University Göttingen
Department of Computer Science
Patrick Höhn

Exercise 3 / October 17, 2023
HPC System Administration / WiSe 2023/24
35 Minutes Total

Installing a NFS Server client infrastructure

NFS (either Network File System or Network File Service) is the most common protocol for sharing files between Unix systems over a network. NFS servers export directories from their local hard disks to NFS clients, which mount them so that they can be accessed like any other directory. The infrastructure consists of a server and a client. The server hosts the data and the client accesses it over the network. For this exercise we will set up a NFS server and a client.

Contents

Task 1: Install NFS Server (10 min) 1

Task 2: configure etc/exports (15 min) 2

Task 3: Install NFS client (10 min) 3

For the block course, you will use course accounts with access to GWDG cloud resources such that you can roll out your own VMs and follow along with the hands-on exercises. Follow the instructions below to prepare your frontend VM. During the course you will deploy additional worker VMs.

Task 1: Install NFS Server (10 min)

1. Connect to the server-manager
```
ssh -i hpcsa-course-vm-key.pem -o ServerAliveInterval=60 cloud@YOUR IP
```
2. Installation of NFS server on CentOS/Rocky/Fedora
```
yum -y install nfs-utils
```
3. Create Root NFS Directory
```
sudo mkdir /nfs
```
4. Check your user and its group
   a) user: `whoami`
   b) group: `groups $(whoami)`
5. Set the permissions for your user and his/her group
```
sudo chown $(whoami):$(id -gn $(whoami)) /nfs
```

Task 2: configure etc/exports (15 min)

Configuration

1. Access to a single client
   `/nfs {clientIP}(rw,sync,no_subtree_check)`

2. Access to several clients
   `/nfs {clientIP}(rw,sync,no_subtree_check)`
   `{clientIP-2}(...)`
   `{clientIP-3}(...)`

3. Access to an entire subnet
   `/nfs {subnetIP}/{subnetMask}(rw,sync,no_subtree_check)`

4. With exportfs we can now make the shared directory available
   `sudo exportfs -a #making the file share available`
   `sudo systemctl restart nfs-server #restarting the NFS server`

   Listing 1: Commented /etc/exports example file

```
# The /usr/games directory can be mounted by the systems named Alice, Bob, and
# Claire. They can read data and run programs from the directory, but they
# cannot write in the directory.
/usr/games -ro,access=Alice:Bob:Claire
# The /home directory can be mounted by the system Bob. Bob has root access
# allowed for the directory.
/home -root=Bob,access=Bob
# Any client can mount the /var/tmp directory. (Notice the absence of an access list.)
/var/tmp
# Only clients designated the \textit{clients} netgroup are allowed to mount the
# /usr/lib directory.
/usr/lib -access=clients
# Access to the directory /accounts/database is only allowed only to clients in
# the accmachines netgroup using NFS version 4 protocol using Kerberos 5
# authentication. Root access is allowed only from accmachine1.
/accounts/database -vers=4,sec=krb5,access=accmachines,root=accmachine1
# Export the /tmp folder using NFS Version 3 to be only mounted read-only
/tmp -vers=3,ro
```

5. If you run a firewall you have to allow access
   `sudo ufw allow`

Task 3: Install NFS client (10 min)

1. Installation of the NFS client on CentOS/Rocky/Fedora
   `sudo yum install nfs-utils`

2. We create a local directory as a mount point for the NFS share
   `sudo mkdir /nfs`

3. Now we mount the NFS share by running the mount command

   - `sudo mount -t nfs {ip of NFS server}:{path on server} /nfs`

   - `sudo mount -t nfs 10.254.1.234:/nfs /nfs`

   The mount point now acts as the root of the nfs-share, displaying all subfolders contained in the folder on the server

4. To verify the correct mounting of the NFS share run either

   - `mount`

   - `df -h`

5. Remote NFS directories can be mounted on startup automatically. They need to be defined in the file /etc/fstab.
   Take a look at the file on your system, if it exists. Listing 2 below gives you an idea how it may look like.

Listing 2: /etc/fstab

```
# /etc/fstab: static file system information.
#
# Lines that start with # are comments
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/sda1 /media/sda1 ntfs defaults,nls=utf8,umask=007,gid=46 0 0
/dev/sda2 /media/sda2 ntfs defaults,nls=utf8,umask=007,gid=46 0 0
# /dev/sda6
UUID=03b77228-ed4c-4218-910e-11b9f77c4b46 / ext4 defaults 0 1
# /dev/sda7
UUID=8883dbc8-80f8-49b8-8c5f-13a32baefe98 none swap sw 0 0
/dev/hda /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/cdrom /media/cdrom0 udf,iso9660 user,noauto 0 0
/dev/ /media/floppy0 auto rw,user,noauto 0 0

# external ntfs-harddrive for data exchange; not integrated by UUID
/dev/sdb1 /media/ntfs-usbdisk ntfs rw,user,noauto,uid=0,gid=46,umask=007,nls=utf8 0 0

# to be filled out:
/dev/sda5 /media/data vfat rw,auto,user,umask=0000 0 0

# set memory for QEMU to a max of 400 MB RAM
none /dev/shm tmpfs defaults,size=400M

# Prepare the inclusion of a filesystem from another pc with fuse/ssh (later "mount
    ↪ /lokaler/mountpoint" suffices
sshfs#username@rechnername:/path/on_remote_pc
/lokal/mountpoint fuse uid=1000,gid=100,umask=0,allow_other,defaults,noauto 0 0

# At the end of the fstab file a blank line has to be placed, otherwise you would get
    ↪ the error: no final newline at the end of /etc/fstab
```

6. As before we create a local directory as a mount point for the NFS share

```
sudo mkdir /nfs
```

7. We edit the /etc/fstab, adding a line for every file share we want to include, similar to /etc/export when installing the server
   ```
   nano /etc/fstab
   ```
   ```
   vim /etc/fstab
   ```
   We add a line defining the NFS share, dividing the parameter with tabulator. It should be one line with no line breaks.

8. The last three parameters are NFS options, which we leave on default
   ```
   {IP of NFS server}:{folder path on server} /nfs nfs defaults 0 0
   ```

9. Examples of NFS options would be
   - The last three parameters are NFS options, which we leave on default
     - `{ip of NFS server}:{path on server} /nfs nfs defaults 0 0`
   - examples of NFS options would be
     - `timeo=n`
       - How long the client waits for a response before it retries an NFS request. Default is 60 seconds (600 * 1/10 second)
     - `rsize=n`
       - The maximum number of bytes in each network READ request. The largest read payload supported by the Linux NFS client is 1,048,576 bytes (one megabyte). The rsize value is a multiple of 1024 with a minimum of 4096 and maximum of 1048576 rounded down to the nearest multiple of 1024.

10. Finally we will mount the share

    a) `mount /nfs`

    b) `mount {IP of NFS server}:{folder path on server}`