Disks and partitions

- LVM - Creation of first physical volume, volume group and logical group.
- LVM - extend the volume
- NFS export
- Moving most consuming directory to the separate disk

LVM - Creation of first physical volume, volume group and logical group.

Author does not take any responsibilities on the guide below.

Create new physical volume, volume group and logical group.

Mount it to freshly created directory.

Add to fstab for automatic mount during boot process.

```
apt install \
    lvm2

dnf install
    lvm2
```

Define variable

```
hostname
export host="$(hostname)"

lsblk

export dev="/dev/sdc"
export size="9.9G"
```

Understand which is filesystem type mainly used on the system

```
df -hT
```

```
[13:12:26 Sun Mar 15] @gcp1mx1 ~# df -hT
Filesystem     Type      Size  Used Avail Use% Mounted on
udev           devtmpfs  977M     0  977M   0% /dev
tmpfs          tmpfs     198M  680K  197M   1% /run
/dev/sda1      ext4       99G   63G   32G  67% /
tmpfs          tmpfs     989M  2.3M  986M   1% /dev/shm
tmpfs          tmpfs     5.0M     0  5.0M   0% /run/lock
/dev/sda15     vfat      124M   12M  113M  10% /boot/efi
tmpfs          tmpfs     198M     0  198M   0% /run/user/1002
```

In my case it is 'ext4', I shall continue in similarity to this. Redhat family distros might want to use 'xfs'

```
export fs="ext4"
# export fs="xfs"

pvcreate ${dev}

vgcreate vg-${host}-data ${dev}

lvcreate -L ${size} -n lv-${host}-data vg-${host}-data

lvscan
```

```
[13:15:59 Sun Mar 15] @gcp1mx1 ~# export fs="ext4"
[13:15:59 Sun Mar 15] @gcp1mx1 ~# pvcreate ${dev}
  Physical volume "/dev/sdc" successfully created.
[13:16:04 Sun Mar 15] @gcp1mx1 ~# vgcreate vg-${host}-data ${dev}
  Volume group "vg-gcp1mx1-data" successfully created
[13:16:11 Sun Mar 15] @gcp1mx1 ~# lvcreate -L ${size} -n lv-${host}-data vg-${host}-data
  Rounding up size to full physical extent 9.90 GiB
  Logical volume "lv-gcp1mx1-data" created.
[13:16:17 Sun Mar 15] @gcp1mx1 ~# lvscan
  ACTIVE            '/dev/vg-gcp1mx1-data/lv-gcp1mx1-data' [9.90 GiB] inherit
[13:16:22 Sun Mar 15] @gcp1mx1 ~#
```

```
mkfs.${fs} /dev/vg-${host}-data/lv-${host}-data
mkdir -p /mnt/${host}-data
mount /dev/vg-${host}-data/lv-${host}-data /mnt/${host}-data
df -h
```

```
[13:16:48 Sun Mar 15] @gcp1mx1 ~# mkfs.${fs} /dev/vg-${host}-data/lv-${host}-data
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 2595840 4k blocks and 648960 inodes
Filesystem UUID: c450ff0f-f073-4faa-8e56-cce7195f0514
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

[13:16:48 Sun Mar 15] @gcp1mx1 ~# mkdir -p /mnt/${host}-data
[13:16:54 Sun Mar 15] @gcp1mx1 ~# mount /dev/vg-${host}-data/lv-${host}-data /mnt/${host}-data
[13:16:59 Sun Mar 15] @gcp1mx1 ~# df -h
Filesystem                                      Size  Used Avail Use% Mounted on
udev                                            977M     0  977M   0% /dev
tmpfs                                           198M  684K  197M   1% /run
/dev/sda1                                        99G   63G   32G  67% /
tmpfs                                           989M  2.3M  986M   1% /dev/shm
tmpfs                                           5.0M     0  5.0M   0% /run/lock
/dev/sda15                                      124M   12M  113M  10% /boot/efi
tmpfs                                           198M     0  198M   0% /run/user/1002
/dev/mapper/vg--gcp1mx1--data-lv--gcp1mx1--data  9.7G   24K  9.2G   1% /mnt/gcp1mx1-data
```

```
#TODO: add into /fstab with oneliner
echo "add me to fstab"
mount | grep ${host}
nano /etc/fstab
```

```
/dev/mapper/vg--host--data-lv--host--data /mnt/host-data ext4 defaults 0 1
```

```
  GNU nano 7.2                                                    /etc/fstab *
# /etc/fstab: static file system information
UUID=[redacted] / ext4 rw,discard,errors=remount-ro,x-systemd.growfs 0 1
UUID=[redacted] /boot/efi vfat defaults 0 0
/dev/mapper/vg--gcp1mx1--data-lv--gcp1mx1--data /mnt/gcp1mx1-data ext4 defaults 0 1
```

Reload fstab and remount all drives. New drive must be mounted.

```
systemctl daemon-reload
mount -a
```

```
[13:19:18 Sun Mar 15] @gcp1mx1 ~# nano /etc/fstab
[13:20:05 Sun Mar 15] @gcp1mx1 ~# systemctl daemon-reload
[13:20:10 Sun Mar 15] @gcp1mx1 ~# mount -a
[13:20:12 Sun Mar 15] @gcp1mx1 ~# df -h
Filesystem                                        Size  Used Avail Use% Mounted on
udev                                              977M     0  977M   0% /dev
tmpfs                                             198M  688K  197M   1% /run
/dev/sda1                                          99G   63G   32G  67% /
tmpfs                                             989M  2.3M  986M   1% /dev/shm
tmpfs                                             5.0M     0  5.0M   0% /run/lock
/dev/sda15                                        124M   12M  113M  10% /boot/efi
tmpfs                                             198M     0  198M   0% /run/user/1002
/dev/mapper/vg--gcp1mx1--data-lv--gcp1mx1--data  9.7G   24K  9.2G   1% /mnt/gcp1mx1-data
[13:20:40 Sun Mar 15] @gcp1mx1 ~#
```

When possible, test VM restart to ensure disk will be mounted

```
shutdown -r now
```

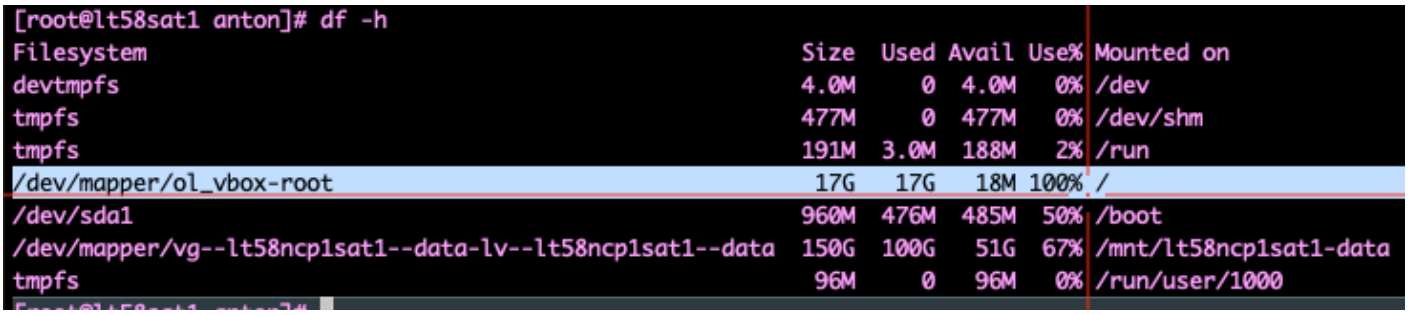
LVM - extend the volume

To extend the volume, we have to attach new physical (virtual) disk on hypervisor.

Preparations

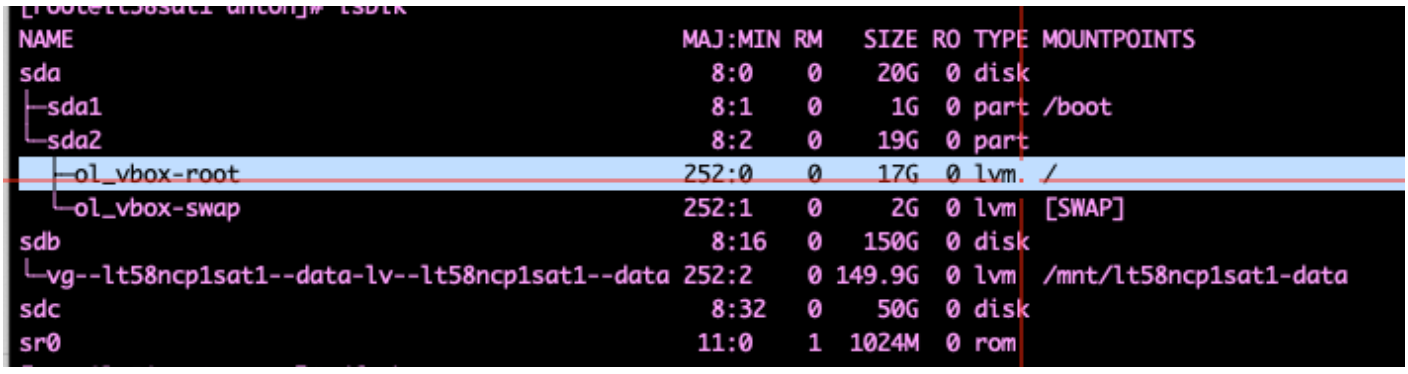
We need to understand which disk need to be extended: In my case it is a root which is full.

```
df -h
```



and verify it is LVM

```
lsblk
```



After adding the disk, it should appear in the list:

```
lsblk
```

```
[root@lt58sat1 anton]# lsblk
NAME                                          MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                                             8:0    0    20G  0 disk
├─sda1                                          8:1    0     1G  0 part /boot
└─sda2                                          8:2    0    19G  0 part
  ├─ol_vbox-root                              252:0    0    17G  0 lvm  /
  └─ol_vbox-swap                              252:1    0     2G  0 lvm  [SWAP]
sdb                                             8:16   0   150G  0 disk
└─vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data 252:2    0 149.9G  0 lvm  /mnt/lt58ncp1sat1-data
sdc                                             8:32   0    50G  0 disk
sr0                                            11:0    1  1024M  0 rom
```

Let's note which file system in use: 'XFS' or 'ext4', will be needed later..

```
lsblk -f
```

```
[root@lt58sat1 anton]# lsblk -f
NAME                                          FSTYPE      FSVER    LABEL UUID                                   FSAVAIL FSUSE% MOUNTPOINTS
sda
├─sda1                                        xfs                      03636341-0b72-4bea-9c7b-da2bf8850754    484.9M    49% /boot
└─sda2                                        LVM2_member LVM2 001     lNSByB-LaV5-PwbI-J7uK-tyer-c7TL-vg30PB
  ├─ol_vbox-root                              xfs                      8e68b127-555a-47ab-aa26-a50ece118e91     17.1M   100% /
  └─ol_vbox-swap                              swap        1            9fd91766-82f0-49e4-9281-03f20efb6632                [SWAP]
sdb                                           LVM2_member LVM2 001     eFwsQp-rLnt-WRGw-eYrG-h2Y6-qc9x-NdISvc
└─vg--lt58ncp1sat1--data-lv--lt58ncp1sat1--data xfs                      79f14e77-d9bf-4145-ace1-b10cdaa0c702     50.3G    66% /mnt/lt58ncp1sat1-data
sdc
sr0
```

We are good to go, prepare for activities. Replace your values for variables `dev` and `fs` with correct ones. Subtract `0.1` from partition size:

```
export dev="/dev/sdc"
```

Disk need to be marked as a "Physical Volume" to be able to join the "volume group" and list:

```
pvcreate ${dev}
pvscan
```

```
[root@lt58sat1 anton]# pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created.
[root@lt58sat1 anton]# pvscan
  PV /dev/sda2   VG ol_vbox               lvm2 [<19.00 GiB / 0    free]
  PV /dev/sdb    VG vg-lt58ncp1sat1-data  lvm2 [<150.00 GiB / 96.00 MiB free]
  PV /dev/sdc                             lvm2 [50.00 GiB]
  Total: 3 [218.99 GiB] / in use: 2 [168.99 GiB] / in no VG: 1 [50.00 GiB]
```

First, let's define, which 'volume group' need to be extended

```
vgscan
```

```
[root@lt58sat1 anton]# vgscan
  Found volume group "ol_vbox" using metadata type lvm2
  Found volume group "vg-lt58ncp1sat1-data" using metadata type lvm2
```

In my case, it is `ol_vbox`. Let's define it:

```
export vg="ol_vbox"
```

Let add "new Physical Volume" to the "Virtual Group":

```
echo ${vg}
echo ${dev}
vgextend ${vg} ${dev}
```

```
[root@lt58sat1 anton]# vgextend ${vg} ${dev}
  Volume group "ol_vbox" successfully extended
```

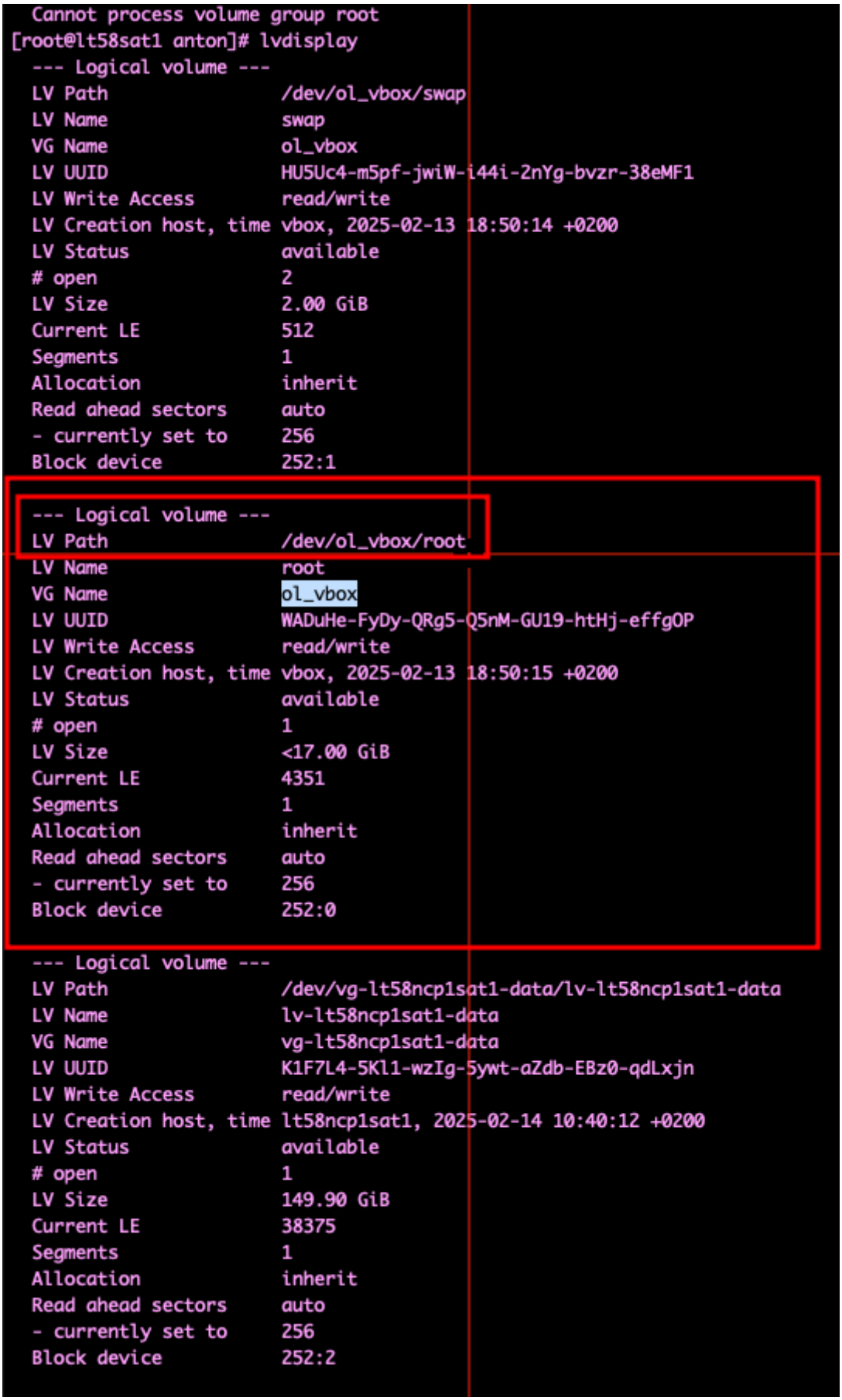
List Physical Volumes, new disk should be shown

```
pvscan
```

```
[root@lt58sat1 anton]# pvscan
  PV /dev/sda2   VG ol_vbox                lvm2 [<19.00 GiB / 0    free]
  PV /dev/sdc    VG ol_vbox                lvm2 [<50.00 GiB / <50.00 GiB free]
  PV /dev/sdb    VG vg-lt58ncp1sat1-data   lvm2 [<150.00 GiB / 96.00 MiB free]
  Total: 3 [<218.99 GiB] / in use: 3 [<218.99 GiB] / in no VG: 0 [0   ]
```

Now it is a time to extend Logical Volume. Let's observe current ones first

```
lvdisplay
```



In my case, "Volume Group" is linked to "Logical Volume" has path "/dev/ol_vbox/root". Set up extension size.

```
export lv="/dev/ol_vbox/root"
export size="49.9G"
```

and extend "Logical Volume"

```
lvextend -L +${size} ${lv}
```

```
[root@lt58sat1 anton]# lvextend -L +${size} ${lv}
  Rounding size to boundary between physical extents: 49.90 GiB.
  Size of logical volume ol_vbox/root changed from <17.00 GiB (4351 extents) to <66.90 GiB (17126 extents).
  Logical volume ol_vbox/root successfully resized.
```

Depending on which file system is in use, note current partition size, expand it and check size again:

```
df -h

# for "ext4"
resize2fx ${lv}

# for "xfs"
xfs_growfs ${lv}

df -h
```

```
[root@lt58sat1 anton]# df -h /
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/ol_vbox-root   17G   17G   18M 100% /
[root@lt58sat1 anton]# xfs_growfs ${lv}
meta-data=/dev/mapper/ol_vbox-root isize=512    agcount=4, agsize=1113856 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=1, sparse=1, rmapbt=0
         =                       reflink=1    bigtime=1 inobtcount=1 nrext64=0
         =                       exchange=0
data     =                       bsize=4096   blocks=4455424, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0, ftype=1, parent=0
log      =internal log           bsize=4096   blocks=16384, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 4455424 to 17537024
[root@lt58sat1 anton]# df -h /
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/ol_vbox-root   67G   18G   50G  26% /
```

Q.E.D.

Storage management using LVM is a logical process, once understood.

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/
```

Moving most consuming directory to the separate disk

In my case, one directory is consuming most of the disk space dedicated for the system.

Stop the service.

```
systemctl stop wazuh*
systemctl | grep wazuh
```

Rename directory

```
export src="/var/ossec"
mv ${src} ${src}.backup
```

Configure LVM, mount disk to the location guide, but do not create directory not mount the disk.

```
mkdir -p ${src}
export host=$(hostname)
mount /dev/vg-${host}-data/lv-${host}-data ${src}
df -h
```

Add mount point to the fstab. Copy needed line to the buffer and paste it into the /etc/fstab

```
mount | grep ${host}
```

```
/dev/mapper/vg--host--data-lv--host--data on /var/ossec type ext4 (rw,relatime)
```

```
vi /etc/fstab
```

```
/dev/mapper/vg--host--data-lv--host--data /var/ossec    ext4    errors=remount-ro    0 1
```

```
systemctl daemon-reload
mount -a
df -h
```

```
Filesystem                                  Size  Used Avail Use% Mounted on
/dev/root                                   29G   20G  8.5G  70% /
tmpfs                                      2.0G     0  2.0G   0% /dev/shm
tmpfs                                      783M 1000K  782M   1% /run
tmpfs                                      5.0M     0  5.0M   0% /run/lock
efivarfs                                    56K   24K   27K  48% /sys/firmware/efi/efivars
/dev/sda16                                  881M   70M  749M   9% /boot
/dev/sda15                                  105M  6.1M   99M   6% /boot/efi
tmpfs                                      392M   12K  392M   1% /run/user/1003
/dev/mapper/vg--host--data-lv--host--data            98G   24K   93G   1% /var/ossec
```

Run the screen (tmux). Move files from backup to new destination

```
tmux a
sudo su
apt install rsync
export src="/var/ossec"
# trailing slash matters! telling copy content, nor directory itself.
rsync -avAXH --progress ${src}.backup/ ${src}
```

Verify integrity (output must be empty)

```
# checksums, permissions, timestamps
rsync -avcn --delete ${src}.backup ${src}

# content only
diff -qr ${src}.backup ${src}
```

Verify size

```
du -s ${src}
```

```
8905080 /var/ossec
```

```
du -s ${src}.backup
```

```
8905676 /var/ossec.backup
```

Perform VM restart to ensure disk will be mounted and service is working properly

```
shutdown -r now
```

Once happy, remove source

```
export src="/var/ossec"
rm -rf ${src}.backup
```