# ANNEX A · PRACTICAL 6.3 — NFS

**Name:** _____ **Date:** _____ **Group:** DAM-1

Server: `ubu-srv` (use `ssh srv`). Client: `deb-lab`. Snapshot `pre-nfs` of **both** first.

## TASK 1 Export a directory (server)

```
sudo apt install nfs-kernel-server
sudo mkdir -p /srv/nfs/projects
echo "/srv/nfs/projects 192.168.122.0/24(rw,sync,no_subtree_check)" \
  | sudo tee -a /etc/exports
sudo exportfs -ra
sudo exportfs -v
```

Two options that you did not write but `exportfs -v` shows: _____

☐ **DONE WHEN** `exportfs -v` shows `/srv/nfs/projects` with `rw`.

## TASK 2 Mount it (client)

```
sudo apt install nfs-common
showmount -e ubu-srv
sudo mkdir -p /mnt/projects
sudo mount ubu-srv:/srv/nfs/projects /mnt/projects
df -hT /mnt/projects
touch /mnt/projects/test.txt
sudo touch /mnt/projects/test.txt
```

Type shown by `df -hT`: _____ Result of the two `touch`: _____

The export says `rw`. Why can you not write? Your guess: _____

☐ **DONE WHEN** `df -hT` shows the server's directory on `/mnt/projects`.

## TASK 3 Numbers, not names

Client: `id -u` → _____ Server: `ls -ldn /srv/nfs/projects` → owner number: _____

Server: `sudo chown 1000 /srv/nfs/projects` (use your number from the client). Client:

```
touch /mnt/projects/mine.txt
ls -ln /mnt/projects
sudo touch /mnt/projects/root.txt
```

Server: `ls -l /srv/nfs/projects` → owner **name** of `mine.txt`: _____ Did `sudo touch` work? _____

Why? _____

☐ **DONE WHEN** `mine.txt` exists and you can explain what happened to root.

## TASK 4 Who does the server think you are? (start on the server)

```
id -u joan
sudo mkdir -m 700 /srv/nfs/projects/joan-only
sudo chown joan /srv/nfs/projects/joan-only
```

Joan's number: _____ Client: `getent passwd NUMBER`. If it prints a user, use that name in place of `visitor` and skip the first line:

```
sudo useradd -u NUMBER visitor
sudo -u visitor touch /mnt/projects/joan-only/hello.txt
```

Server: `sudo ls -l /srv/nfs/projects/joan-only` → owner of `hello.txt`: _____

Joan never did this. What did the server believe?

_____

☐ **DONE WHEN** the file is there, owned by `joan`.

## TASK 5 Mount at boot (client)

Add this line at the end of `/etc/fstab` (`sudo nano /etc/fstab`), all on one line:

```
ubu-srv:/srv/nfs/projects  /mnt/projects  nfs  defaults,_netdev,nofail  0  0
```

```
sudo umount /mnt/projects
sudo systemctl daemon-reload
sudo mount -a
findmnt /mnt/projects
```

If `mount -a` shows no error: reboot `deb-lab` and run `findmnt /mnt/projects` again.

☐ **DONE WHEN** after the reboot `findmnt` still shows `ubu-srv:/srv/nfs/projects`.

## TASK 6 A read-only export and the space trap

**6.1** Server: `sudo mkdir -p /srv/nfs/docs`, add this line to `/etc/exports` and run `sudo exportfs -ra`:

```
/srv/nfs/docs 192.168.122.0/24(ro,sync,no_subtree_check)
```

**6.2** Client:

```
sudo mkdir -p /mnt/docs && sudo mount ubu-srv:/srv/nfs/docs /mnt/docs
touch /mnt/docs/x
```

Message: _____

**6.3** Server: put **one space** before `(ro` in that line. `sudo exportfs -ra`, then `sudo exportfs -v`. Who can mount `/srv/nfs/docs` now? _____ Remove the space and run `sudo exportfs -ra` again.

☐ **DONE WHEN** `exportfs -ra` prints no warning and `exportfs -v` has no `<world>` line.

## TASK 7 SMB or NFS?

| | SMB (Samba) | NFS |
|---|---|---|
| Port | | |
| Who is identified | | |
| Typical clients | | |

**Hand in** on Classroom: `practical-6.3-surname.md` (commands, outputs, answers). **Marking (10):** export 1.5 · mount 1.5 · refusal of task 2 explained 1 · UID experiment explained 2 · `fstab` works after reboot 1.5 · read-only export and space trap 1.5 · comparison 0.5 · clear report 0.5.

---

**Exit ticket** — alone, no notes. One answer each.

**1.** How does an NFS server (default set-up) know which user is writing?

☐ **a)** By the user's password
☐ **b)** By the UID number the client sends
☐ **c)** By the user name
☐ **d)** By an SSH key

**2.** What does root squash do?

☐ **a)** Deletes root's files
☐ **b)** Asks root for a password
☐ **c)** Blocks the client machine
☐ **d)** Treats the client's root as `nobody`

**3.** `/srv/docs 10.0.0.0/24 (rw)` — with that space, who can write?

☐ **a)** Every machine that reaches the server
☐ **b)** Only `10.0.0.0/24`
☐ **c)** Nobody
☐ **d)** Only root

**4.** What is `nofail` for in an `fstab` line of a network mount?

☐ **a)** It makes the mount faster
☐ **b)** It mounts read-only
☐ **c)** The client still boots if the server is off
☐ **d)** It hides errors from the log