can be directly pasted into ups.conf. Otherwise, we have to do it the hard way.

If, you are running a distribution that does have a working nut-scanner, you can use the following command (modified for your system). This was tested on FreeBSD running on a Raspberry Pi. It will look for SNMP and USB attached UPS'

```
nut-scanner -U -S -m 192.168.100.0/24 -c public
```

On systems without a working nut-scanner, we'll do it the way that is more susceptible to error. Modify the following to match your setup, then put it in ups.conf

ups.conf

```
maxretry = 3
# the part in the [] ([ups1]) is the name of the UPS, which you will
use several places later
[ups1]
   driver = snmp-ups
   # the IP and port of the snmp enabled UPS
   port = 192.168.1.17:161
   # the community name it will respond to, preferably read/only access
   community = public
   # the version of snmp on the UPS
   snmp_version = v1
   # an arbitray description, if you want
   desc = "Primary UPS"
```

Now, set up where the server (the NUT server on your master) will listen for other machines to make the connection. 0.0.0.0 means “anyplace”, and 3493 is the default port

upsd.conf

```
LISTEN 0.0.0.0 3493
```

Set up a user. Here, we're setting up a user names upsmon, with the password 'mypassword' and

setting that user to be a master, ie the other machines will talk here, using this user to check status of the UPS. There will be another line in upsmon.conf that will reference this user.

upsd.users

```
[upsmon]
   password = mypassword
   upsmon master
```

Finally, set up the monitoring itself with upsmon.conf The first few lines are some defaults with the standard nut install; the main thing we are chaning is the final line.

The final line says Monitor the system (UPS) named ups1 on the local machine (ups1@localhost), with a powervalue of 1 (almost always a 1, not more). Note, if you are stictly monitoring a UPS that you are not attached to, put a 0 here.

upsmon is the username to connect to that server (ups1@localhost), and mypassword is the password.

'master' designates this as a master server; it will NOT shut down immediately when power goes critical. 'slave' in this part will shut down when power goes critical on ups1@localhost

upsmon.conf

```
MINSUPPLIES 1
SHUTDOWNCMD "/sbin/shutdown -h +0"
POLLFREQ 5
POLLFREQALERT 5
HOSTSYNC 15
DEADTIME 15
POWERDOWNFLAG /etc/killpower
RBWARNTIME 43200
NOCOMMWARNTIME 300
FINALDELAY 5
monitor ups1@localhost 1 upsmon mypassword master
```

I don't really understand upssched.conf yet, but it is one of the powerhouses of nut. Working on it.

upssched.conf

```
CMDSCRIPT /bin/upssched-cmd
```

Test

First, restart the service. On Devuan, it is called nut-server

```
service nut-server restart
```

Test to make sure you have a connection and the server is correctly reading the UPS. You do not have to put in the @localhost; that is the default.

```
# get list of all UPS' defined on localhost
upsc -L localhost
# get list of all values on one particular UPS (ups1)
upsc ups1@localhost
```

NUT ups commands

- upsc - lightweight UPS client. Really, set up to be an example of what you can create, but very useful.

  ```
  upsc -c upsname
  ```

  will show the clients which are monitoring a particular ups.

- upscmd - send commands to UPS if it will accept. Try

  ```
  upscmd -l <upsname>
  ```

- upsd - This is the primary daemon that should always run to provide information about one or more UPS'. Generally auto started by the init system
- upsdrvctl - standard interface to the UPS drivers. Not generally needed unless you are writing/debugging a particular driver.
- upslog - A tool designed to read one or more UPS' and log the readings. Very useful for upssched if you want a nice log of how your UPS is working.
- upsmon - A daemon which monitors all UPS' defined in upsmon.conf
- upsrw - allows you to change some variables *on the ups itself* if the ups allows it.
- upssched - scheduler to run commands listed in upssched.conf. This is called by upsmon and should not be used directly.
- upssched-cmd - A sample upssched file. A simple shell script that performs commands in response to upssched.

Links

- https://www.jeffgeerling.com/blog/2025/nut-on-my-pi-so-my-servers-dont-die
- https://technotim.live/posts/NUT-server-guide/
- https://networkupstools.org/docs/man/snmp-ups.html

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/software/nut/master

Last update: **2025/07/03 07:45**