# NFS export

Mount it to freshly created directory. Add to fstab for automatic mount during boot process.

# Preparations:

```
sudo su
dnf install \
    nfs-utils
```

# Configure

```
# vi /etc/idmapd.conf
```

```
# [General]
# Verbosity = 0
# Domain = (domain)
```

```
# systemctl daemon-reload
# systemctl restart rpcbind
```

# Allow network connectivity:

For Redhat family:

```
firewall-cmd --list-all
firewall-cmd --permanent --add-service=nfs
firewall-cmd --reload
firewall-cmd --list-all
```

For Debian family:

```
ufw status
ufw allow nfs
ufw enable
```

# Check connecivity

Decide which declaration will be used: FQDN or IP address (depending on situation and purpose).

```
export src_host="(source host)"
ping ${src_host}
showmount -e ${src_host}
```

Verify that needed export is listed in the output and set a variable:

```
export src_export="(name of exported directory)"
```

When we are mounting shared network drive, concept "one-to-many". The shared storage must be identified in mount table by source host and exported path in the hierarchial order. This will facilitate of adding additional mounts from the same host. Preparations are as follows:

```
export dir="/mnt/${src_host}/${src_export}/"
echo ${dir}
read -p "Correct?"
mkdir -p ${dir}
ls -la ${dir}
```

Skip this, if you created directory using previous (one-to-many) concept. Create local directory where a NFS export will be mounted. Naming used here comes from the name of attached disk to the VM (usually). To clarify, that it is a data disk, `-data` suffix is added. This is concept "one-to-one", which mean that the disk will be mounted only on single host.

```
export host="$(hostname -s)"
export dir="/mnt/${host}-data/"
echo ${dir}
mkdir -p ${dir}
```

If/when there is a need to use mounted resource on different machines, consider creating symbolic links. This will be useful while configuring the applications' configurations. In my case, I shall mount NFS export for Nextcloud application (ncp1 = NextCloud Production, 1st environment), change it to whatever you need. Do not worry about double forward slashes, shell interpreter will ignore them

and shrink to a single slash.

```
export symlink="/mnt/ncp1"
ln -s ${dir} ${symlink}
ls -la ${symlink}
```

should end up like this:

```
lrwxrwxrwx. 1 root root 21 Apr 29 13:37 /mnt/ncp1 -> /mnt/(src_host)/(src_export)
```

and double-check mountpoint and free disk space

```
df -h ${symlink}
```

As we can see, we are still 'on local drive', because nothing is mounted yet. Before mounting the export, check where target directory is mounted, it should be mounted on the root `/` and it should be empty.

```
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/ol-root   99G  8.2G   91G   9% /
```

```
ls -la ${dir}
```

```
total 0
drwxr-xr-x. 2 root root  6 Apr 29 13:56 .
drwxr-xr-x. 3 root root 18 Apr 29 13:56 ..
```

Finally, check variables and mount the export.

```
echo ${src_host}
echo ${src_export}
echo ${dir}
read -p "Sure to process?"
mount -t nfs4 -o nfsvers=4 ${src_host}:${src_export} ${dir}
mount | grep nfs
```

Check again

```
df -h ${dir}
ls -la ${dir}
```

Mounting should be presented as IP address. Exported path should be mounted in the correct destination. For example:

```
df -h ${dir}
```

```
Filesystem                  Size  Used Avail Use% Mounted on
Filesystem                     Size  Used Avail Use% Mounted on
10.x.x.x:/ifs/data/ARCHAZ/NCFS  1.0T     0  1.0T   0% /mnt/10.x.x.x/(src_export)
```

Let's add to new mounting point to `fstab` to automatically mount on the system boot:

```
cat /etc/fstab
```

```
echo "${src_host}:${src_export} ${dir} nfs4 nfsvers=4,defaults,_netdev,rw,sec=sys 0 0" | tee -a /etc/fstab
tail -n5 /etc/fstab
systemctl daemon-reload
```

Umount and check automatic mounts, simulating restart.

```
umount ${dir}
mount | grep nfs
read -p "Umounted?"
mount -aF
mount | grep nfs
df -h ${dir}
```

# Content

Check for permissions to the destination directory

```
namei -mo ${dir}
```

Try to write to the destination

```
touch ${dir}/test.md
ls -latr ${dir}
```

When possible, test VM restart to ensure disk will be properly mounted, simulating real restart.

```
uptime
shutdown -r now

uptime
mount | grep nfs
```

To troubleshhot, enable RPC NFS logging. Tail the log:

```
rpcdebug -m nfsd -s a
tail -f /var/log/messages
```

To rtroubleshoot NFS, increase verbosity:

```
cat /proc/sys/sunrpc/nfsd_debug
cat /proc/sys/sunrpc/nfs_debug

echo 10 > /proc/sys/sunrpc/nfs_debug
```

Disable logging

```
rpcdebug - nfsd -c
```

In case, RPC bind is not possible, let configure operating system not to check and match user accounts bindinds. Please, acknowledge, that this will effect permissions changes.

```
# for Redhat Linux
sysctl -w nfs.nfs4_disable_idmapping=1

# for Oracle Linux
cat /sys/module/nfs/parameters/nfs4_disable_idmapping
echo "Y" > /sys/module/nfs/parameters/nfs4_disable_idmapping
nfsidmap -c
```

For permanent

```
# for Redhat Linux
echo "nfs.nfs4_disable_idmapping=1" | tee -a /etc/sysctl.d/99-nfs-disable-idmapping.conf
sysctl --system

# for Oracle
echo "options nfs nfs4_disable_idmapping=1" | tee -a /etc/modprobe.d/nfs.conf
```

```
dracut -f
shutdown -r now
```

Check permissions are not assigned to `nobody:nobody` anymore

```
namei -mo /mnt/ncp/
```

---

Revision #10
Created 16 March 2025 11:29:15 by Anton
Updated 1 May 2025 11:04:17 by Anton