



ownCloud auf dem Wi-Fi Router

**oder: wie weit komme ich mit dem Router als
Server zuhause**

Übersicht



- Das Ziel: Router als Home-Server
 - OpenWRT – was ist das ?
- ownCloud nach Rezept
 - Benchmarks
- ownCloud mit tuning
 - Mehr Benchmarks
- Fazit

Router als Server



- Ein WiFi-Router ist oft schon da – oder auch ein zweiter um WiFi Abdeckung zu erreichen.
 - Oft haben diese schon Funktionen wie caching, NAS etc.
- Es liegt nahe alle weiteren Funktionen auf diese Geräte zu konzentrieren.
 - ownCloud ist hier das Ziel

Die Fragen:

- Geht es ? (LMGTFY: ja, aber ...)
- Ist es wirklich nutzbar ?

Was ist OpenWRT ?



Eine Linux-Distribution für Home-Router

Also einfach das passende Paket installieren und fertig ?

Was ist OpenWRT ?



Also einfach das passende Paket installieren und fertig ?

So einfach ist es leider nicht:

- OpenWRT (und DD-WRT) sind schlank getrimmt, ab 8 MB Flash und 16 MB RAM
- `procd` statt `systemd` oder `init`
- `ubus` statt `dbus`
- Configuration in nvram via `luci`
- Beschreibbares Filesystem optional
- Grössere Datenmengen auf USB drives

Was ist OpenWRT ?



Es lohnt sich OpenWRT und seine Ableger zu kennen:

- Immer öfter auch die Basis des OS vom Hersteller, z.B. Technicolor
- Und vom ISP gelieferten Geräten
- Kandidat für Server-Container im NFV Kontext [NFV = Network Function Virtualization]

Links:

- <https://wiki.openwrt.org/toh/start> Hardwareliste
- <http://openwrtsummit.org> Kongress Berlin Okt. 2016

Der Server



Netgear R7000 AC1900 Nighthawk

- 2x ARMv7 @ 1GHz
- 128 MB Flash
- 256 MB RAM
- 5x 1 GE Ethernet
- 1x USB3 und 1 USB2 ports
- Dualband WiFi MIMO 3x3

Disk:

- 500GB SSD via USB3



ownCloud Installation



Outline:

- Externen Speicher aktivieren
- Package Management aufsetzen (opkg)
- Alle notwendigen Abhängigkeiten installieren
- ownCloud installieren
- ownCloud, php, lighthttp etc. konfigurieren
- Alles für auto-startup aktivieren

Links:

- <http://www.dd-wrt.com/phpBB2/viewtopic.php?p=1016782&sid=5a2cdb33ac3e0623001e60a00748ee85>
- <https://www.hqt.ro/owncloud-through-lighthttpd-entware-ng/>

ownCloud Installation



Externen Speicher aktivieren

--- /dev/sda1

Block device, size 486.3 MiB (509935104 bytes)

Linux swap, version 2, subversion 1, 4 KiB pages, little-endian

Swap size 486.3 MiB (509923328 bytes, 124493 pages of 4 KiB)

/dev/sda1 mounted to **swap**

--- /dev/sda2

Block device, size 465.3 GiB (499595281920 bytes)

Ext4 file system

UUID 6381851E-9989-40F2-A9C9-7087E5176712 (DCE, v4)

Volume size 465.3 GiB (499595280384 bytes, 121971504 blocks of 4 KiB)

/dev/sda2 mounted to **/opt**

ownCloud Installation



Package Management aufsetzen (opkg)

```
root@DD-WRT:~# df -h
Filesystem      Size      Used Available Use% Mounted on
/dev/root       19.3M     19.3M      0 100% /
/dev/sdb2       457.9G    69.9M    434.5G   0% /opt
/dev/mtdblock/5 93.8M     2.4M     91.4M   3% /jffs

root@DD-WRT:/jffs# wget -c -O entware-ngu-setup.sh http://goo.gl/hshQkA
entware-ngu-setup.sh 100%
root@DD-WRT:/jffs# chmod +x ./entware-ngu-setup.sh
root@DD-WRT:/jffs# mkdir /jffs/scripts
root@DD-WRT:/jffs# ./entware-ngu-setup.sh
[...]
Info: Looking for available partitions...
[1] --> /opt
=> Please enter partition number or 0 to exit
[0-1]: 1
Info: /opt selected.
[...]
```

ownCloud Installation



Alle notwendigen Abhängigkeiten installieren

```
root@DD-WRT:~# opkg install bzip2 php5-cgi lighttpd-mod-fastcgi lighttpd-mod-access \  
  php5-mod-ctype php5-mod-curl php5-mod-dom php5-mod-fileinfo php5-mod-gd \  
  php5-mod-hash php5-mod-iconv php5-mod-json \ php5-mod-mbstring php5-mod-pdo \  
  php5-mod-pdo-sqlite php5-mod-session php5-mod-simplexml php5-mod-sqlite3 \  
  php5-mod-xml php5-mod-zip php5-mod-xmlwriter php5-mod-ldap php5-mod-openssl \  
  openssl-util php5-mod-xmlreader
```

Und ownCloud installieren

```
root@DD-WRT:/opt/share/www# wget http://download.owncloud.org/community/  
owncloud-9.1.2.tar.bz2  
Connecting to download.owncloud.org (138.201.139.178:80)  
owncloud-9.1.2.tar.b 100% |*****| 28360k  
  
root@DD-WRT:/opt/share/www# bzip2 -cd owncloud-9.1.2.tar.bz2 | tar -xv  
[...]  
root@DD-WRT:/opt/share/www# mv owncloud-9.1.2.tar.bz2 /opt
```

ownCloud Installation



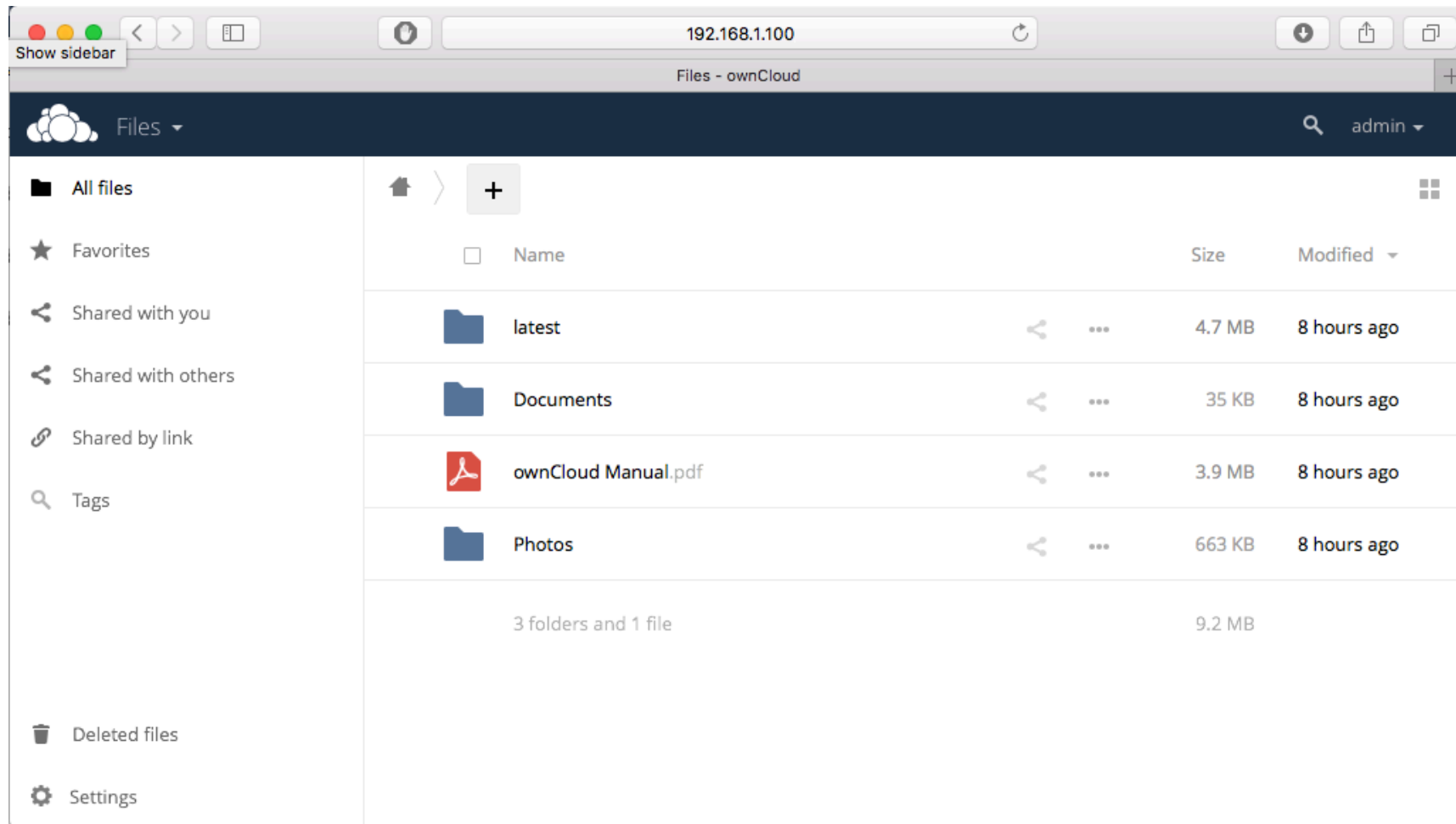
- ownCloud, php, lighthttp etc. konfigurieren
- Alles für auto-startup aktivieren

Change / Create:

- /opt/etc/lighthttpd/lighthttpd.conf
- /opt/etc/php.ini
- /jffs/startup/myStartup.sh
- nvram setting for startup

Endlich ...

ownCloud Installation



Benchmarks



Aber funktioniert das auch vernünftig ?

- Synchronisieren eines Archiv-folders, 4.2GB, ca. 19k files & folders
- ownCloud client auf Mac, direkt verbunden via Gigabit Ethernet

```
Reference Fileset:  
Cbook3 mnbokaem$ du -sh .  
4.2G      .  
cbook3 mnbokaem$ ls -lR . | wc -l  
19085
```

Resultat: Gestoppt nach 70 Minuten mit ~600MB / 6k files erledigt

```
Mem: 209820K used, 44696K free, 0K shrd, 7716K buff, 103388K cached  
CPU: 78.0% usr 16.1% sys 0.0% nic 1.1% idle 0.0% io 0.0% irq 4.6% sirq  
Load average: 2.69 1.11 0.51 5/91 3284  
  PID  PPID  USER   STAT   VSZ  %VSZ  CPU  %CPU  COMMAND  
 2594  2592  root    R      34028 13.2   1 30.0  /opt/bin/php-cgi  
 2597  2595  root    R      39796 15.5   1 28.9  /opt/bin/php-cgi  
 2591  2589  root    R      39280 15.3   0 28.5  /opt/bin/php-cgi  
 2585     1  root    S       5684  2.2   0  4.0  lighttpd
```

=> Das ist noch nicht so toll

Benchmarks - Referenz



Was macht ein 'richtiger Server' (Container on XeonD)

- Installiert nach Rezept (kein tuning)
- Der gleich Archiv-Folder synchronisiert in 20 Minuten

Das Netzwerk kommt nicht auf Gigabit – ist aber nicht das Problem

```
root@DD-WRT:~# iperf -c 192.168.1.33
-----
Client connecting to 192.168.1.33, TCP port 5001
TCP window size: 45.0 KByte (default)
-----
[ ID] Interval      Transfer    Bandwidth
[  3] 0.0-10.0 sec  370 MBytes  310 Mbits/sec
```

```
cbook3:~ mnbokaem$ iperf -c 192.168.1.100 -w 256k
-----
Client connecting to 192.168.1.100, TCP port 5001
TCP window size: 257 KByte (WARNING: requested 256 KByte)
-----
[ ID] Interval      Transfer    Bandwidth
[  4] 0.0-10.0 sec  533 MBytes  447 Mbits/sec
```

Tuning I

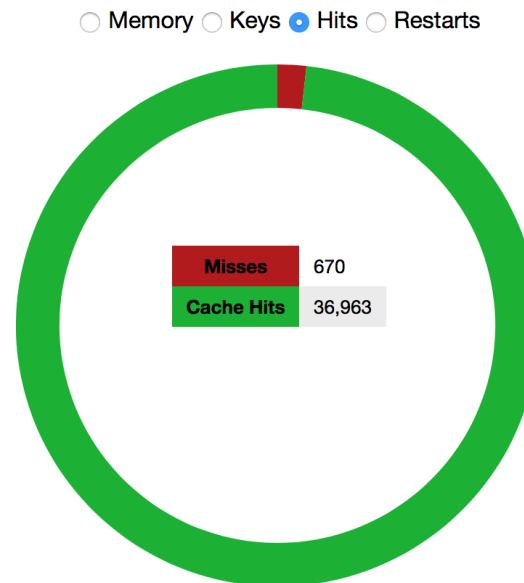


ownCloud ist ein PHP Monster

- Code-Caching ist heute meist default, nicht aktuell in opkg

```
root@DD-WRT:/opt/etc# opkg install php5-mod-opcache
Installing php5-mod-opcache (5.6.26-1) to root...
Downloading http://pkg.entware.net/binaries/armv7/php5-mod-
opcache_5.6.26-1_armv7soft.ipk.
Configuring php5-mod-opcache.
```

- Nach 5 Minuten:



Benchmark II



- Folder synchronisiert in 2 Stunden, ca. Faktor 2.5 schneller

```
Mem: 206404K used, 48112K free, 0K shrd, 33944K buff, 110436K cached
CPU: 50.2% usr  6.3% sys  0.0% nic 41.8% idle  1.1% io  0.0% irq  0.3% sirq
Load average: 2.80 2.79 1.99 3/92 7024
  PID  PPID  USER      STAT   VSZ  %VSZ  CPU  %CPU  COMMAND
 6814  6453  root       R      95560 37.3   0 18.7  /opt/bin/php-cgi
 6881  6447  root       D      97016 37.9   1 16.8  /opt/bin/php-cgi
 6878  6450  root       R      95576 37.3   1 16.7  /opt/bin/php-cgi
 6446  6444  root       S      98220 38.3   1  2.7  /opt/bin/php-cgi
 1023     2  root       SW           0 0.0   0  0.6  [usb-storage]
```

Die CPU hat noch Reserven – da sollte noch mehr gehen

Tuning II



Weiteres Tuning

- `mysql` statt `sqlite` => error

```
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key  
was too long; max key length is 1000 bytes
```

- Redis für objekt-cache und locking
 - `php5-redis` nicht verfügbar als opkg

Beide Probleme sollten zu lösen sein
aber Zeit & Geduld waren am Ende ...

Fazit 0.1



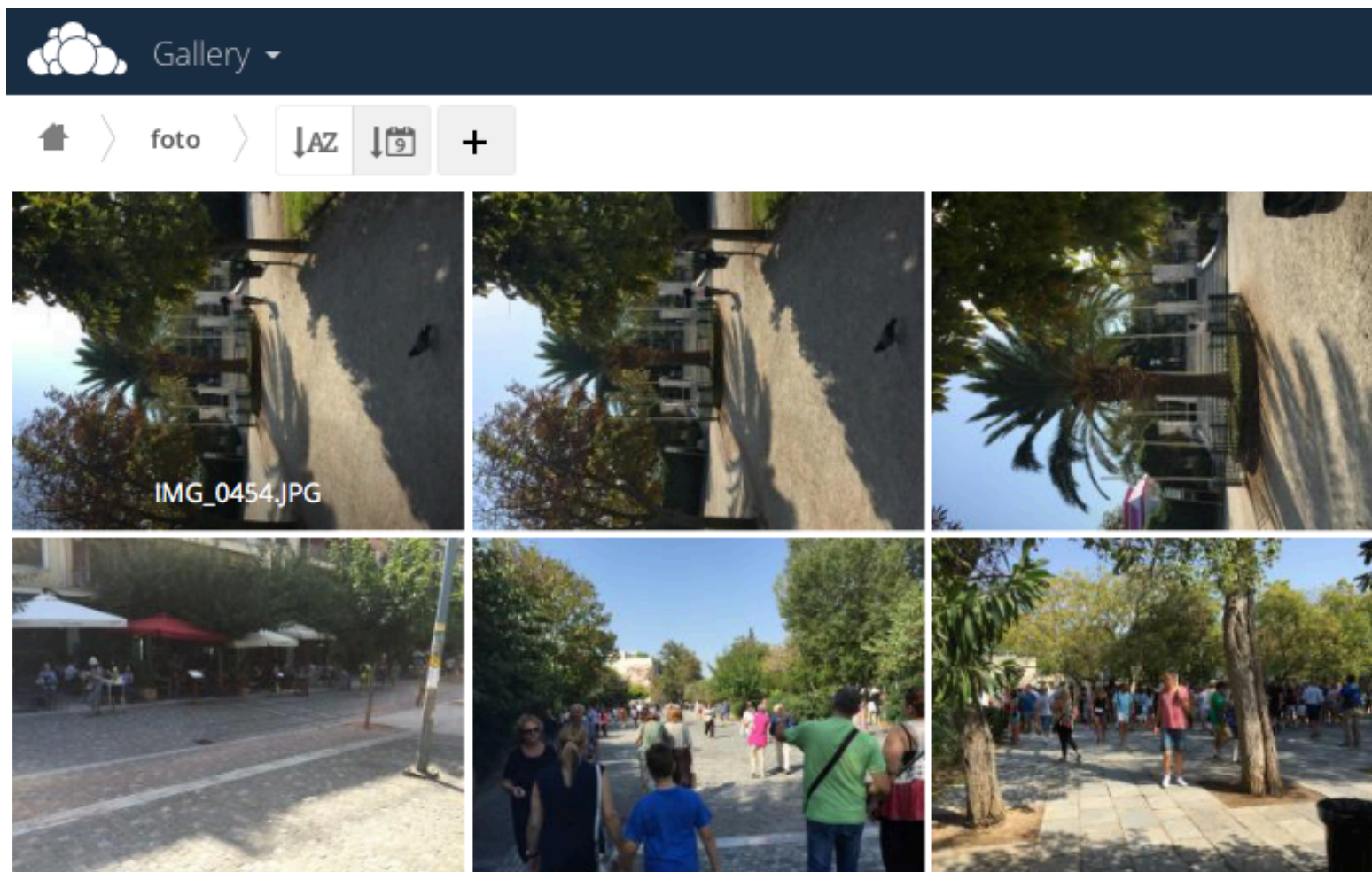
Nach der ersten Runde tuning ist das System benutzbar
Auch bei subjektiven 'wie lange braucht die Seite' Tests.

Weitere Verbesserungen um Faktor 2x scheinen plausibel
Kosten aber mehr Arbeit

Ich kann mir den Einsatz vorstellen, wenn der Router nicht das primäre System ist:
Wenn bei jeder 'Spielerei' Telefon und Internet wackeln macht es keinen Spass.

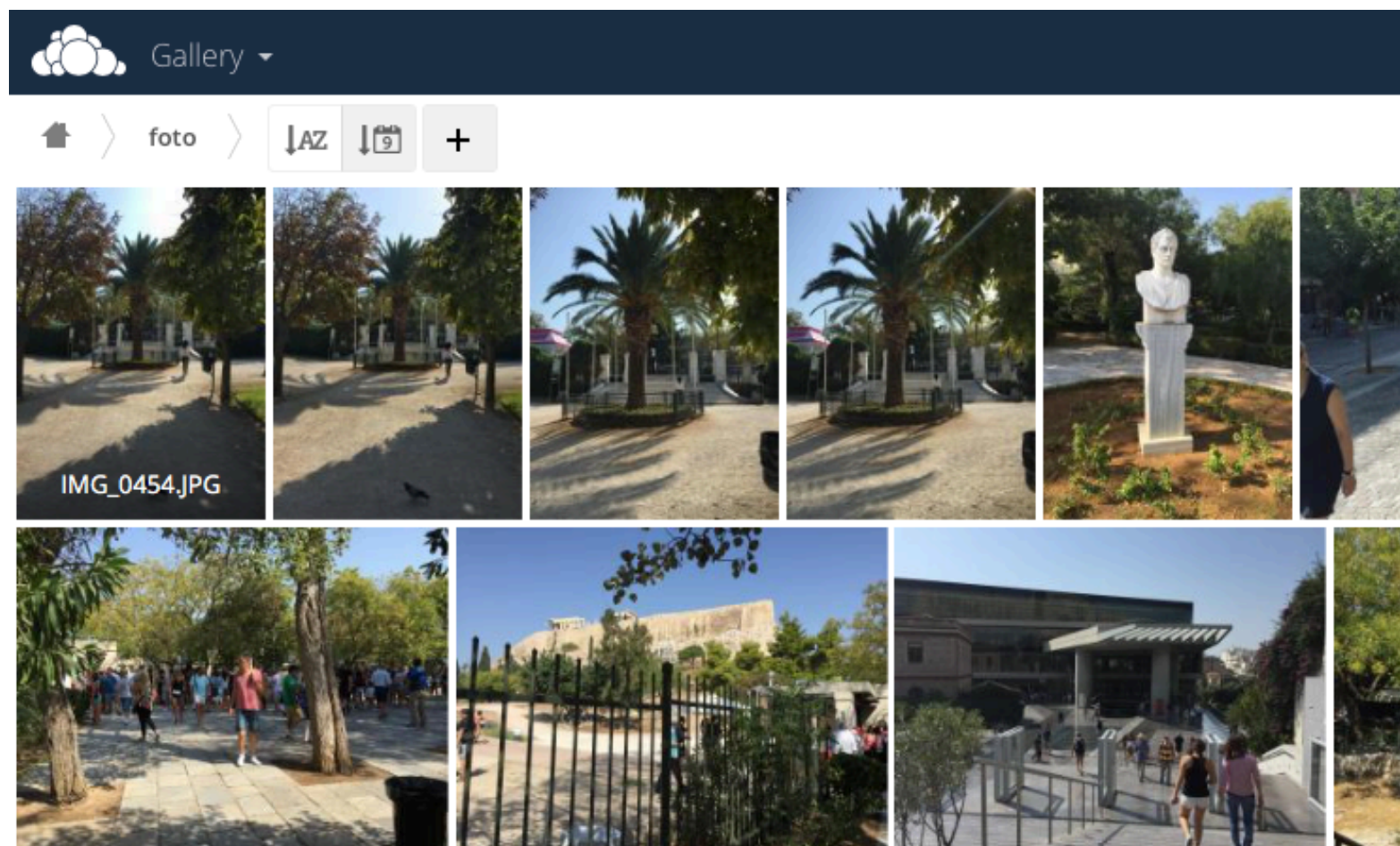
Gerade als dieses Fazit fertig wurde ...

Oops - OpenWRT



KNF Kongress 2016, Martin Bokämper

Server on Debian 8



KNF Kongress 2016, Martin Bokämper

Fazit 1.0



opkg Bibliotheken hängen dem Rest der Welt zum teil deutlich hinterher.
Abhängigkeiten muss man nicht selten selbst bauen.

In den regulären (Server-)distributionen (CentOS, Debian, Ubuntu, SUSE ...) werden entsprechende Pakete besser gepflegt und es gibt die Wahl: Falls ein bestimmter Dienst unter CentOS besser integriert ist, einfach einen Container (z.B. LXC) damit starten ...

Was zur Frage 'Container unter OpenWRT' führt
=> es geht, erfordert heute aber custom kernel und einigen Aufwand
[und sprengt hier den Rahmen]

Vielen Dank! Gibt es Fragen ?