

Installare e Configurare un NAS virtuale

Laboratorio di Amministrazione di Sistemi T
Massimiliano Mattetti - Marco Prandini



Percorso

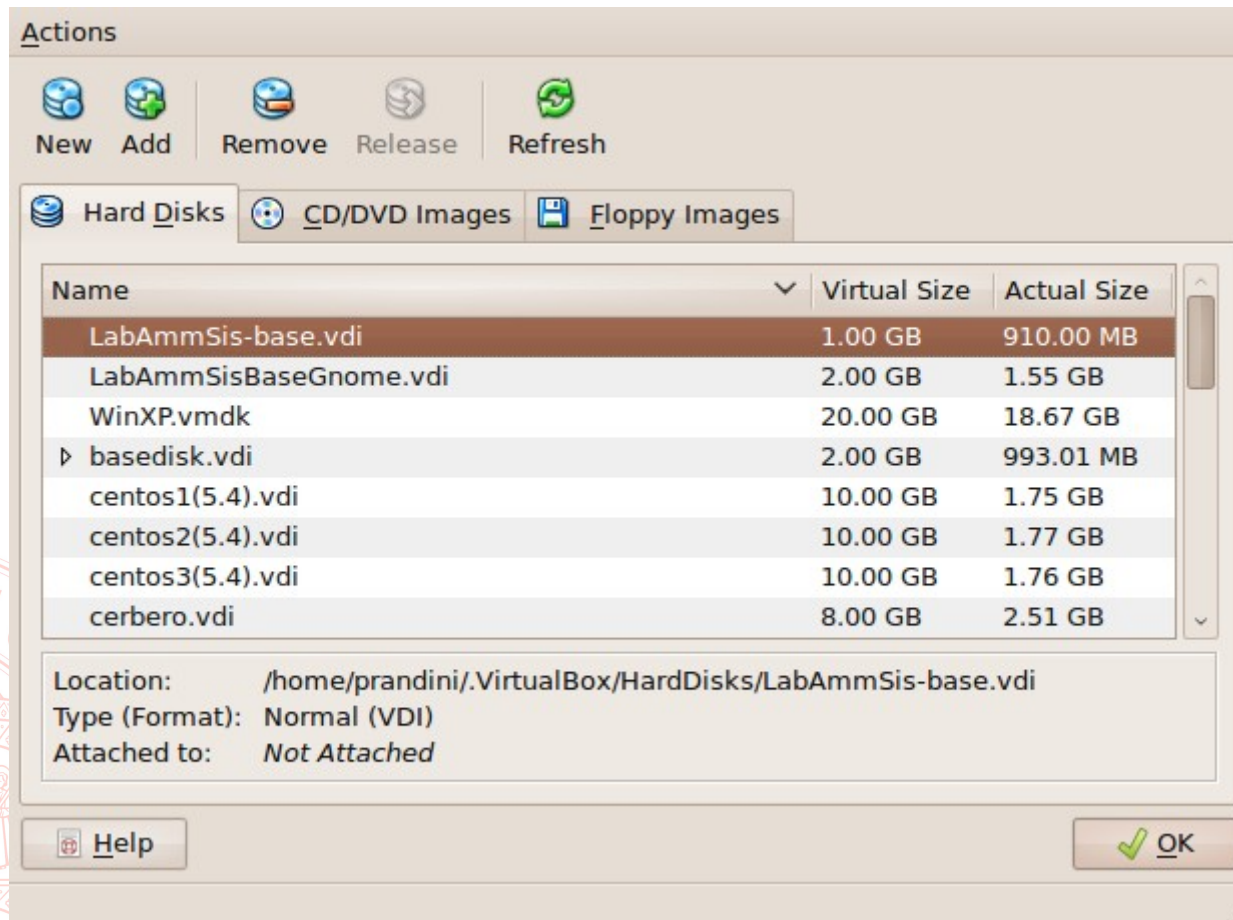
- **In questa esercitazione simuleremo la configurazione di un NAS affidabile e flessibile per mezzo dei tool Linux**
 - Utilizzeremo dischi virtuali in RAID per simulare un array tollerante ai guasti
 - Utilizzeremo LVM per realizzare un volume di dimensioni facilmente modificabili
 - Utilizzeremo NFS per condividere questo filesystem tra diversi client

- ***Tutte le seguenti operazioni saranno svolte sulla VM Router, salvo diversamente specificato (dalla slide 18 in poi)***



Creazione dei dischi virtuali (1)

- Definiamo in VirtualBox 4 nuovi hard disk (non supportato da tutte le versioni; in alternativa: crearli direttamente dalla VM)
 - File → Virtual Media Manager



Creazione dei dischi virtuali (2)

- → **New**
 - “dynamically expanded storage”
 - `~/LAS.VM/HardDisks/d0.vdi`
 - dimensione 1GB
- **Ripetere l'operazione per creare allo stesso modo d1, d2, d3**
 - ricordando sempre di collocarli in `~/LAS.VM/HardDisks/`
- **Chiudere il Virtual Media Manager**

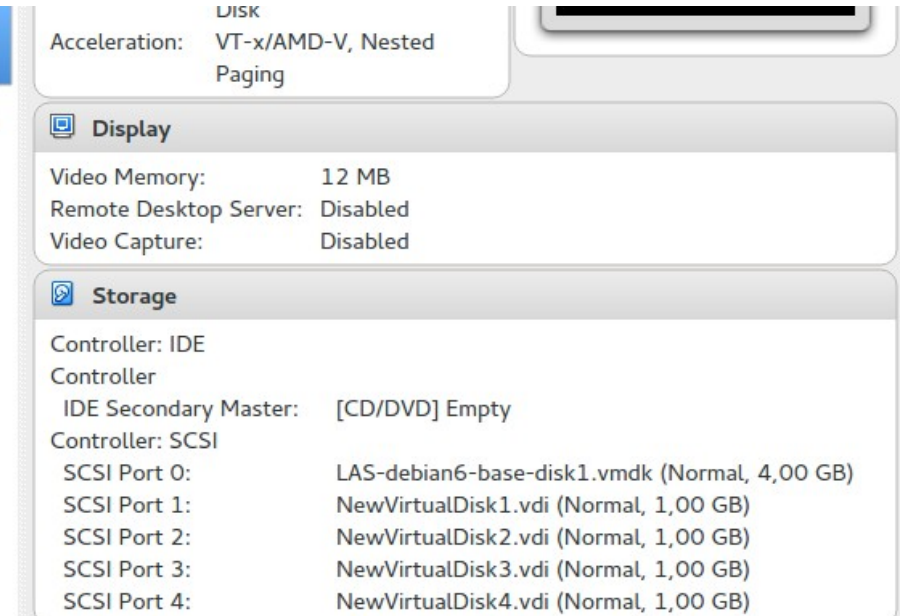
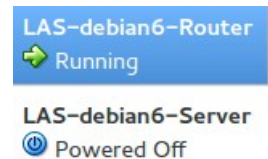
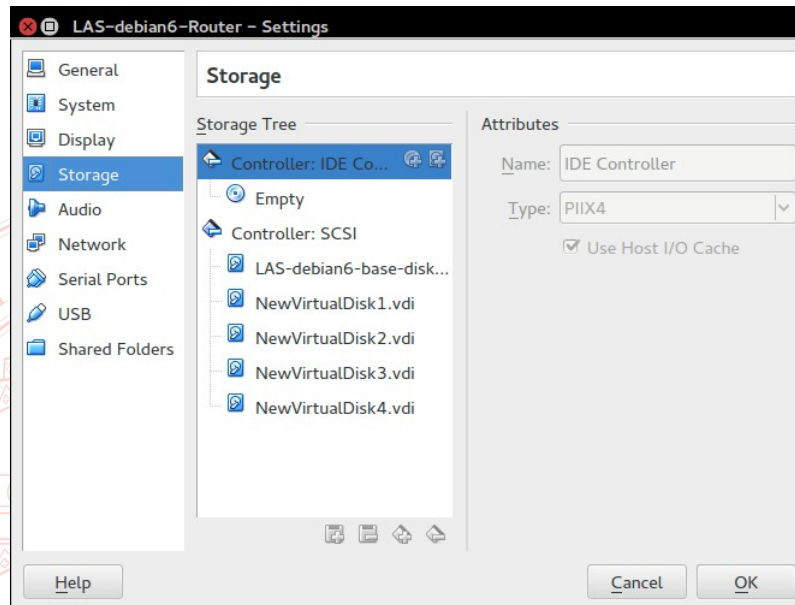


Connessione dei dischi virtuali alla VM (1)

■ Dai settaggi dello storage

- collegate i 4 dischi; seconda icona sulla riga del nuovo controller
 - choose existing disk → cercate quelli che avete creato
 - create new disk se non si poteva col VMM, usando le stesse indicazioni

■ Al termine del procedimento, la situazione dovrebbe essere:



Avvio della VM

- Durante l'avvio si può constatare se il BIOS riporta la presenza dei nuovi dischi
- Dal S.O., per verificare l'esistenza dei dischi, usare ad esempio:

```
dmesg | grep sd
```

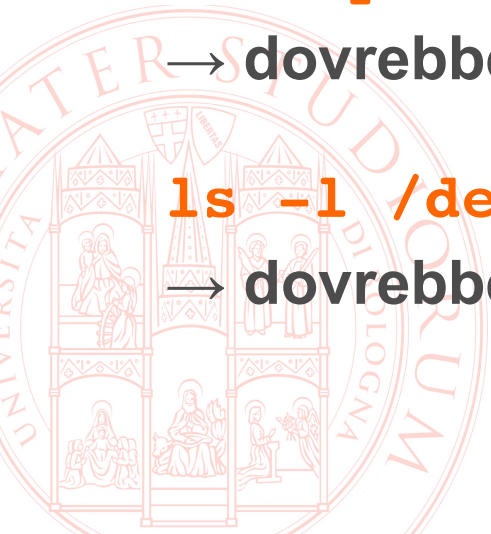
→ dovrebbe riportare i messaggi del kernel al riconoscimento dei dischi

```
cat /proc/scsi/scsi
```

→ dovrebbe riportare 4 harddisk

```
ls -l /dev/sd*
```

→ dovrebbero essere presenti i file sda, sdb, sdc, sdd



Partizionamento dei dischi (1)

Si utilizza il comando **fdisk /dev/sdb** per partizionare il primo disco

■ Comandi essenziali di fdisk:

- m: help (manual)
- p: print partition table
- n: new partition
- t: change partition type
- w: write and quit
- q: quit without saving



Partizionamento dei dischi (2)

Ipotizziamo di voler riservare parte dello spazio allo swap, e parte alla realizzazione di volumi con cui costruire un RAID:

- Creazione prima partizione → comando ***n***
 - Primo parametro richiesto: tipo di partizione → scegliere ***p*** per primaria
 - Secondo parametro richiesto: numero di partizione → ***1***
 - Terzo parametro richiesto: primo cilindro → digitare invio per il primo disponibile
 - Quarto parametro richiesto: ultimo cilindro → ad es. ***4***
- Verificate con ***p*** che la partizione sia stata creata
- Assegnamento del tipo alla partizione → Comando ***t***
 - Primo parametro: numero di partizione → ***1***
 - Secondo parametro: il tipo → ***82*** (con L si possono listare tutti i tipi)



Partizionamento dei dischi (3)

- Creazione seconda partizione → comando *n*
 - Primo parametro richiesto: tipo di partizione → scegliere *p* per primaria
 - Secondo parametro richiesto: numero di partizione → *2*
 - Terzo parametro richiesto: primo cilindro → digitare invio per il primo disponibile
 - Quarto parametro richiesto: ultimo cilindro → digitare invio per usare tutto lo spazio
- Assegnamento del tipo alla partizione → Comando *t*
 - Primo parametro: numero di partizione → *2*
 - Secondo parametro: il tipo → *fd*
- Verificate con *p* che la partizione sia stata creata

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	4	32098+	82	Linux swap / Solaris
/dev/sdb2		5	130	1012095	fd	Linux raid autodetect

- Salvate ed uscite con *w*

Partizionamento dei dischi (4)

- Clonate la stessa configurazione sugli altri tre dischi

```
sfdisk -d /dev/sdb | sfdisk /dev/sdc  
sfdisk -d /dev/sdb | sfdisk /dev/sdd  
sfdisk -d /dev/sdb | sfdisk /dev/sde
```

- Verificate il risultato con

```
fdisk -l  
ls -l /dev/sd*
```



Definizione di un metadvice RAID1

- Per realizzare un mirror con due delle partizioni create ora:

Device creato

Livello 1 = mirror

```
mdadm --create --verbose /dev/md0 --level=1 \  
--raid-devices=2 /dev/sdb2 /dev/sdc2
```

2 membri: la dichiarazione serve perchè potrei avviare l'array con uno solo, o con tre (di cui uno resta spare)

Per verificare il funzionamento del device:

```
cat /proc/mdstat
```

Per salvare lo stato running nel file di configurazione:

```
echo DEVICE /dev/sd[bcde]2 > /etc/mdadm/mdadm.conf  
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

LVM: prima definizione

- Marcatura dei block device che voglio usare come PV

```
pvcreate /dev/md0
```

- Per visualizzare i PV disponibili:

```
pvdisplay
```

- Definizione del VG ed assegnazione del PV al VG

```
vgcreate vg_raid1 /dev/md0
```

- Per visualizzare i VG disponibili:

```
vgdisplay
```

- Allocazione di un LV su di un VG

```
lvcreate -n datadisk -L 100M vg_raid1
```

- Per visualizzare i LV disponibili:

```
lvdisplay
```



Utilizzo del Logical Volume

- Il LV è utilizzabile come una normale partizione:

- va formattato

```
mkfs.ext3 /dev/vg_raid1/datadisk
```

- e montato

```
mkdir /home/las/sharedir
```

```
mount /dev/vg_raid1/datadisk /home/las/sharedir
```

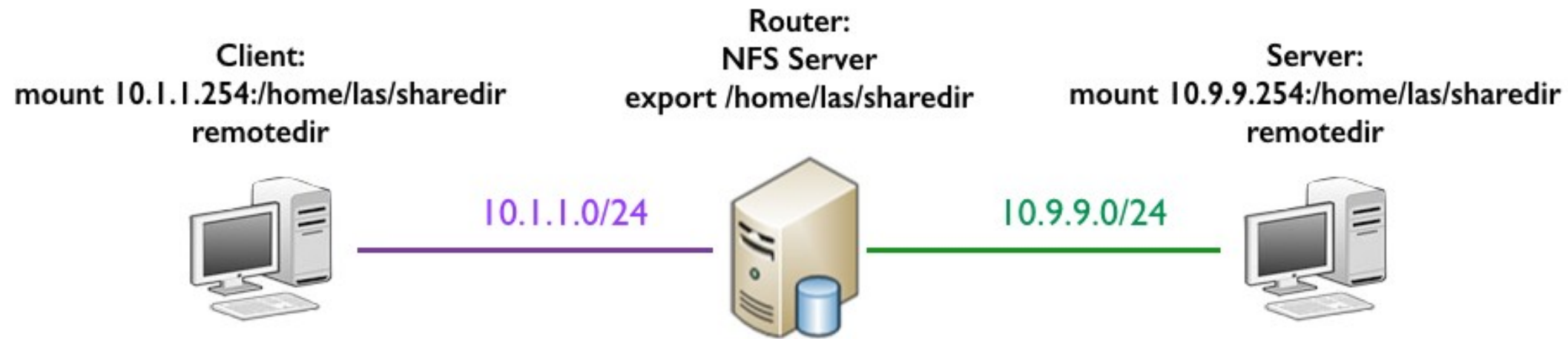
- Controlliamo lo spazio reso disponibile nel filesystem, e la corrispondente diminuzione di PE disponibili sul VG:

```
df
```

```
vgdisplay
```



NFS: Configurazione target



Configurazione NFS server su Router

- Nelle macchine virtuali sono già installati i pacchetti *portmap* e *nfs-common*

- Avviare il servizio portmap:

```
sudo service portmap start
```

per eseguirlo automaticamente all'avvio:

```
sudo update-rc.d portmap enable
```

- Scaricare sulla macchina virtuale Router il pacchetto *nfs-kernel-server* ed installarlo:

```
wget http://lia.deis.unibo.it/Courses/AmmSistemi1314/Lab/nfs/nfs-kernel-server.deb
```

```
sudo dpkg -i nfs-kernel-server.deb
```

lanciando il comando `rpcinfo -p` viene mostrata la lista dei demoni NFS registrati presso il portmapper

- Configurare `/etc/exports` aggiungendo la seguente riga:

```
/home/las/sharedir *(rw,root_squash,sync,no_subtree_check)
```

- Lanciare il comando:

```
sudo exportfs -r
```

Controllo degli accessi (1)

- Il controllo degli accessi per il portmapper e i demoni NFS può essere configurato attraverso i file `/etc/hosts.allow` e `/etc/hosts.deny`
 - Nota: questi file gestiscono l'accesso anche ad altri demoni di sistema come sshd o initd
- La policy di default è quella di consentire l'accesso a tutto ciò che non è esplicitamente negato. Una richiesta di accesso ad un demone, da parte di un client, avvia il seguente processo di valutazione:
 - 1) Viene concesso l'accesso se il client fa match con una regola di `/etc/hosts.allow`
 - 2) Altrimenti, l'accesso è negato se viene trovato un match in `/etc/hosts.deny`
 - 3) Altrimenti, viene concesso l'accesso



Controllo degli accessi (2)

- Configurare `/etc/hosts.deny` per impedire l'accesso ai demoni NFS:

```
rpcbind mountd nfsd statd lockd rquotad : ALL
```

- Configurare `/etc/hosts.allow` per consentire l'accesso ai demoni NFS agli host desiderati:

```
rpcbind mountd nfsd statd lockd rquotad : 10.1.1.1 10.9.9.1
```



Configurazione delle VM Client e Server

- Avviare i servizi portmapper e nfs-common su entrambe le macchine virtuali:

```
sudo service portmap start
```

```
sudo service nfs-common start
```

- Creare la directory *remotedir* su Client e Server ed effettuare il mount con il comando:

Su Client: `sudo mount 10.1.1.254:/home/las/sharedir remotedir`

Su Server: `sudo mount 10.9.9.254:/home/las/sharedir remotedir`

- Per abilitare il mount automatico della directory all'avvio aggiungere la seguente riga al file `/etc/fstab`

Su Client:

```
10.1.1.254:/home/las/sharedir /home/las/remotedir nfs defaults 0 0
```

Su Server:

```
10.9.9.254:/home/las/sharedir /home/las/remotedir nfs defaults 0 0
```

Test / approfondimenti proposti

- **Da Client creare un file in /home/las/remotedir**
 - Che errore si ottiene? Perché?
(studiare il significato delle opzioni in /etc/exports)
- **Da Router creare una directory /home/las/sharedir/dati**
 - Attribuire las:las come utente e gruppo proprietari
 - Verificare che su Client e Server appare in /home/las/remotedir
 - Verificare che dalle due macchine si possono creare file e vedere



Espansione dello spazio disponibile (1)

Supponiamo di avere la duplice necessità di

- allargare il filesystem appena creato
 - definire un nuovo filesystem, per un totale di dimensioni superiori allo spazio libero sul VG
- Generiamo un nuovo disco virtuale RAID1 con gli altri 2 dischi, controlliamo lo stato, e aggiorniamo il file di configurazione

```
mdadm --create --verbose /dev/md1 --level=1 \  
      --raid-devices=2 /dev/sdd2 /dev/sde2
```

```
cat /etc/mdadm/mdadm.conf
```

```
echo DEVICE /dev/sd[bcde]2 > /etc/mdadm/mdadm.conf
```

```
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

- Aggiunta del nuovo device al VG

```
pvcreate /dev/md1
```

```
vgextend vg_raid1 /dev/md1
```

- verificate sempre il risultato con **vgdisplay**

Espansione dello spazio disponibile (2)

- (1) Estensione del LV esistente e del filesystem

```
lvextend -L+200M /dev/vg_raid1/datadisk
```

Verifica con `df` : ovviamente l'allargamento del supporto non si propaga automaticamente al filesystem, per cui:

```
resize2fs /dev/vg_raid1/datadisk
```

- (2) Definizione di un nuovo LV ed uso in un nuovo filesystem

```
lvcreate -n userdisk -L 900M vg_raid1
```

```
mkfs.ext3 /dev/vg_raid1/userdisk
```

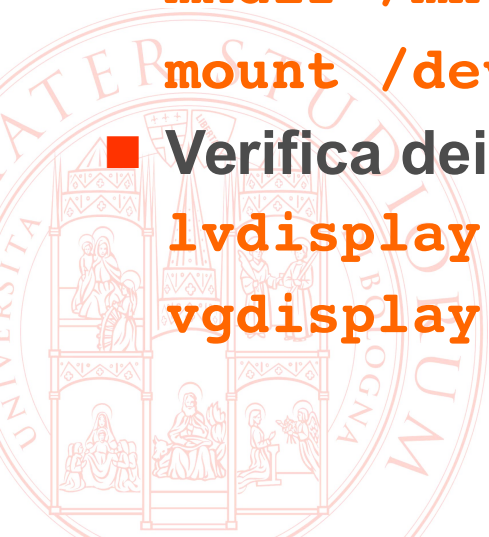
```
mkdir /mnt/userdisk
```

```
mount /dev/vg_raid1/userdisk /mnt/userdisk
```

- Verifica dei LV e dello spazio disponibile nel VG:

```
lvdisplay
```

```
vgdisplay
```



Automazione del mount

- Tipicamente il detect degli array raid e dei componenti LVM è già predisposto nei sistemi Linux. Naturalmente il mount deve essere configurato manualmente, aggiungendo al file

/etc/fstab

righe come

```
/dev/vg_raid1/datadisk /home/las/sharedir ext3 defaults 0 0  
/dev/vg_raid1/userdisk /mnt/userdisk ext3 defaults 0 0
```

- Verifica del funzionamento / esercizi proposti

- Riavviare la VM. Verificare lo stato dei RAID, lo stato di LVM, il filesystem.
- Spegnere la VM e da VirtualBox scollegare i dischi 1 e 3. Avviare e ripetere le verifiche. Creare file nelle partizioni sharedir e userdisk
- Spegnere la VM e da VirtualBox scollegare i dischi 1 e 3. Avviare e ripetere le verifiche.
 - Analizzare nel dettaglio i superblock dei membri del RAID.
 - Come si può forzare il sistema RAID a risincronizzare i dischi ricollegati?