

9.3. NFS-Server-Konfiguration

Es gibt drei Arten, einen NFS-Server unter Red Hat Enterprise Linux zu konfigurieren: die Verwendung von **NFS Server Configuration Tool** (`redhat-config-nfs`), die manuelle Veränderung der Konfigurationsdatei (`/etc/exports`), oder die Verwendung des `/usr/sbin/exportfs`-Befehls.

Für Anweisungen bezüglich des **NFS Server Configuration Tool**, siehe Kapitel **Network File System (NFS)** im **Red Hat Enterprise Linux Handbuch zur System-Administration**. Der Rest dieses Abschnittes behandelt die manuelle Bearbeitung von `/etc/exports` und die Verwendung des `/usr/sbin/exportfs` Befehls für den Export von NFS-Dateisystemen.

9.3.1. Die `/etc/exports`-Konfigurationsdatei

Die Datei `/etc/exports` kontrolliert, welche Dateisysteme an Remote-Hosts exportiert werden und legt die Optionen fest. Leere Zeilen werden ignoriert, Kommentare können mithilfe des Rautezeichens `#` am Zeilenanfang eingegeben werden. Lange Zeilen können durch einen inversen Schrägstrich (`\`) umgebrochen werden. Jedes exportierte Dateisystem sollte eine eigene Zeile haben. Listen von autorisierten Hosts, die nach einem exportierten Dateisystem platziert sind, müssen durch Leerzeichen getrennt werden. Die Optionen für alle Hosts müssen in Klammern direkt nach der Hostbezeichnung stehen. Zwischen dem Host und der ersten Klammern darf kein Leerzeichen sein.

Eine Zeile für ein exportiertes Dateisystem hat die folgende Struktur:

```
<export> <host1>(<options>) <hostN>(<options>)...
```

Ersetzen Sie in dieser Struktur `<export>` mit dem Verzeichnis, das exportiert wird. Ersetzen Sie `<host1>` mit dem Host oder Netzwerk, mit dem der Export gemeinsam verwendet wird und ersetzen Sie `<options>` mit den Optionen für diesen Host oder dieses Netzwerk. Zusätzliche Hosts können in einer Liste durch Leerzeichen getrennt festgelegt werden.

Die folgende Methode kann dazu verwendet werden, Hostnamen festzulegen:

- *single host* — Ein bestimmter Host, einschließlich des kompletten Domain-Namens, des Hostnamens oder der IP-Adresse wird festgelegt.
- *wildcards* — Dort, wo ein `*` oder ein `?` verwendet werden, um eine Gruppierung von voll qualifizierten Domainnamen zu berücksichtigen, die mit einer bestimmten Reihe von Buchstaben übereinstimmen. Wildcards dürfen nicht mit IP-Adressen verwendet werden. Es kann aber möglich sein, dass diese zufällig funktionieren, wenn die umgekehrten DNS- Lookups scheitern.

Seien Sie jedoch beim Umgang mit Wildcards im Zusammenhang mit voll qualifizierten Domainnamen vorsichtig, da sie genauer arbeiten können als erwartet. So erlaubt die Verwendung von `*.example.com` als Wildcard zum Beispiel `sales.example.com` den Zugriff auf das exportierte Dateisystem, aber nicht `bob.sales.example.com`. Um beide Möglichkeiten zu erfassen, müssen sowohl `*.example.com` als auch `*.*.example.com` festgelegt werden.

- *IP networks* — Erlaubt die Übereinstimmung von Hosts auf der Basis ihrer IP-Adressen in einem großen Netzwerk. `192.168.0.0/28` lässt zum Beispiel die ersten 16 IP-Adressen von `192.168.0.0` bis `192.168.0.15` zu, um auf das exportierte Dateisystem zuzugreifen, aber nicht `192.168.0.16` und höher.
- *netgroups* — Lässt einen NIS-Netgroup-Namen zu, der wie folgt geschrieben wird: `@<group-name>`. Dadurch übernimmt der NIS-Server die Kontrolle für den Zugriff auf diese exportierten Dateisysteme, und Benutzer können ohne Auswirkung auf `/etc/exports` zu einer NFS- Gruppe hinzugefügt oder aus einer solchen entfernt werden.

`/etc/exports` benötigt in seiner einfachsten Form nur die Festlegung des Verzeichnisses, das exportiert wird und des Hosts, der es verwenden kann - wie in folgendem Beispiel:

```
/exported/directory bob.example.com
```

In diesem Beispiel kann `bob.example.com /exported/directory/` mounten. Weil in diesem Beispiel keine Optionen festgelegt wurden, treten die folgenden NFS-Optionen in Kraft:

- *ro* — Schreibgeschützte Mounts des exportierten Dateisystems. Remote-Hosts können die Daten, die auf diesem

Dateisystem gemeinsam verwendet werden, nicht verändern. Wenn Sie zulassen möchten, dass Hosts in dem Dateisystem Änderungen vornehmen, müssen Sie die Option `rw` verwenden (lesen und schreiben).

- `wdelay` — Weist den NFS-Server an, das Schreiben auf einer Platte zu verzögern, wenn das Aufheben des Schreibschutz-Modus bevorsteht. Dies kann die Leistung verbessern, indem die Anzahl der einzelnen Schreibbefehle für die Platte verringert wird. Mit der Option `no_wdelay` kann diese Funktion deaktiviert werden, die nur funktioniert, wenn Sie die Option `sync` verwenden.
- `root_squash` — Nimmt Root-Benutzern, welche von außen verbunden sind, deren Root-Rechte, indem diese die Benutzer-ID `nfsnobody` erhalten. Auf diese Weise wird die Kontrolle des Remote-Root-Users auf den niedrigsten lokalen Benutzer "gequetscht", was verhindert, dass Dateien auf dem Remote-Server unerlaubterweise verändert werden. Als Alternative können Sie mit der Option `no_root_squash` das "Sqashing" des Roots deaktivieren. Um jeden Remote-Benutzer, einschließlich Root, zu quetschen, verwenden Sie die Option `all_squash`. Um die Benutzer- und Gruppen-IDs festzulegen, die mit Remote-Benutzern eines bestimmten Hosts verwendet werden sollen, benutzen Sie die Optionen `anonuid` und `anongid`. Auf diese Weise können Sie für Remote NFS-Benutzer ein spezielles Benutzer-Konto (`anonuid=<uid-value>`, `anongid=<gid-value>`) erstellen und gemeinsam verwenden, wobei `<uid-value>` die Benutzer-ID-Nummer und `<gid-value>` die Gruppen-ID-Nummer ist.



Wichtig

Standardmäßig werden *access control lists (ACLs)* von NFS unter Red Hat Enterprise Linux unterstützt. Um diese Funktion zu deaktivieren, legen Sie die `no_acl` Option fest, wenn Sie das Dateisystem exportieren. Mehr Informationen zu dieser Funktion finden Sie im Kapitel **Network File System (NFS)** im **Red Hat Enterprise Linux Handbuch zur System-Administration**.

Jede standardmäßige Einstellung für jedes exportierte Dateisystem muss ausdrücklich überschrieben werden. Wenn z.B. die `rw` Option nicht festgelegt ist, dann wird das exportierte Dateisystem schreibgeschützt gemeinsam verwendet. Im Folgenden eine Beispielszeile von `/etc/exports`, die zwei Standardoptionen überschreibt:

```
/another/exported/directory 192.168.0.3(rw, sync)
```

In diesem Beispiel kann `192.168.0.3 /another/exported/directory/` beschreibbar mounten und alle Übertragungen auf die Platte werden auf der Platte gesichert, bevor die Schreibanfrage des Client abgeschlossen wird.

Wo keine Standardwerte angegeben sind, stehen zusätzliche Optionen zur Verfügung. Diese bieten die Möglichkeit, das Überprüfen der Sub-Trees zu deaktivieren, erlauben den Zugriff von unsicheren Ports sowie das Sperren unsicherer Dateien (für bestimmte frühere NFS-Client-Implementierungen notwendig). Auf der `exports-man`-Seite finden Sie weitere Details über diese weniger verwendeten Optionen.



Warnung

Das Format der Datei `/etc/exports` ist sehr präzise gestaltet, besonders im Bezug auf Leerzeichen. Denken Sie daran, exportierte Dateisystem immer getrennt von Hosts aufzuführen und Hosts durch Leerzeichen voneinander trennen. Es sollten jedoch keine weiteren Leerzeichen in der Datei sein, es sei denn, sie werden in Kommentarzeilen verwendet.

So bedeuten zum Beispiel die folgenden beiden Zeilen nicht das gleiche:

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

Die erste Zeile erlaubt nur Benutzern von `bob.example.com` den Zugriff im beschreibbaren Modus auf das Verzeichnis `/home`. Die zweite Zeile erlaubt Benutzern von `bob.example.com`, das Verzeichnis im schreibgeschützten Modus zu mounten (der Standard), wohingegen alle anderen es im Lese-/Schreib-Modus mounten können.

Für genaue Anweisungen zur Konfiguration eines NFS-Servers durch Bearbeitung von `/etc/exports`, siehe Kapitel **Server Security** in der **Red Hat Enterprise Linux Handbuch zur System-Administration**.

9.3.2. Der `exportfs` Befehl

Jedes Dateisystem, das via NFS an Remote-Benutzer exportiert wird sowie die Zugriffsrechte für diese Dateisysteme werden in der Datei `/etc/exports` abgelegt. Wenn der `nfs` Dienst startet, wird diese Datei mit dem `/usr/sbin/exportfs` Befehl geöffnet und gelesen. Dann werden die für Remote-Benutzer verfügbaren Dateien an `rpc.mountd` (falls NFSv2 oder NFSv3) und `rpc.nfsd` übergeben.

Mit der manuellen Ausführung des Befehls `/usr/sbin/exportfs` wird dem Root-Benutzer gestattet, Verzeichnisse selektiv zu exportieren oder nicht zu exportieren, ohne den NFS-Dienst neu starten zu müssen. Wenn `/usr/sbin/exportfs` die korrekten Optionen erhält, wird das exportierte Dateisystem in `/var/lib/nfs/xtab` gespeichert. Da `rpc.mountd` sich für das Festlegen der Privilegien für den Zugriff auf ein Dateisystem auf die Datei `xtab` bezieht, werden Änderungen an der Liste der exportierten Dateisysteme sofort wirksam.

Bei der Verwendung des Befehls `/usr/sbin/exportfs` stehen folgende übliche Optionen zur Verfügung:

- `-r` — Alle in `/etc/exports` aufgelisteten Verzeichnisse werden exportiert und in `/etc/lib/nfs/xtab` wird eine neue Exportliste erstellt. Durch diese Option wird die Exportliste einschließlich aller Änderungen, die in `/etc/exports` vorgenommen wurden, aktualisiert.
- `-a` — Alle Verzeichnisse werden exportiert oder nicht exportiert, je nachdem, welche anderen Optionen in `/usr/sbin/exportfs` gewählt wurden. Wenn keine anderen Optionen festgelegt wurden, exportiert `/usr/sbin/exportfs` alle in `/etc/exports` festgelegten Dateisysteme.
- `-o file-systems` — Ermöglicht dem Benutzer Verzeichnisse zum Exportieren festzulegen, die nicht in `/etc/exports` aufgeführt sind. Ersetzen Sie `file-systems` mit den zusätzlichen Dateisystemen, die exportiert werden sollen. Diese Dateisysteme müssen auf dieselbe Weise formatiert werden, wie sie in `/etc/exports` angegeben sind. Siehe [Abschnitt 9.3.1](#) für weitere Informationen über die `/etc/exports` Syntax. Diese Option wird oft verwendet, um exportierte Dateisysteme zu testen, bevor sie endgültig zu der Liste der zu exportierenden Dateisysteme hinzugefügt werden.
- `-i` — Ignoriert `/etc/exports`; nur die Optionen, die von der Befehlszeile aus eingegeben wurden, werden zum Definieren der exportierten Dateisysteme verwendet.
- `-u` — Hebt den Export aller gemeinsam verwendeter Verzeichnisse auf. Der Befehl `/usr/sbin/exportfs -ua` unterbricht die gemeinsame Verwendung von NFS-Dateien, während alle NFS-Daemone weiter arbeiten. Geben Sie den Befehl `exportfs -r` ein, um die gemeinsame Verwendung von NFS-Dateien wieder fortzusetzen.
- `-v` — Verbose Operation, bei der exportierte oder nicht exportierte Dateisysteme detaillierter angezeigt werden, wenn der Befehl `exportfs` ausgeführt wird.

Wenn für den Befehl `/usr/sbin/exportfs` keine Optionen eingegeben werden, wird eine Liste der aktuell exportierten Dateisysteme angezeigt.

Für weitere Informationen über den `/usr/sbin/exportfs` Befehlsiehe `exportfs man`-Seite.

9.3.2.1. Den Befehl `exportfs` mit NFSv4 verwenden

Da NFSv4 nicht mehr länger das Protokoll `rpc.mountd` wie in NFSv2 und NFSv3 anwendet, hat sich das Mounten von Dateisystemen geändert.

Ein NFSv4-Client hat nunmehr die Fähigkeit alle Exporte des NFSv4-Servers als ein einziges Dateisystem zu sehen, welches NFSv4 Pseudo-Dateisystem genannt wird. Auf Red Hat Enterprise Linux wird das Pseudo-Dateisystem als ein einzelnes, reales Dateisystem beim Export mit der Option `fsid=0` identifiziert.

So können zum Beispiel folgende Befehle auf einem NFSv4-Server ausgeführt werden:

```
mkdir /exports
mkdir /exports/opt
mkdir /exports/etc
mount --bind /usr/local/opt /exports/opt
mount --bind /usr/local/etc /exports/etc
exportfs -o fsid=0,insecure,no_subtree_check gss/krb5p:/exports
exportfs -o rw,nohide,insecure,no_subtree_check gss/krb5p:/exports/opt
exportfs -o rw,nohide,insecure,no_subtree_check gss/krb5p:/exports/etc
```

In diesem Beispiel werden Clients mit mehrfachen zu mountenden Dateisystemen mittels der `--bind`-Option versehen.

[Zurück](#)

Starten und Anhalten von NFS

[Zum Anfang](#)

[Nach oben](#)

[Nach vorne](#)

NFS-Client-Konfigurationsdateien