

# Menyiapkan Mount NFS di Ubuntu 20.04

## [Langkah 1 — Mengunduh dan Menginstal Komponen](#)

Kita akan mulai dengan menginstal komponen yang diperlukan di setiap server.

### [Di Hos](#)

Di server **hos**, instal paket `nfs-kernel-server`, yang akan memungkinkan Anda berbagi direktori. Karena ini adalah operasi pertama yang Anda lakukan dengan `apt` di sesi ini, perbarui indeks paket lokal Anda sebelum instalasi:

```
1. sudo apt update
2.
3. sudo apt install nfs-kernel-server
4.
```

Copy

Setelah paket-paket ini diinstal, beralihlah ke server **klien**.

### [Di Klien](#)

Di server **klien**, kita perlu menginstal paket bernama `nfs-common` yang menyediakan fungsionalitas NFS tanpa menyertakan komponen server apa pun. Sekali lagi, perbarui indeks paket lokal sebelum instalasi untuk memastikan Anda memiliki informasi terbaru:

```
1. sudo apt update
2.
3. sudo apt install nfs-common
4.
```

Copy

Karena kedua server memiliki paket yang diperlukan, kita dapat mulai mengonfigurasinya.

## [Langkah 2 — Membuat Direktori Bersama di Hos](#)

Kita akan berbagi dua direktori terpisah, dengan pengaturan konfigurasi yang berbeda, untuk menggambarkan dua cara utama mengonfigurasi mount NFS berkenaan dengan akses superuser.

Superuser dapat melakukan apa saja di sistem mereka. Walau demikian, direktori yang melakukan mount NFS bukanlah bagian dari sistem tempat direktori itu di-mount, jadi secara alami, server NFS menolak melakukan operasi yang membutuhkan privilese superuser. Pembatasan alami ini berarti superuser di **klien** tidak dapat menulis berkas sebagai **root**, menetapkan ulang kepemilikan, atau melakukan tugas superuser lainnya pada mount NFS.

Walaupun demikian, terkadang ada pengguna tepercaya di sistem **klien** yang perlu melakukan tindakan-tindakan ini di sistem berkas yang melakukan mount tetapi tidak perlu memiliki akses superuser di **hos**. Anda dapat mengonfigurasi server NFS untuk mengizinkan hal ini, walaupun menimbulkan unsur risiko, karena pengguna tersebut `_dapat_` memperoleh akses root ke seluruh sistem **hos**.

### [Contoh 1: Mengekspor Mount Serbaguna](#)

Dalam contoh pertama, kita akan membuat mount NFS serbaguna yang menggunakan perilaku NFS asali untuk mempersulit pengguna dengan privilese root di mesin **klien** dalam berinteraksi dengan **hos** menggunakan privilese superuser **klien** tersebut. Anda dapat menggunakan sesuatu seperti ini untuk menyimpan berkas yang diunggah menggunakan sistem manajemen konten atau membuat ruang bagi pengguna agar mudah berbagi berkas proyek.

Pertama-tama, buat direktori bersama:

```
1. sudo mkdir /var/nfs/general -p
2.
```

Copy

Karena kita membuatnya dengan `sudo`, direktori dimiliki oleh pengguna **root hos**:

```
1. ls -la /var/nfs/general
2.
```

Copy

Output

```
drwxr-xr-x 2 root root 4096 May 14 18:36 .
```

NFS akan menerjemahkan operasi **root** apa pun di **klien** ke kredensial `nobody:nogroup` sebagai tindakan keamanan. Karenanya, kita perlu mengubah kepemilikan direktori agar sesuai dengan kredensial itu.

```
1. sudo chown nobody:nogroup /var/nfs/general
2.
```

Copy

Sekarang Anda siap mengekspor direktori ini.

### [Contoh 2: Mengekspor Direktori Rumah](#)

Dalam contoh kedua, tujuannya adalah membuat direktori rumah pengguna yang disimpan di **hos** tersedia di server **klien**, sekaligus memberikan akses yang dibutuhkan administrator tepercaya dari server **klien** tersebut untuk mengelola pengguna dengan mudah.

Untuk melakukannya, kita akan mengekspor direktori `/home`. Karena sudah ada, kita tidak perlu membuatnya. Kita juga tidak akan mengubah izinnya. Jika *dilakukan*, ini dapat menyebabkan serangkaian masalah bagi siapa saja yang memiliki direktori rumah di mesin **hos** tersebut.

### [Langkah 3 — Mengonfigurasi Ekspor NFS di Server Hos](#)

Selanjutnya, kita akan masuk ke dalam berkas konfigurasi NFS untuk persiapan berbagi sumber daya ini.

Di mesin **hos**, buka berkas `/etc/export` dalam editor teks Anda dengan privilese **root**:

```
1. sudo nano /etc/exports
2.
```

Copy

Berkas itu memiliki komentar yang menunjukkan struktur umum setiap baris konfigurasi. Sintaksnya adalah sebagai berikut:

```
/etc/exports
```

```
directory_to_share client(share_option1,...,share_optionN)
```

Kita perlu membuat sebuah baris untuk setiap direktori yang rencananya akan kita gunakan bersama. Pastikan mengubah tempat teks `client_ip` yang ditunjukkan di sini ke alamat IP Anda yang sebenarnya:

```
/etc/exports
```

```
/var/nfs/general client_ip(rw, sync, no_subtree_check)
```

```
/home client_ip(rw, sync, no_root_squash, no_subtree_check)
```

Di sini, kita menggunakan opsi konfigurasi yang sama untuk kedua direktori dengan pengecualian `no_root_squash`. Mari kita lihat arti setiap opsi ini:

- `rw`: Opsi ini memberi komputer **klien** dengan akses baca dan tulis ke volume.
- `sync`: Opsi ini memaksa NFS menulis perubahan ke disk sebelum menjawab. Ini menghasilkan lingkungan yang lebih stabil dan konsisten karena balasan tersebut mencerminkan kondisi aktual dari volume jauh. Namun, hal ini juga mengurangi kecepatan operasi berkas.
- `no_subtree_check`: Opsi ini mencegah pemeriksaan subtree, yaitu suatu proses ketika **hos** harus memeriksa apakah berkas tersebut benar-benar masih tersedia di tree yang diekspor untuk setiap permintaan. Hal ini dapat menyebabkan banyak masalah bila berkas diubah namanya ketika **klien** telah dibuka. Dalam kebanyakan kasus, lebih baik menonaktifkan pemeriksaan subtree.
- `no_root_squash`: Secara asali, NFS menerjemahkan permintaan pengguna **root** dari jauh menjadi pengguna non-privilese di server. Cara ini dimaksudkan sebagai fitur keamanan untuk mencegah akun **root** di **klien** menggunakan sistem berkas **hos** sebagai **root**. `no_root_squash` menonaktifkan perilaku ini untuk direktori bersama tertentu.

Bila Anda selesai membuat perubahan, simpan dan tutup berkas tersebut. Lalu, untuk membuat direktori bersama dapat digunakan oleh klien yang Anda konfigurasi, mulai ulang server NFS dengan perintah berikut:

```
1. sudo systemctl restart nfs-kernel-server
2.
```

Copy

Namun, sebelum Anda dapat menggunakan direktori bersama yang baru, Anda perlu memastikan lalu lintas ke direktori bersama itu diizinkan oleh aturan firewall.

## [Langkah 4 — Menyesuaikan Firewall di Hos](#)

Pertama-tama, mari kita periksa status firewall untuk melihat jika sudah diaktifkan dan, jika sudah, untuk melihat yang diizinkan saat ini:

```
1. sudo ufw status
2.
```

Copy

Output

```
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere

```
OpenSSH (v6)          ALLOW          Anywhere (v6)
```

Di sistem kita, hanya lalu lintas SSH yang boleh melintas, sehingga kita perlu menambahkan aturan untuk lalu lintas NFS.

Pada banyak aplikasi, Anda dapat menggunakan `sudo ufw app list` dan mengaktifkannya berdasarkan nama, tetapi `nfs` tidak termasuk. Namun, karena `ufw` juga memeriksa `/etc/services` untuk porta dan protokol layanan, kita masih dapat menambah NFS berdasarkan nama. Praktik terbaik menyarankan Anda untuk mengaktifkan aturan paling ketat yang akan tetap mengizinkan lalu lintas yang Anda inginkan, sehingga daripada mengaktifkan lalu lintas dari mana saja, kita akan lebih spesifik.

Gunakan perintah berikut untuk membuka porta `2049` di **hos**, pastikan mengganti alamat IP **klien** Anda:

```
1. sudo ufw allow from client_ip to any port nfs
2.
```

Copy

Anda dapat memverifikasi perubahan dengan mengetik:

```
1. sudo ufw status
2.
```

Copy

Anda akan melihat lalu lintas yang diizinkan dari porta `2049` dalam keluaran:

```
Output
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
2049 ALLOW 203.0.113.24
OpenSSH (v6) ALLOW Anywhere (v6)
```

Ini memastikan bahwa UFW hanya akan mengizinkan lalu lintas NFS di porta `2049` dari mesin **klien** kita.

### [Langkah 5 — Membuat Titik Mount dan Melakukan Mount Direktori di Klien](#)

Karena kini server **hos** telah dikonfigurasi dan menyediakan direktori bersamanya, kita akan menyiapkan **klien** kita.

Untuk membuat direktori jauh bersama di **klien**, kita perlu melakukan mount direktori di **hos** yang ingin dibagikan untuk mengosongkan direktori di **klien**.

**Catatan:** Jika ada berkas dan direktori di titik mount, itu akan disembunyikan begitu Anda melakukan mount direktori bersama NFS. Untuk menghindari kehilangan berkas penting, pastikan direktori itu kosong jika Anda melakukan mount di direktori yang sudah ada.

Kita akan membuat dua direktori untuk melakukan mount:

```
1. sudo mkdir -p /nfs/general
2.
3. sudo mkdir -p /nfs/home
4.
```

Copy

Karena kini kita memiliki lokasi untuk memasukkan direktori jauh bersama dan telah membuka firewall, kita dapat melakukan mount direktori bersama menggunakan alamat IP server **hos** kita:

1. `sudo mount host_ip:/var/nfs/general /nfs/general`
- 2.
3. `sudo mount host_ip:/home /nfs/home`
- 4.

Copy

Perintah ini akan melakukan mount direktori bersama dari komputer hos ke mesin **klien**. Anda dapat memeriksa lagi apakah direktori berhasil melakukan mount dengan beberapa cara. Anda dapat memeriksanya dengan perintah `mount` atau `findmnt`, tetapi `df -h` memberikan keluaran yang lebih mudah dibaca:

1. `df -h`
- 2.

Copy

Output

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	474M	0	474M	0%	/dev
tmpfs	99M	936K	98M	1%	/run
/dev/vda1	25G	1.8G	23G	8%	/
tmpfs	491M	0	491M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	491M	0	491M	0%	/sys/fs/cgroup
/dev/vda15	105M	3.9M	101M	4%	/boot/efi
tmpfs	99M	0	99M	0%	/run/user/1000
10.132.212.247:/var/nfs/general	25G	1.8G	23G	8%	/nfs/general
10.132.212.247:/home	25G	1.8G	23G	8%	/nfs/home

Kedua direktori bersama yang kita mount akan muncul di bagian bawah. Karena dilakukan mount dari sistem berkas yang sama, direktori itu akan menunjukkan penggunaan diska yang sama. Untuk melihat seberapa banyak ruang yang sesungguhnya digunakan di setiap titik mount, gunakan perintah penggunaan diska `du` dan jalur mount. Bendera `-s` memberikan ringkasan penggunaan, bukan menampilkan penggunaan untuk setiap berkas. `-h` mencetak keluaran yang mudah dipahami orang.

Misalnya:

1. `du -sh /nfs/home`
- 2.

Copy

Output

```
36K    /nfs/home
```

Ini menunjukkan kepada kita seluruh isi direktori rumah hanya menggunakan 36K dari ruang yang tersedia.

## [Langkah 6 — Menguji Akses NFS](#)

Selanjutnya, mari kita uji akses ke direktori bersama dengan menulis sesuatu untuk masing-masing direktori bersama.

### Contoh 1: Direktori Bersama Serbaguna

Pertama-tama, tulis berkas percobaan ke direktori bersama `/var/nfs/general:`

1. `sudo touch /nfs/general/general.test`
- 2.

Copy

Lalu, periksa kepemilikannya:

1. `ls -l /nfs/general/general.test`
- 2.

Copy

Output

```
-rw-r--r-- 1 nobody nogroup 0 Aug  1 13:31 /nfs/general/general.test
```

Karena kita telah melakukan mount volume ini tanpa mengubah perilaku asli NFS dan membuat berkas sebagai pengguna **root** mesin **klien** melalui perintah `sudo`, kepemilikan berkas asli menjadi `nobody:nogroup`. Superuser **klien** tidak akan dapat melakukan tindakan administratif umum, seperti mengubah pemilik berkas atau membuat direktori baru untuk grup pengguna, di direktori bersama yang melakukan mount sebagai NFS.

### Contoh 2: Berbagi Direktori Rumah

Untuk membandingkan izin direktori bersama Serbaguna dengan Direktori Rumah, buat berkas di `/nfs/home` dengan cara yang sama:

1. `sudo touch /nfs/home/home.test`
- 2.

Copy

Lalu, lihat kepemilikan berkas tersebut:

1. `ls -l /nfs/home/home.test`
- 2.

Copy

Output

```
-rw-r--r-- 1 root root 0 Aug  1 13:32 /nfs/home/home.test
```

Kita telah membuat `home.test` sebagai **root** menggunakan perintah `sudo`, persis sama caranya dengan membuat berkas `general.test`. Namun, dalam hal ini, berkas ini dimiliki oleh **root** karena kita mengesampingkan perilaku asli saat menetapkan opsi `no_root_squash` di mount ini. Hal ini memungkinkan pengguna **root** kita di mesin **klien** untuk bertindak sebagai **root** dan membuat administrasi akun pengguna menjadi jauh lebih mudah. Pada saat bersamaan, ini berarti kita tidak perlu memberi akses root kepada pengguna ini di **hos**.

### Langkah 7 — Melakukan Mount NFS di Direktori Jauh saat Boot

Kita dapat melakukan mount NFS di direktori bersama jauh secara otomatis saat boot dengan menambahkannya ke berkas `/etc/fstab` pada **klien**.

Buka berkas ini dengan privilese root di editor teks Anda:

1. `sudo nano /etc/fstab`
- 2.

Copy

Di bagian bawah berkas, tambahkan baris untuk masing-masing direktori bersama kita. Hasilnya akan seperti ini:

```
                                /etc/fstab
. . .
host_ip:/var/nfs/general    /nfs/general    nfs
auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
host_ip:/home              /nfs/home       nfs
auto,nofail,noatime,nolock,intr,tcp,actimeo=1800 0 0
```

**Catatan:** Anda dapat menemukan informasi lebih lanjut tentang opsi yang kita tetapkan di sini pada laman utama NFS. Anda dapat mengaksesnya dengan menjalankan perintah berikut:

1. `man nfs`
- 2.

Copy

**Klien** akan secara otomatis melakukan mount partisi jauh saat boot, meskipun mungkin perlu beberapa waktu untuk membuat koneksi dan agar direktori bersama tersedia.

### [Langkah 8 — Melakukan Unmount NFS di Direktori Bersama Jauh](#)

Jika tidak ingin direktori jauh melakukan mount di sistem, Anda dapat melakukan unmount dengan mengeluarkan struktur direktori bersama dan melakukan unmount, seperti ini:

1. `cd ~`
- 2.
3. `sudo umount /nfs/home`
- 4.
5. `sudo umount /nfs/general`
- 6.

Copy

Perhatikan bahwa perintah bernama `umount` tidak melakukan `unmount` seperti yang Anda harapkan.

Perintah ini akan menghapus direktori bersama jauh, dengan membiarkan penyimpanan lokal saja yang dapat diakses:

1. `df -h`
- 2.

Copy

Output

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	474M	0	474M	0%	/dev
tmpfs	99M	936K	98M	1%	/run
/dev/vda1	25G	1.8G	23G	8%	/
tmpfs	491M	0	491M	0%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	491M	0	491M	0%	/sys/fs/cgroup
/dev/vda15	105M	3.9M	101M	4%	/boot/efi
tmpfs	99M	0	99M	0%	/run/user/1000

Jika Anda juga ingin mencegah dilakukan mount kembali saat boot ulang berikutnya, edit `/etc/fstab` dan hapus baris tersebut atau jadikan komentar dengan menempatkan karakter `#` di awal baris. Anda juga dapat mencegah mount otomatis dengan menghapus opsi `auto`, yang akan memungkinkan Anda tetap dapat melakukan mount secara manual.

### Kesimpulan

Dalam tutorial ini, kita telah membuat hos NFS dan menggambarkan beberapa perilaku utama NFS dengan membuat dua mount NFS berbeda, yang kita gunakan bersama dengan klien NFS.

Jika Anda ingin menerapkan NFS dalam produksi, perlu diingat bahwa protokolnya tidak dienkripsi. Jika Anda berbagi melalui jaringan privat, mungkin tidak menjadi masalah. Dalam kasus lain, akan diperlukan VPN atau beberapa tipe terowongan terenkripsi untuk melindungi data Anda.