Mini-PC als Heimwolke für LINUX und Windows

axel.keller@mucl.de

Sicherungsarten

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

Was?

- Betriebssystem, Anwendungen
- **persönliche Daten**

Wo?

- auf direkt angeschlossenem Speicher (externe Platte)
- auf NAS (Network Attached Storage, Dateiserver)
- im Internet (Cloud)
- **im Intranet (Universal-Server → Heimwolke)**

Wie?

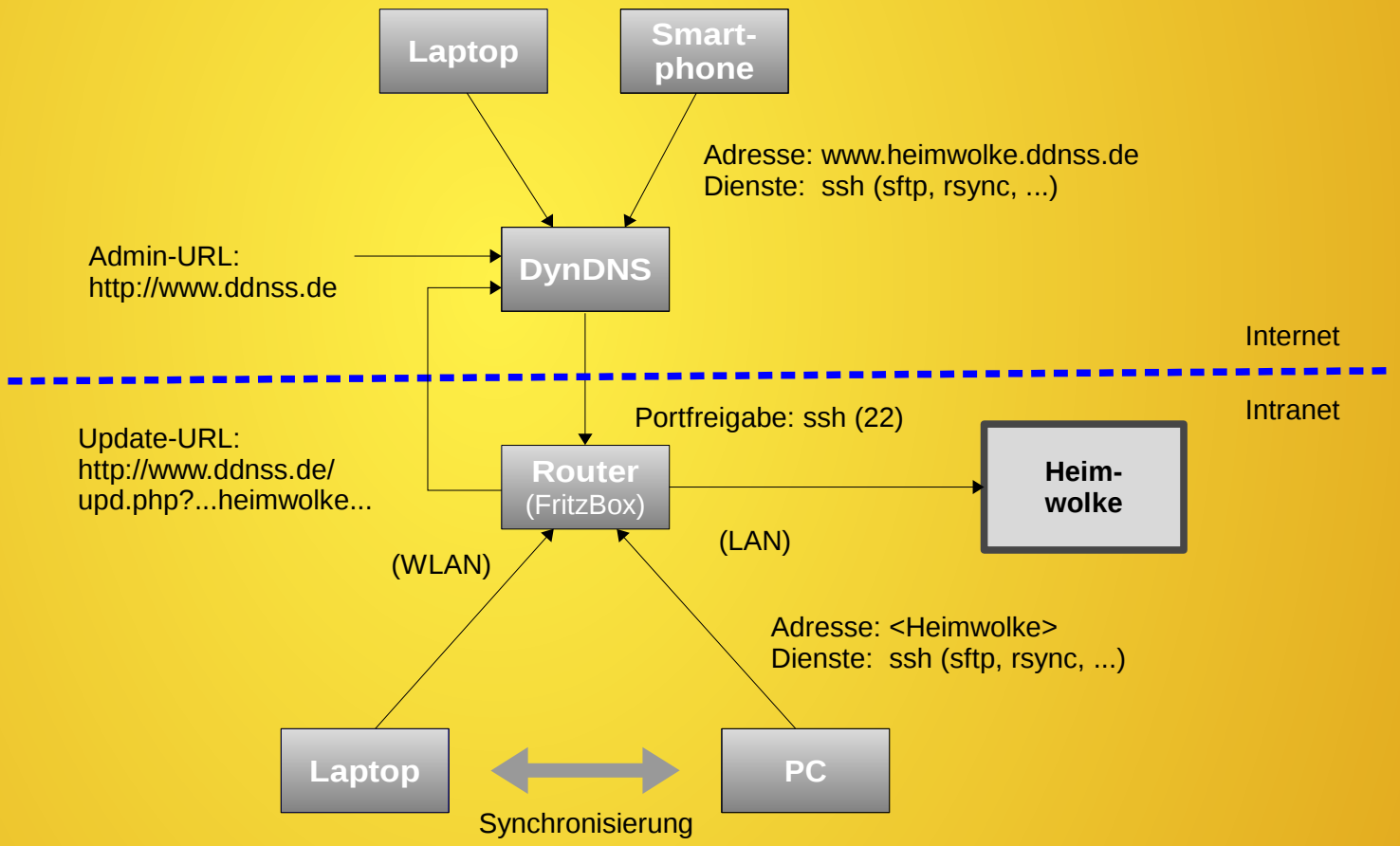
- Schnappschüsse der lokalen Dateien,
  Verwaltung mehrerer Versionen
- **Deltasicherung**

Wann?

- benutzergesteuert, Disziplin erforderlich
- **automatisch, am Ende einer Sitzung**

Internet, Intranet (LAN)

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat



Universal-Server (Mini-PCs)

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

	Nanos G20	Raspberry Pi 2
Quelle	`www.ledato.de`	`www.raspberrypi.org/products/`
CPU	Atmel® ARM AT91SAM9G20, 400 MHz	Quad-Core ARM Cortex-A7, 900 MHz
Speicher	512 MB Flash, 128 MB SDRAM, **Micro-SD card**	1GB LPDDR2 SDRAM, **Micro-SD card**
Strom	< 1 W at full system load	0,5 W < x < 1 W
Anschlüsse	Ethernet, 2 x USB, RS232	Ethernet , 4 x USB, Full HDMI
Grafik	–	VideoCore IV 3D graphics
Betriebs-system	Debian Linux „Wheezy“	• Raspbian (Debian Linux „Jessie“) • **Ubuntu MATE** • Windows 10 • ...
Preis	ca. 200 € (mit 64 GB)	ca. 100 € (mit 128 GB)

Formatierung und Konfiguration der Micro-SD-Karte

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

Benötigt: Filesystem mit verlässlichen Zeitstempeln und Zugriffsrechten als Basis für eine Delta-Synchronisation

	Nanos G20	Raspberry Pi 2
File-system	EXT3 (Root File System)	EXT4 (Root File System) + FAT (Boot-Partition)
Betriebs-system	Dateien mit *Root File System* werden kopiert	Filesystem-Image mit *Root File System* wird kopiert
System-Software	**rsync openssh**	
Erster Kontakt	Konsolverbindung über LAN: `ssh root@192.168.178.?`	Konsolverbindung über LAN: `ssh pi@raspberrypi`
Fernbe-dienung	–	VNC-Verbindung über LAN

ssh ohne Passwort-Abfrage

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

- **rsync** kann man über eine ssh-Verbindung betreiben. Dabei muss im Server (Heimwolke) kein rsyncd (deamon) installiert und gestartet sein.
- **ssh** kann man mit einer asymmetrischen Passwort-Verschlüsselung einrichten. Damit vermeidet man eine Passwort-Abfrage bei jedem direkten und indirekten Aufruf von ssh.

Dies ist erforderlich, wenn z.B. nach Abschluss einer Sitzung automatisch eine Synchronisation durchgeführt werden soll. Da die Bedienoberfläche geschlossen ist, kann man ggf. kein Passwort mehr eingeben.

<u>Privater Schlüssel</u>

Linux-PC: `/home/<benutzer>/.ssh/id_rsa`

Windows-PC: `C:cygwin[64]\home\<benutzer>\.ssh\id_rsa`

<u>Öffentlicher Schlüssel</u>

Heimwolke: `/root/.ssh/id_rsa.pub`

Synchronisierungsanweisungen

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

LINUX (push)

```
rsync -av --delete --include='.thunderbird' \
   --exclude='.*' \
   home/<user>/ root@wolke:/Ablage/Home/
```

LINUX (pull)

```
rsync -av --delete --include='.thunderbird' \
   --exclude='.*' \
   root@wolke:/Ablage/Home/ /home/<user>/
```

Windows (push)

```
c:\cygwin64\bin\bash -l -c
   "/cygdrive/C/cygwin64/bin/rsync -av --delete
   /cygdrive/C/Users/<user>/ root@wolke:/Windows-
Ablage/"
```

Windows (pull)

```
c:\cygwin64\bin\bash -l -c
   "/cygdrive/C/cygwin64/bin/rsync -av --delete
   root@wolke:/Windows-Ablage/
/cygdrive/C/Users/<user>/"
```

Externer Zugriff über DynDNS

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

Da man als Privatkunde vom DSL-Betreiber täglich eine neue IP-Adresse bekommt, benötigt man die Dienste eines **Dynamic Domain Name Services**. Hierfür gibt es viele kostenlose Anbieter.

- Benutzer-Account beim Diensteanbieter einrichten
- Subdomäne definieren, z.B. **`meinewolke.ddnss.de`**

Hat man zu Hause z.B. eine **Fritzbox** als Router, kann man diesen so konfigurieren, dass bei jeder Änderung der IP-Adresse eine Benachrichtigung darüber an den Dynamic Domain Name Service gesendet wird.

- Update-Link unter *Internet → Freigaben → Dynamic DNS*

Außerdem muss eine Portfreigabe für den **Port 22 (ssh)** eingerichtet werden, die alle Kontakte von außen an diesen Port intern an die „Heimwolke“ weiterleitet.

- Portfreigabe unter *Internet → Freigaben → Portfreigaben*:

Automatische Aufrufe

1. Zweck
2. Netz
3. Server
4. SD-Karte
5. Passwort
6. RSync
7. DynDNS
8. Automat

Sitzungsbezogene Synchronisierungen:

LINUX/KDE

Hier können Skripts für die **Anmeldung** und für die **Abmeldung** hinterlegt werden.

Startmenü → Arbeitsplatz einrichten → Systemverwaltung → Starten und Beenden → Autostart

Windows

Hier können im *Local Group Policy Editor (?)* Scripts für **Startup** und **Shutdown** hinterlegt werden.

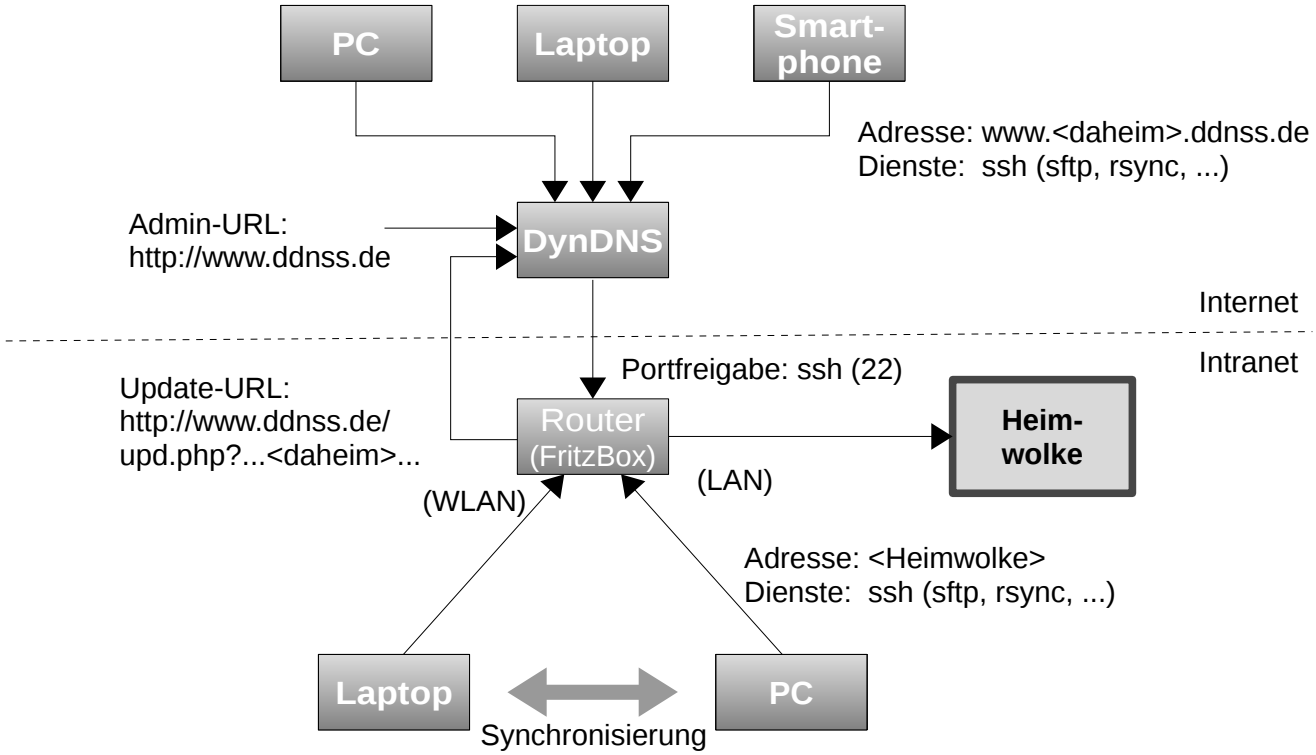
Dr. Axel Keller
26.01.2016

Mini-PC als Heimwolke für Linux und Windows

Zweck

Es gibt heute viele Angebote, seine bisher auf einem PC gespeicherten, persönlichen Daten über das Internet in einer „Cloud“ zu halten bzw. zu sichern, um diese Daten von überall her zugänglich zu haben, egal wo man ist. Inwieweit diese Daten vor Dritten geschützt sind, darüber mag man spekulieren. Bis zu einem geringen Speichervolumen, etwa 2 GB, sind viele (Lock-)Angebote noch kostenfrei. Bei höherem Bedarf kommen dann Kosten zwischen 1 € und 20 € / 100 GB und Monat.

Eine alternative Möglichkeit hierzu bieten sogenannte „Home Clouds“, also „Heimwolken“, deren Infrastruktur man bei sich zu Hause betreibt. Die hier vorgestellte Lösung kostet weniger als 200 € und dient nicht nur als Sicherungsserver für den eigenen PC, sondern ermöglicht auch die Synchronisierung aller Anwenderdaten zwischen mehreren Geräten (PCs, Laptops). Mit einem Fernzugang auf die Heimwolke über einen dynamischen Domänen-Namens-Service (DynDNS) hat man von überall her Zugriff alle seine persönlichen Daten.



Dr. Axel Keller
26.01.2016

Mini-PC als Sicherungsserver

Bei der Auswahl des Sicherungsservers kam es hier darauf an, ein Gerät zu finden, das sehr wenig Strom verbraucht und deshalb keine rotierenden Speicherplatten verwendet. Ein solches ist der Mini-PC **NanosG20** von *Ledato.* Informationen und Preise über den Mini-PC NanosG20 findet man unter

`http://www.ledato.de/` → Produkte: NanosG20

NanosG20 (180 – 200 €)

- CPU: Atmel® ARM AT91SAM9G20 microcontroller
- 400 MHz, 512 MB Flash and 128 MB SDRAM
- Energy consumption < 1W at full system load
- Ethernet, MicroSD, 2 x USB host, RS232 and RS485
- OS Debian Linux „Wheezy“

Man kann das Gerät mit Micro-SD-Karte kaufen, auf denen das Betriebssystem (Debian) vorinstalliert ist. Allerdings haben diese Karten wenig Speicher und sind dafür recht teuer (*4 GB*: 28,56€, *8 GB*: 42,84 €). Dabei ist es möglich, Micro-SD-Karten bis zu 64 GB zu verwenden (20 €).

Alternative: Man kauft sich eine Micro-SD-Karte formatiert und konfiguriert sie selbst (s.u.)

Ein anderes ist der **Raspberry Pi 2** von der *Raspberry Pi Foundation.* Informationen über den Mini-PC NanosG20 findet man unter

`http://www.raspberrypi.org/products/raspberry-pi-2-model-b/`

Peise findet bei Amazon.

Raspberry Pi 2 Model B (ca. 50 – 60 €)

- CPU: Quad-Core ARM Cortex-A7
- 900MHz, 1GB LPDDR2 SDRAM
- Energy consumption: between 0,5 W and 1 W
- OS Raspbian Linux (Debian Jessie)
- Ethernet , 4 USB
- Full HDMI port
- VideoCore IV 3D graphics core
- Micro SD card slot

Diese Gerät kann mit einer Micro-SD mit bis zu 128 GB (45 €) betrieben werden.

Formatierung und Konfiguration der Micro-SD-Karte

Das Linux-System (Debian) arbeitet verlässlich mit einem Linux-kompatiblen Dateisystem wie z.B. EXT3. Windows bietet standardmäßig nur NTFS oder FAT zum Formatieren an. Linux könnte auch FAT, das aber unterstützt keine verlässlichen **Zeitstempel** und **Zugriffsrechten**, ist für das Betriebssystem und für Sicherungsdateien daher ungeeignet. NTFS leidet unter Fragmentierung und ist proprietär.

Für den **NanosG20** kann die SD-Karte z.B. mit EXT3 formatiert werden (EXT4 noch nicht ausprobiert). Da die Mini-PCs kein BIOS besitzen, mit Hilfe dessen das Gerät von einem Datenträger gebootet bzw. installiert werden kann, muss das Betriebssystem auf die Micro-SD-Karte kopiert werden. Bei Linux-Systemen ist das das sogenannte *Root File System*.

Auf den **Raspberry Pi 2** wird gleich ein *Disk-Image* übertragen, das zwei Partitionen enthält: eine kleine zum Booten (FAT) und eine etwa 4 GB große mit dem *Root File System* (EXT4).

Auf Linux-Arbeitsplätzen geht das mit einfachen Bordmitteln. Auf Windows-Arbeitsplätzen müssen dazu noch Hilfswerkzeuge abgerufen werden, mit denen dann auf der Micro-SD-Karte das *Image* (raspbian.img) übertragen wird.

Sicherungsmethode

Die Synchronisierung zwischen PC und Mini-PC (Heimwolke) erfolgt mit der Dateisicherungswerkzeug **rsync** (lange erprobtes Werkzeug aus der `r*`-Familie der Unix-Welt). Man betreibt die Synchronisierung sinnvollerweise als **Deltasicherung**. Wenn man die rsync-Verbindung zum Mini-PC über **ssh** betreibt, benötigt man auf dem Mini-PC keinen rsync-Dämonen (rsyncd), sondern nur die Client-Version.

Eine allgemeine Einführung zur Verwendung von **rsync** zur Dateisynchronisation mit sicherer Verbindung über **ssh** unter Windows findet man unter:

http://www.webmasterpro.de/server/article/backup-sicheres-rsync-mit-cygwin-auf-windows.html

Unter Linux sind **rsync** und **ssh** standardmäßig vorhanden. Für Windows sind sie als **cygwin**-Portierungen verfügbar und müssen installiert werden.

Schritt für Schritt

Die folgenden Schritte sind ein Leitfaden zur Installation und Konfiguration eines Mini-PC als Heimwolke. Sie sind getrennt nach Mini-PCs (NanosG20 oder Raspi2) und nach verwendeten Arbeitsplätzen (Linux und Windows).

1. Micro-SD-Karte konfigurieren

(am LINUX-Arbeitsplatz für NanosG20)

- **Software** abrufen
  - Betriebssystem unter `http://www.ledato.de/`
    → Neues Update für NanosG20 Mini-PC: Das neue Root File System

    `nanosg20-rootfs-debian-wheezy.tar.bz2` (62,5 MB, Version „Wheezy“)
  - unter `http://www.packages.debian.org/wheezy/rsync` die Datei

    `rsync_3.0.9-4_armel.deb`
- **SD-Karte** (micro 64 GB)
  - am LINUX-Arbeitsplatz in USB-Anschluss einsteckenmit Dateibrowser öffnen und mit dem Kommando

    ```
    # df -k
    ```

    überprüfen, an welcher Gerätedatei die Karte angeschlossen ist (z.B. /dev/sdb1)
  - SD-Karte am LINUX-Arbeitsplatz erneut einstecken, aber nicht mit Dateibrowser öffnen, dann mit EXT3 formatieren:

    ```
    # mkfs.ext3 /dev/sdb1
    ```
  - SD-Karte am LINUX-Arbeitsplatz erneut einstecken
  - mit Dateibrowser (Systemverwalter) öffnen: `/run/media/<user>/???`
  - **`nanosg20-rootfs-debian-wheezy.tar.bz2`** im Dateibrowser (Systemverwalter) mit *Ark* öffnen und auf Micro-SD-Karte exportieren unter `/run/media/<user>/???/`
  - Datei **`rsync_3.0.9-4_armel.deb`** auf Micro-SD-Karte nach `/run/media/<user>/???/`**`opt`** kopieren

(am LINUX-Arbeitsplatz für Raspberry Pi 2)

- **Software**
  - Betriebssystem **Raspbian** abrufen unter
    `http://downloads.raspberrypi.org/raspbian_latest`
    `2015-11-21-raspbian-jessie.zip` (1,3 GB, Version „Jessie“)
  - Daraus das SD-Karten Image **`2015-11-21-raspbian-jessie.img`** expandieren

- **SD-Karte** (micro 64/128 GB)
  - am LINUX-Arbeitsplatz in USB-Anschluss einstecken und nicht mit Dateibrowser öffnen
  - SD-Karten Image auf Micro-SD-Karte übertragen:

```
# dd bs=4M if=2015-11-21-raspbian-jessie.img of=/dev/sdb
```

    Damit wird eine kleine boot-Partition (FAT) und eine 4 GB große Systempartition (EXT4) auf die SD-Karte geschrieben.
  - Systempartition (/dev/sdb2) mit **YaST**-Partitionierer auf maximale Größe erweitern. (Kann später auch mit Raspian über "**raspi-config**" gemacht werden)

(am Windows-Arbeitsplatz für Raspberry Pi 2)

- **Software** abrufen
  - Betriebssystem **Raspbian** unter
    `http://downloads.raspberrypi.org/raspbian_latest`

    `2015-11-21-raspbian-jessie.zip` (1,3 GB, Version „Jessie")
  - SD Formatierprogramm unter
    `http://www.sdcard.org/downloads/formatter_4/`
  - Win32 Disk Imager unter
    `http://sourceforge.net/projects/win32diskimager/`
- **SD-Karte** (micro 64/128 GB) am Windows-Arbeitsplatz in USB-Anschluss einstecken und nicht mit Dateibrowser öffnen, dann z.B. mit SD Formatter 4.0 formatieren (FAT???)
- **SD-Karte** (micro 64/128 GB) am Windows-Arbeitsplatz erneut einstecken
  - SD-Karten Image **`2015-11-21-raspbian-jessie.img`** aus **Raspbian** expandieren
  - Image mit Win32 Disk Imager auf die Micro-SD-Karte übertragen

2. Software-Vorbereitungen für Windows-Arbeitsplätze

- **cygwin:** `setup-x86_64.exe` aus dem Internet (z.B. von `www.heise.de/download/cygwin.html`) abrufen und installieren,
  dabei unter Paket NET die Komponenten **openssh** und **rsync** auswählen!
- **WinSCP (Datei Browser):** `winscp576setup.exe` aus dem Internet (z.B. von `www-w.winscp.net/eng/docs/lang:de`) abrufen und installieren (Installer Package)

3. NanosG20 installieren und konfigurieren

- **Mini-PC** einrichten:
  - Gerät zusammenbauen (schrauben)
  - Micro-SD-Karte in den Schlitz einstecken
  - LAN-Kabel einstecken
  - mit Strom versorgen
  - Vom Router (FritzBox) die zugewiesene **IP-Adresse** (DHCP) ermitteln und für diese IP-Adresse einen Namen vergeben, z.B. „**wolke**“

Alle folgenden Kommandos erfolgen über eine LINUX-Konsole (bash) oder ein Cygwin64-Terminal in Windows.

- **Konsolverbindung über LAN** mit Debian-System:

```
# ssh root@wolke
Passwort: ledato
```

  - Neues Passwort für *root* setzen:

```
debian:~# passwd root
<password>
<password>
```

  - Uhr setzen:

```
debian:~# date -u `date -u '+%m%d%H%M%Y.%S'`
debian:~# hwclock -w -u
```

  - **`rsync`** installieren:

```
debian:~# dpkg -i /opt/rsync_3.0.9-4_armel.deb
```

  - Sicherungsverzeichnisse einrichten

```
debian:~# mkdir /Ablage
debian:~# mkdir /Ablage/Home
debian:~# mkdir /Windows-Ablage
...
```

  - Debian-System aktualisieren

```
debian:~# apt-get update && apt-get upgrade -y
```

  - Blinkende LED am NanosG20 ausschalten:

```
debian:~# echo 0 > /sys/class/leds/LED4/brightness
```

- **`ssh` ohne Passwort-Abfrage** für Benutzer einrichten, damit `rsync` im Hintergrund laufen kann. Ansonsten erscheint auf der Bedienoberfläche eine Aufforderung zur Passworteingabe, was nach Abmeldung von der Sitzung nicht mehr bedienbar ist.
  – Unter der Benutzer-Kennung am LINUX-Arbeitsplatz ein Key-Pair erzeugen:

```
> ssh-keygen -t rsa
```

  – Öffentlichen Schlüssel auf Debian kopieren/anhängen

```
> scp ~/.ssh/id_rsa.pub  root@wolke:luser.pub.tmp
Passwort: <root-password>
> ssh root@wolke
Passwort: <root-password>
debian:~# cat luser.pub.tmp >> .ssh/authorized_keys
debian:~# chmod 700 .ssh
debian:~# chmod 600 .ssh/authorized_keys
debian:~# rm  luser.pub.tmp
```

4. Raspbery Pi 2 installieren und konfigurieren

- **Mini-PC** einrichten:
  - Gerät zusammenbauen
  - Micro-SD-Karte in den Schlitz einstecken
  - LAN-Kabel einstecken
  - mit Strom versorgen
  - Im Router (FritzBox) den der **IP-Adresse** (DHCP) zugewiesenen Namen ermitteln „**raspberrypi**“ und ggf. einen neuen Namen vergeben, z.B. „**wolke**“

Alle folgenden Kommandos erfolgen über eine LINUX-Konsole (bash) oder ein Cygwin64-Terminal in Windows.

- **Konsolverbindung über LAN** mit Raspbian-System:

```
# ssh pi@raspberrypi
Password: raspberry
```

  - Passwort für *root* setzen:

```
pi@raspberrypi:~$ sudo passwd root
<root-password>
<root-password>
pi@raspberrypi:~$ su
Password: <root-password>
root@raspberrypi: #
```

  - Sicherungsverzeichnisse einrichten

```
root@raspberrypi:~# mkdir /Ablage
root@raspberrypi:~# mkdir /Ablage/Home
root@raspberrypi:~# mkdir /Windows-Ablage
...
```

  - Raspbian-System aktualisieren

```
root@raspberrypi:~# apt-get update && apt-get upgrade -y
```

- **`ssh` ohne Passwort-Abfrage** für Benutzer einrichten, damit `rsync` im Hintergrund laufen kann. Ansonsten erscheint auf der Bedienoberfläche eine Aufforderung zur Passworteingabe, was nach Abmeldung von der Sitzung nicht mehr bedienbar ist.
  - Unter der Benutzer-Kennung am LINUX-Arbeitsplatz ein Key-Pair erzeugen:

```
> ssh-keygen -t rsa
```

– Öffentlichen Schlüssel auf Raspbian kopieren/anhängen
(erst für Benutzer **pi**, dann für **root**)

```
> scp ~/.ssh/id_rsa.pub  pi@raspberrypi:luser.pub.tmp
Passwort: raspberry
> ssh pi@raspberrypi
Password: raspberry
pi@raspberrypi:~# cat luser.pub.tmp >> .ssh/authorized_keys
pi@raspberrypi:~# chmod 700 .ssh
pi@raspberrypi:~# chmod 600 .ssh/authorized_keys
pi@raspberrypi:~# rm  luser.pub.tmp

pi@raspberrypi:~# su -
Password: <root-password>
root@raspberrypi:~# mkdir .ssh
root@raspberrypi:~# cp /home/pi/.ssh/authorized_keys  .ssh
root@raspberrypi:~# chmod 700 .ssh
root@raspberrypi:~# chmod 600 .ssh/authorized_keys
```

- Grafische Fernsteuerung über **vnc**

```
root@raspberrypi:~# apt-get install tightvncserver
…
root@raspberrypi:~# tightvncserver          (Passwort definieren)
Password: <vnc-password>
Verify:   <vnc-password>
root@raspberrypi:~# raspi-config
...
```

→ Boot Options → Desktop → reboot

5. Externer Zugriff auf die Heimwolke über DynDNS / FRITZ!Box

- Anmeldung und Aktivierung unter **`www.ddnss.de`**
  - Benutzer-Account einrichten (kostenlos)
  - In *Domain-Verwaltung → Neu Erstellen* eigenen Domänennamen festlegen:
    <daheim>.ddnss.de
- **Update-Link** in der *Fritzbox* eintragen unter *Internet → Freigaben → Dynamic DNS*:

  DNS-Anbieter: **Benutzerdefiniert**
  Update-URL : (mit vorgegebenen Platzhaltern <...>)
  Domainname: **<daheim>**
  Benutzer/Kennwort: **<username>** / **<pass>**
- **Portfreigabe** in der *Fritzbox* eintragen unter *Internet → Freigaben → Portfreigaben*:

  Bezeichnung: **SSH-Server**
  Protokoll/Port: **TCP/22** oder **HTTPS/443**
  an Computer/Port: **wolke/22**
- Externer Zugriff auf „Wolke“ über Kommando (Port 443, wenn Port 22 gesperrt)

```
~> sftp [-oPort=443] <daheim>.ddnss.de
```

  oder über URL: **`sftp://<daheim>.ddnss.de[:443]`**

6. Synchronisierungsanweisungen

(Intranet)

Die Optionen *-av* bestimmen eine Deltasynchronisation, durch die Option *-delete* wird veranlasst, Dateien am Ziel löschen, die an der Quelle nicht vorhanden sind.

- Push (Sicherung zur Wolke):

  LINUX

  ```
  ~> rsync -av --delete /Wurzel/verz1 /Wurzel/verz2 … \
                root@wolke:/Ablage/
  ~> rsync -av --delete --include='.thunderbird' --exclude='.*' \
                home/<user>/ \
                root@wolke:/Ablage/Home/
  ```

  Windows

  ```
  c:\cygwin64\bin\bash -l -c
     "/cygdrive/C/cygwin64/bin/rsync -av --delete
     /cygdrive/C/Users/<user>/verz1 root@wolke:/Windows-Ablage/"
  ```

- Pull (Restaurierung von Wolke):

  LINUX

  ```
  ~> rsync -av --delete root@wolke:/Ablage/verz1 :/Ablage/verz2 … \
                /Wurzel/
  ~> rsync -av --delete --include='.thunderbird' --exclude='.*' \
                root@wolke:/Ablage/Home/ \
                /home/<user>/
  ```

  Windows

  ```
  c:\cygwin64\bin\bash -l -c
     "/cygdrive/C/cygwin64/bin/rsync -av --delete
     root@wolke:/Windows-Ablage/verz1 /cygdrive/C/Users/<user>/"
  ```

Die jeweils zweite Anweisung gibt ein Beispiel, wie in Linux die Thunderbird-Postfächer unter `/home/<user>/.thunderbird/` gesichert werden, die restlichen `.`-Dateien aber nicht.

(Internet)

Die Optionen *-av* bestimmen eine Deltasynchronisation, durch die Option *-delete* wird veranlasst, Dateien am Ziel löschen, die an der Quelle nicht vorhanden sind.

Das Synchronisierungswerkzeug **rsync** nutzt als Verbindungsbasis **ssh**, das üblicherweise den Port 22 benutzt. Wenn im externen WLAN der Port 22 (ssh) gesperrt ist, der Port 443 (https) aber frei ist und im Intranet-Router Port 443 auf Port 22 umgelenkt wird, kann mit „`-e 'ssh -D 443'`“ für die Verbindung Port 443 als Alternative verwendet werden.

- Push (Sicherung zur Wolke):

LINUX

```
~> rsync -av –delete [-e 'ssh -D 443'] /Wurzel/verz1 /Wurzel/verz2 … \
                     root@heimwolke.ddnss.de:/Ablage/
~> rsync -av –delete [-e 'ssh -D 443'] \
        --include='.thunderbird' --exclude='.*' \
                     /home/<user>/ \
                     root@heimwolke.ddnss.de:/Ablage/Home/
```

- Pull (Restaurierung von Wolke):

LINUX

```
~> rsync -av --delete [-e 'ssh -D 443'] root@heimwolke.ddnss.de \
                     :/Ablage/verz1 :/Ablage/verz2 …  /Wurzel/
~> rsync -av --delete [-e 'ssh -D 443'] \
        --include='.thunderbird' --exclude='.*' \
                     root@heimwolke.ddnss.de:/Ablage/Home/ \
                     /home/<user>/
```

7. Automatische Aufrufe

Um die Sicherungs- bzw. Restaurierungsanweisungen nicht regelmäßig manuell aufrufen zu müssen, empfiehlt es sich, diese über Skripts auszuführen, die man beim Beenden bzw. Starten von Sitzungen aufrufen lässt. Dazu gibt es unter Linux/KDE und Windows entsprechende Möglichkeiten.

Linux/KDE

Startmenü → Arbeitsfläche einrichten (Systemeinstellungen) → Systemverwaltung → Starten und Beenden → Autostart: **Skript hinzufügen …**

– Skriptdatei: **auswählen**

– Ausführungszeitpunkt bestimmen: **Anmeldung** / **Abmeldung**

OK

Windows

Open the Local Group Policy Editor:

– In the console tree, click **Scripts (Startup/Shutdown)** . The path is **Computer Configuration\Windows Settings\Scripts (Startup/Shutdown)** .

– In the results pane, double-click **Startup** resp. **Shutdown** .

– In the **Startup Properties** resp. **Shutdown Properties** dialog box, click **Add** .

– In the **Add a Script** dialog box, do the following:

  • In **Script Name** , type the path to the script, or click **Browse** to search for the script file in the Netlogon shared folder on the domain controller.

  • In **Script Parameters** , type any parameters that you want, the same way as you would type them on the command line. For example, if your script includes parameters called //logo (display banner) and //I (interactive mode), type **//logo //I** .

– In the **Shutdown Properties** dialog box, specify the options that you want:

  • **Startup** resp. **Shutdown Scripts for <Group Policy object>** : Lists all the scripts that are currently assigned to the selected Group Policy object (GPO). If you assign multiple scripts, the scripts are processed in the order that you specify. To move a script up in the list, click it and then click **Up** . To move a script down in the list, click it and then click **Down** .

  • **Add** : Opens the **Add a Script** dialog box, where you can specify any additional scripts to use.

  • **Edit** : Opens the **Edit Script** dialog box, where you can modify script information, such as name and parameters.

  • **Remove** : Removes the selected script from the **Startup** resp. **Shutdown Scripts** list.

  • **Show Files** : Displays the script files that are stored in the selected GPO.