Network File System

Linux System Administration
Fall 2026

NFS

- Disk access across the network
- Designed by Sun in 1984
  - Storage for diskless workstations
- De facto standard for Unix/Linux
  - NFSv2 1989 - based on UDP
  - NFSv3 1995 - use TCP for congestion
  - NFSv4 2003-2015 - performance and security

NFS Alternatives

- SMB/CIFS (using Samba on Linux)
  - Windows networking file system
- GlusterFS
  - High performance cluster distributed filesystem
  - Load balancing, failover, snapshots, etc.
- Many others

NFS Server

- Install and start
  - `dnf install nfs-utils`
- Set directories to export in `/etc/exports`
  - `/dir subnet(options)`
- Enable and Start server
  - `systemctl enable nfs-server`
  - `systemctl restart nfs-server`

Managing exports

- Export `/var/ftp` to `100.64.42.0/24`

  ```
  /var/ftp 100.64.42.0/24(rw,sync,no_root_squash)
  ```

- Can list several `subnet(option)` space separated
- Set options by subnet
- Refresh with command `exportfs -r`
- Can also add exports to `/etc/exports.d`
- NFS4 should export trees, but mostly does not

export options

- `ro` read only
- `rw` read/write
- `root_squash` UID0=anon
- `no_root_squash`
- `all_squash` UID=anon
- `no_all_squash`
- `anonuid=`
- `anongid=`
- `sync` wait for disk
- `async`
- `secure_locks`
- `insecure_locs`

Testing nfs

- `nfsstat` Shows server and client usage
  - Call stats and types
- `showmount -e` Shows directories exported
  - Run on B as `showmount -e 10.21.32.2`
- `mount` Just try it
  - `root_squash` will bite you

NFS user mapping

- UID or username?
  GID or group?
  - NFS 2&3 used only UID & GID
  - NFS4 uses identity mapping (username and group)
- Identity mapping used in translating uid/gid to names but not for authentication (file access)
  - The **server** controls permissions
- Sanity requires consistent username/UID and group/ GID across systems
- Linux defaults to domain name (*so you can ignore this in Lab 11*)

NFS client

- Install with `apt install nfs-common`
- Mount with `mount host:/dir /localdir`
- Mounts are just like a physical disk or LV, but mount options are somewhat different
- **If the server dies, the client will hang**
  - Use `autofs` to mount as needed

nfs mount options

- `rw` read/write
- `ro` read only
- `bg` try in background
- `hard` block on fail
- `soft` error on fail
- `intr` interruptable
- `nointr` not interruptable
- `retrans` tries before fail
- `timeo` timeout
- `rsize` read buffer size
- `wsize` write buffer size
- `sec` security flavor
- `proto` UDP or TCP
- `nfsvers` NFS version

automount

- Mount the directory tree when you access it
  - Slight delay on first access, unmount on idle
  - Minimizes client hang when server dies
- `autofs` is the kernel implementation
  - `/etc/auto.master` is the master map
  - `/etc/auto.*` are other maps

automount maps

- `master` defines direct and indirect maps
- `direct` map mounts ad hoc directories
  - Example: `/var/mail`
- `indirect` maps mount to a shared directory
  - Example: `/raid/*` and `/net -hosts`
  - NIS maps

Lab 11 hints

- KISS - nothing fancy is required
- Machine E
  - Create directory to export
  - Install NFS, set the export, enable the server
- Machine C (and D)
  - Create mount point (only on C)
  - Figure out access
  - Install NFS, mount the directory, make mount permanent
  - Extra credit: set up `autofs`