Wiki / NFS

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

- **Ubuntu 14.04** Trusty Tahr
- **Ubuntu 12.04** Precise Pangolin

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Installation von Programmen**
2. **Ein Terminal öffnen**
3. **Einen Editor öffnen**
4. **fstab**

Inhaltsverzeichnis

[//media-cdn.ubuntu-de.org/wiki/attachments/59/28/service.png] **NFS** [http://de.wikipedia.org/wiki/Network_File_System] (Network File System) ist ein stabiles und gut funktionierendes Netzwerk-Protokoll von Sun, um Dateien über das lokale Netzwerk auszutauschen. Prinzipiell würde es auch über das Internet funktionieren, was aber aus Sicherheitsgründen nicht zu empfehlen ist. NFS ist im Prinzip das *NIX-Pendant zu **SMB** [https://wiki.ubuntuusers.de/Samba/] aus der Windows-Welt.

Einsatzszenario

NFS setzt für einen reibungslosen und sicheren Betrieb voraus, dass

1. alle Benutzer im Netzwerk eindeutige **UIDs** [https://wiki.ubuntuusers.de/Benutzer_und_Gruppen/#Nummerierung-der-UID-GID] haben und
2. alle Rechner im Netzwerk zentral administriert werden

Die Rechner müssen also so konfiguriert werden, dass jeder Benutzer netzweit seine eigene feste, numerische UID erhält, die auf allen Rechnern dann gleich ist. Bei größeren Netzwerken stellt man das mit einem **LDAP** [http://de.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol]- oder **NIS** [http://de.wikipedia.org/wiki/Network_Information_Service]-Server sicher. Die Zugriffskontrolle auf die einzelnen Dateien geschieht dann auf dem Server über das reguläre Dateiberechtigungssystem.

Wenn die Benutzer Root-Rechte auf ihren eigenen Rechnern haben bzw. ihre eigenen Notebooks ins Netz einbinden dürfen, können sie das aber umgehen und sich auf ihren Rechnern beliebige UIDs besorgen, die vom NFS-Server auch nicht weiter getestet werden. In diesem Fall muss dann entweder ein zusätzliches Sicherheitsprotokoll wie Kerberos zum Einsatz kommen (nicht-trivial) oder gleich **Samba** [https://wiki.ubuntuusers.de/Samba/] benutzt werden (langsamer).

Installation

Sollte NFS noch nicht vorhanden sein, lässt es sich sehr schnell installieren. Folgende Pakete und deren Abhängigkeiten müssen über die Paketverwaltung [1] installiert werden:

Wenn der Rechner als Server dienen soll, der Dateien bereitstellt:

- **nfs-kernel-server**

Jetzt installieren [apt://nfs-kernel-server] mit **apturl** [https://wiki.ubuntuusers.de/apturl/]

```
sudo apt-get install nfs-kernel-server
```

Wenn der Rechner nur als Client agieren soll, der auf andere Freigaben zugreift:

- **nfs-common**

Jetzt installieren [apt://nfs-common] mit **apturl** [https://wiki.ubuntuusers.de/apturl/]

```
sudo apt-get install nfs-common
```

Freigaben

Die Freigaben von Verzeichnissen und Dateien auf dem Server lassen sich durch direkte Bearbeitung der Datei **/etc/exports** verwalten. Dazu muss diese Datei angelegt und/oder bearbeitet werden [3]. Die Freigabe eines Verzeichnisses lässt sich mit einer Zeile nach folgendem Muster anlegen:

```
<pfad>   <computername>(<optionen>)
```

Hier sind einige Beispiele:

```
# freigabe1 wird für zwei Rechner freigegeben
# notebook darf nur lesen (ro)
# desktop darf lesen und schreiben (rw)
/pfad/zur/freigabe1      notebook(ro,async) desktop(rw,async)
```

Alternativ kann die IP-Adresse angegeben werden:

```
# Freigabe gilt nur für 192.168.1.13, jedoch nur mit Leserechten:
/pfad/zur/freigabe2      192.168.1.13(ro,async)
# Freigabe gilt für alle IPs von 192.168.1.1 bis 192.168.1.255, mit Lese-/Schreibrechten:
/pfad/zur/freigabe3       192.168.1.0/255.255.255.0(rw,async)
# Freigabe gilt nur für den Rechner mit dem Namen notebook
/pfad/zur/freigabe4      notebook(ro,async)
```

Die Parameter in den Klammern lauten:

Optionen in der /etc/exports		
Option	**Beschreibung**	**default**
rw	Lesen und Schreiben	-
ro	nur Lesen	X
sync	Synchroner Datentransfer.	X*
async	Asynchroner Datentransfer. In der Regel führt diese Option zu einer Leistungsverbesserung auf Kosten möglicher Datenverluste durch Server-Abstürze bzw. Neustarts. Bis nfs-utils 1.0.0, war die Option 'async' standardmäßig aktiviert.	-*
wdelay	Diese Option hat nur im Zusammenhang mit 'sync' einen Effekt, siehe 'no_wdelay'	X
no_wdelay	Diese Option hat nur im Zusammenhang mit 'sync' einen Effekt. Ein NFS-Server würde normalerweise einen Schreibzugriff auf die Platte etwas verschieben, wenn er davon ausgeht, daß ein anderer, verwandter Schreibzugriff gerade oder bald ankommt. Das ermöglicht es, mehrere Schreibzugriffe in einem Abwasch zu erledigen was die Performance steigern kann. Wenn ein NFS-Server hauptsächlich kleine und nicht zusammengehörige Schreibaufträge erhält, kann dieses Feature aber stören und die Performance drücken. Für diesen Fall ist die Option gedacht, um es auszuschalten. Standardmäßig ist 'wdelay' aktiviert.	-
secure	Ports oberhalb 1024 nicht verwenden.	X
insecure	Ports oberhalb 1024 auch verwenden.	-
hide	"Platzhalter"	X

nohide	Wenn unterhalb eines exportierten Verzeichnisses (z.B. **/home/user** auf /dev/hda1) ein weiteres Dateisystem eingebunden wurde (z.B. /dev/hdb1 in **/home/user/Musik**), so wird dieses Verzeichnis durch einen eigenen exports-eintrag exportiert. Im Normalfall (option 'hide') sieht der Client dieses Unterverzeichnis nicht, wenn er nur das Oberverzeichnis einbindet, weswegen er beide einbinden muss. Durch die Option 'nohide' werden die eingebundenen Unterverzeichnisse dem Client nicht mehr als eigene Partitionen präsentiert, sondern als normale, zum Oberverzeichnis gehörende Verzeichnisse. Daher muss man zwar am Server noch alles explizit exportieren, am Client aber nur noch das Oberverzeichnis einbinden. Die Option nohide funktioniert nur, wenn die Client-Angabe ein festgelegter Rechner ist. Bei Wildcards oder ganzen IP-Bereichen klappt das nicht, dort könnte die Option crossmnt zum gewünschten Erfolg führen.	-
crossmnt	Diese Option ist so ähnlich wie 'nohide'. Die Option 'crossmnt' ermöglicht es, dass Clients auf exportierte Dateisysteme innerhalb der Freigabe zugreifen können. Wenn ein Kind-Dateisystem "B" auf einem übergeordneten "A" gemounted wird, hat die Einstellung crossmnt auf "A" den gleichen Effekt wie die Einstellung "nohide" auf B.	-
subtree_check	Wenn nur einzelne Verzeichnisse eines Dateisystems freigegeben wurden, wird hiermit überprüft, ob eine vom Client angeforderte Datei in diesen Verzeichnissen des Dateisystems ist. Wurde das komplette Dateisystem freigegeben, werden die Übertragungsgeschwindigkeiten beim Deaktivieren mittels 'no_subtree_check' erhöht. Darüberhinaus stellt diese Option sicher, dass beim Einbinden eines NFS-Verzeichnisses mittels 'root_squash' alle Dateien, die dem user 'root' gehören, nicht abrufbar sind (selbst wenn die Datei-Rechte dies vorsehen). Diese Option kann jedoch Probleme verursachen, insbesondere wenn Dateien umbenannt werden, die auf dem Client gerade geöffnet sind.	-*
no_subtree_check	Diese Option schaltet das "Subtree-Checking", also die Überprüfung von Unterverzeichnisbäumen ab. Das hat zwar eine leichte Verringerung der Sicherheit zur Folge, kann aber die Verlässlichkeit unter Umständen erhöhen. Wenn ein Unterverzeichnis eines Dateisystems freigegeben (exportiert) wird, aber das ganze Dateisystem nicht, muss der NFS Server jedes mal, wenn er eine Anfrage bekommt nicht nur überprüfen, ob die gewünschte Datei in dem Dateisystem liegt (einfach), sondern auch, ob sie tatsächlich in dem Unterverzeichnis liegt (schwieriger). Diese Überprüfung wird subtree_check genannt. Um diese Überprüfung durchzuführen, muss der Server einige Informationen über die Position der Datei im Dateisystem in der Filehandle integrieren, die an den Client weitergegeben wird. Das kann zu Problemen beim Zugriff auf Dateien führen, die umbenannt werden, während sie noch von einem Client geöffnet sind. (Obwohl es in vielen einfachen Fällen weiter funktioniert). Subtree_checking wird auch benützt, um sicherzustellen, dass Dateien in Verzeichnissen auf die nur root Zugriff hat nur zugreifbar sind, wenn das Dateisystem mit der no_root_squash Option (siehe unten) exportiert wurde, auch wenn die Datei selbst weniger restriktive Zugriffsmodi hat. Als genereller Hinweis mag gelten, dass ein Home-Verzeichnis Dateibaum, der normalerweise von seiner Wurzel aus exportiert wird, und auf dem häufig Dateien umbenannt werden, ohne subtree_checking exportiert werden sollte. Ein Dateisystem, das größtenteils ReadOnly ist, und auf dem zumindest selten Dateien umbenannt werden (z.B. /usr oder /var), und aus dem Unterverzeichnisse exportiert werden, sollte mit subtree_checking versehen werden.	X*
insecure_locks no_auth_nlm	Diese Option (die beiden sind Synonyme) weist den NFS-Server an, bei Dateisperranfragen (locking, z.B. Nachfragen, die das NLM Protokoll benutzen) keine Authentifizierung zu verlangen. Normalerweise würde der NFS-Server einen Sperrmechanismus verlangen, um einen Berechtigungsnachweis für User zu verwalten, die Lesezugriff auf die Datei haben. Mit dieser Option werden keine Zugriffsüberprüfungen gemacht. Alte NFS-Client Implementationen haben keine Berechtigungsnachweise zusammen mit Sperrnachfragen verschickt und es existieren viele NFS-Clients, die auf diesen alten Architekturen basieren. Diese Option sollte also benutzt werden, wenn auffällt, dass nur Dateien gesperrt werden können, die von allen Usern lesbar sind. Die voreingestellte Option, Authentifizierung für NLM-Nachfragen zu verlangen, kann explizit mit einem der Synonymen 'auth_nlm' oder 'secure_locks' angegeben werden.	-
_netdev	The _netdev option tells the system to wait until the network is up before trying to mount the share.	-

Optionen zum UID/GID-Mapping in der /etc/exports		
root_squash	Mappt alle Anfragen der UID/GID 0 auf die UID/GID anonymous. Zu beachten ist, dass damit andere sensible bzw. mächtige UserIDs wie etwa "bin" oder "staff" nicht geändert werden.	X
no_root_squash	Bindet man per NFS Verzeichnisse ein, die auf dem Server dem User root gehören, werden diese auf den User nobody gemappt und man kann diese nicht modifizieren. Um dieses Sicherheitsfeature zu umgehen, dient der Parameter 'no_root_squash' (weitere Info mit man 5 exports). Das bedeutet, 'no_root_squash' verhindert ein Mapping der vom Nutzer root geschriebenen Files und Verteichnissen auf einen anderen Nutzer. UID und GID 0 werden erhalten.	-
all_squash	Mappt alle UserIDs auf "anonymous". Nützlich für NFS-exportierte öffentliche FTP-Verzeichnisse oder News-Spoolverzeichnisse.	-
anonuid anongid	Diese Option setzt die anonyme User- und GruppenID explizit auf die angegebenen Werte. Diese Option ist primär für PC/NFS Clients gedacht, wo davon ausgegangen wird, daß alle Nachfragen von einem bestimmten Rechner immer von einer Person kommen. Beispiel: /home/joe pc001(rw,all_squash,anonuid=150,anongid=100)	-
Optionen zur Kompabilität		
nfsvers=3 vers=3	NFS3 erzwingen. The NFS protocol version number used to contact the server's NFS service. If the server does not support the requested version, the mount request fails. If this option is not specified, the client negotiates a suitable version with the server, trying version 4 first, version 3 second, and version 2 last.	?

*: default Einstellung seit nfs-utils 1.0.1

Achtung!

- Zwischen Freigabe und der Parameterklammer darf kein Leerzeichen stehen: z.B.
  - 192.168.1.13(ro,async) und nicht
  - 192.168.1.13 (ro,async)
- insecure sollte nur verwendet werden, wenn es unbedingt notwendig ist! Da dann auch die unsicheren Ports verwendet werden. Leider verwendet das Mac OS X von Apple diese Ports für NFS-Verbindungen. Ein aktueller Apple Computer kann sich nur dann mit dem NFS-Server verbinden, wenn die Option insecure gesetzt ist.
- bei nohide sollte man beachten, dass es dadurch passieren kann, dass verschiedene Dateien, welche auf unterschiedlichen Partitionen bzw. Dateisystemen dieselbe Inode besitzen, im gemounteten Oberverzeichnis dieselbe Inode auf dem gleichen (NFS-)Dateisystem haben; manche Treiber verkraften das nicht. Ggf. führt es beim Client zu einer Kernel Panic, wenn gleichzeitig lesend und schreibend auf den Server zugegriffen wird.

Beispiel:

NFS-Freigabe für Verzeichnis **/media** für eine VirtualBox (auf Servern sollte man das Verzeichnis **/media** nicht mounten, da sich dort ja auch Wechseldatenträger automatisch einhängen und von anderen auch darauf zugegriffen werden könnte):

```
/media 192.168.56.0/24(rw,async,insecure,no_subtree_check,crossmnt)
```

Die Option crossmnt sorgt dafür, dass der Client auch auf die eingehängten Dateisysteme unterhalb des Verzeichnisses **/media** zugreifen kann, z.B.: **/media/Win7**, **/media/SD-Karte**, **/media/CD** usw.

Damit sich der Rechner notebook auch zu der Freigabe **/PFAD/ZUR/FREIGABE3** verbinden kann, muss er mit der IP-Adresse in der Datei **/etc/hosts** [3] stehen. Die Datei muss wie folgt aufgebaut sein:

```
<ip>  <computername> <computername.domain.tld>
```

z.B.:

```
 192.168.1.12 notebook notebook.meinedomain.local
```

```
192.168.1.13 desktop desktop.meinedomain.local
```

Nun muss dem NFS-Server im Terminal [2] nur angewiesen werden, **/etc/exports** neu einzulesen.

```
sudo exportfs -ra
```

Alternativ kann der gesamte NFS-Server neu gestartet werden:

```
sudo /etc/init.d/nfs-kernel-server restart
```

Die eventuell auftauchende Warnung *"exportfs: No options for..."* kann ignoriert werden.

Die exportierten Freigaben können nun per showmount von einem Client abgefragt werden:

```
showmount -e <nfs-server>
```

<nfs-server> steht hierbei natürlich für den Namen oder Adresse des NFS-Servers

Zugriffskontrolle

Der NFS-Server beachtet die Zugriffsbeschränkungen, die durch die Dateien **/etc/hosts.allow** und **/etc/hosts.deny** beschrieben werden (siehe auch **inetd#tcpwrapper** [https://wiki.ubuntuusers.de/inetd/] und man hosts_access).

Falls man diese Art der Zugriffskontrolle (zusätzlich zu der aus **/etc/exports**) verwenden will, sind folgende Einträge vorzunehmen (für den Fall, dass diese Dateien noch nicht existieren, kann man sie einfach selber anlegen):

In der **/etc/hosts.deny**:

```
portmap: ALL
```

Und in der **/etc/hosts.allow**:

```
# falls nur die IP 192.168.1.13 Zugriff erhalten soll:
portmap: 192.168.1.13

# falls das gesamte LAN Zugriff erhalten soll:
portmap: 192.168.1.
# oder
portmap: 192.168.1.0/24
```

Auf dieselbe Art sollte man dann auch den Zugriff auf den mountd und den statd beschränken. Zu beachten ist, dass für diese Dienste nur IP-Adressen in den hosts_access-Dateien funktionieren, keine Domainnamen.

Hinweis:

Die Einschränkung des statd bietet sich auch auf Client-Rechnern an, insbesondere auf Notebooks, die auch mal in unsicheren Netzen unterwegs sind. Hier muss der Zugriff nur dem/den Server(n) erlaubt werden.

Auf Freigaben zugreifen

Hinweis:

Weil NFS vom **GVFS** nicht unterstützt wird, können die Dateimanager **Nautilus**, **Thunar** und andere nicht direkt auf NFS-Freigaben zugreifen und Netzwerke auch nicht nach NFS-Freigaben durchsuchen. Zum Durchsuchen des Netzwerks eignet sich der Befehl showmount; als Alternative bietet sich auch **Autofs** an.

Damit der Client auf die Freigaben zugreifen kann, muss er sie einfach einbinden können. Hierzu ein Terminal öffnen [2] und

```
cd /media
sudo mkdir MEINEFREIGABE
sudo mount ipadresse:/PFAD/ZUR/FREIGABE /media/MEINEFREIGABE
```

eingeben. Im Falle einer Notebookfreigabe sieht das etwa so aus:

```
cd /media
sudo mkdir server
sudo mount 192.168.1.13:/home /media/server
```

Man könnte nun ein Shellskript schreiben, das bei Aufruf eine Verbindung zum Server herstellt. (Achtung: Wenn das Verzeichnis schon erstellt wurde, muss dieses natürlich nicht mehr erstellt werden.) Die zweite Möglichkeit ist, das Ganze mit Root-Rechten in die **/etc/fstab**-Datei[4] zu schreiben [3].

Beispiel für den Eintrag in die **/etc/fstab**:

```
192.168.6.13:/home /media/server nfs rw 0 0
```

Weitere Optionen in der /etc/fstab	
Option	**Beschreibung**
rw	Lese- und Schreibrechte
ro	Nur Leserechte
hard	Bei Unterbrechungen ohne Timeout warten bis der Server wieder normal erreichbar ist
soft	Bei Unterbrechungen sofort einen Timeout machen (verhindert ein Einfrieren des Dateimanagers)
timeo=<SEKUNDEN>	In Verbindung mit soft kann festgelegt werden, wann der Timeout erfolgen soll.
bg	Bei einem Timeout, wird der mount im Hintergrund weiter versucht. Ist z.B. bei einem Laptop, das im Heimnetz automatisch einen NFS-Server mounten soll, nützlich.
intr	Erlaubt einem wartenden Programm bei Bedarf dennoch zu unterbrechen/killen
nolock	Deaktiviert das Sperren von Dateien. Wird gelegentlich für die Verbindung zu alten NFS-Servern benötigt
rsize=8192,wsize=8192	Mit diesen beiden Optionen kann man die Blockgröße der übertragenen Daten festlegen. In den allermeisten Fällen ist es nicht empfehlenswert, diese Optionen zu setzen. Server und Client handeln diese Werte selbst aus und erreichen so ein Maximum an Performance. **Achtung!** Falsche Werte können die Geschwindigkeit von NFS um bis zu **50%** reduzieren.

Weitere Optionen stehen in der **Manpage** [https://wiki.ubuntuusers.de/man/] zu nfs.

Hinweis:

Portmap öffnet seinen Port standardmäßig an allen Netzwerkschnittstellen, was auf einem Client-Rechner nicht unbedingt erwünscht ist (vor allem bei Laptops, die auch in anderen Netzen unterwegs sind). Man kann das ändern, indem man einfach folgenden Befehl ausführt und die Frage, ob Portmap nur an localhost gebunden werden soll, mit einem *JA* beantwortet. Damit ist der Port von anderen Rechnern nicht mehr erreichbar.

```
sudo dpkg-reconfigure portmap
```

Problembehebung

Sollten beim Zugriff auf NFS-Freigaben Probleme auftreten (z.B. Fehlermeldungen der Art *"Permission denied"*, kein Schreibzugriff, scheinbar leere Ordner oder Ähnliches), so hängt dies sehr häufig mit mangelnden bzw. fehlerhaft vergebenen **Rechten** [https://wiki.ubuntuusers.de/Rechte/] im eingebundenen (entfernten) Dateisystem zusammen.

Weitere Hinweise hierzu finden sich vor allem unter **mount -> Rechte** [https://wiki.ubuntuusers.de/mount/#Rechte] sowie **Externe Laufwerke einhängen** [https://wiki.ubuntuusers.de/mount/#Externe-Laufwerke-einhaengen].

Hinweis:

Dateien und Ordner, die sich auf Partitionen mit dem Dateisystem **NTFS** befinden, können erst mit Kernel-Versionen ab 2.6.27 über NFS freigegeben werden. Dies ist in Ubuntu ab der Version 8.10 (Intrepid Ibex) der Fall. Bei der Freigabe von Dateien auf VFAT-Partitionen (FAT32) über NFS muss mit Problemen gerechnet werden. Als Ausweg empfiehlt sich dann **Samba**.

Sollte die Dateiübertragung langsam sein oder gar abbrechen, so sollte man prüfen, ob in der Datei **/var/log/kernel.log** des Clients Einträge der Form

```
nfs: server <server> not responding, timed out
```

auftauchen. Sollte dies der Fall sein, so könnte es helfen im Mount-Eintrag in **/etc/fstab** die Option hard statt soft zu verwenden.

Probleme mit nfs4, Kerberos und LDAP

Es kommt bei dieser "Unternehmens-Konfiguration" zu Problemen mit **rpcbind / portmap**. Die Laufwerke werden häufig(nur manchmal klappt es) nicht eingehängt (Permission denied). Es ist möglich, das automatische Einhängen abzustellen und erst nach Verfügbarkeit des Netzwerkes die nfs-Laufwerke einzuhängen. Hierzu kann man in der **/etc/fstab** die Einträge für die nfs4-Laufwerke mit dem Parameter **noauto** versehen und mit folgenden Code in der **/etc/rc.local** die Laufwerke einhängen. Der Code muss vor dem "exit" eingebaut und natürlich auf das eigene Netz angepasst werden!

```
echo "Warte aufs Netz ... "
while [ $(ping -w1 -c1 192.168.0.1 | grep -c "0 received") -eq 1 ]; do
    echo "."
done

echo "Netz ist da!"
for i in $(grep noauto /etc/fstab | grep -o "^[^ ]*"); do
    echo $i
    mount $i
done
```

Links

Intern

- **Serverdienste** [https://wiki.ubuntuusers.de/Serverdienste/] – Übersichtsartikel
- **Heimnetzwerk** [https://wiki.ubuntuusers.de/Heimnetzwerk/] – Einführender Artikel; betrifft vor allem einfache Anwendungen
- **Autofs** [https://wiki.ubuntuusers.de/Autofs/] – Erlaubt auch das Browsen und Einbinden von NFS-Freigaben

Extern

- **NFS-Server** [http://de.linwiki.org/wiki/Linuxfibel_-_Netzwerk_Server_-_NFS_Server] – Linuxfibel
- **NFS** [http://www.selflinux.de/selflinux/html/nfs.html] – Ausführliche Erklärung (für fortgeschrittene Benutzer) von Selflinux.de
- **NFSv4Howto** [https://help.ubuntu.com/community/NFSv4Howto] im Ubuntu Help Wiki
- **NFS-HowTo - Troubleshooting** [http://nfs.sourceforge.net/nfs-howto/ar01s07.html] - Gute Hilfestellung, wenn etwas nicht funktionieren sollte!

Diese Revision [https://wiki.ubuntuusers.de/NFS/a/revision/857273/] wurde am 10. Dezember 2015 18:33 von **frustschieber** erstellt.
Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Netzwerk** [https://wiki.ubuntuusers.de/wiki/tags/Netzwerk/], **Server** [https://wiki.ubuntuusers.de/wiki/tags/Server/], **Freigaben** [https://wiki.ubuntuusers.de/wiki/tags/Freigaben/]