

# **FONAMENTS DEL MAQUINARI**

## *PRACTICA 7-*

Implementació d'un sistema redundant en un CPD

Març, 2025

Victor Benjumea Gutierrez

# ÍNDEX

<b>1. Introducció.....</b>	<b>3</b>
<b>2. Configuració del sistema.....</b>	<b>4</b>
2.1 Crear dues màquines virtuals:.....	4
2.2 Configura RAID 1 al Servidor-Principal:.....	7
2.3 Configura sincronització automàtica amb rsync.....	9
<b>3. Seguretat i protecció de xarxa.....</b>	<b>14</b>
3.1 Configuració del firewall.....	14
3.2 Protecció contra atacs amb fail2ban.....	16
<b>4 Instal·lació i configuració de SNMP.....</b>	<b>18</b>
4.1 Instal·lació i configuració de SNMP.....	18
4.2 Consulta d'informació del sistema.....	20
4.3 Validació de la monitorització.....	21
<b>5. Simulació de fallades i recuperació.....</b>	<b>22</b>
<b>6. Conclusió.....</b>	<b>23</b>
<b>7. Webgrafia.....</b>	<b>23</b>

# 1. Introducció

L'objectiu d'aquesta pràctica és crear un sistema senzill i segur dins d'un Centre de Processament de Dades (CPD) . Hem d'aconseguir que el sistema estigui preparat per a errors, protegir les dades i controlar l'estat dels servidors.

Crearem dues màquines virtuals: un **Servidor-Principal** amb més recursos i un **Servidor-Backup** per fer còpies de seguretat. Es configurarà un **RAID 1** al Servidor-Principal per duplicar les dades en dos discs. També es farà una sincronització automàtica amb **rsync** cada sis hores perquè el Servidor-Backup tingui una còpia actualitzada.

Per al tema de la seguretat, usarem un **firewall** per limitar l'accés entre servidors i s'instal·larà **fail2ban** per protegir el servei **SSH** d'atacs. A més, configurarem **SNMP** per controlar l'estat del Servidor-Principal des del Servidor-Backup.

Finalment, simularem errors per comprovar i aplicar la configuració que hem fet.

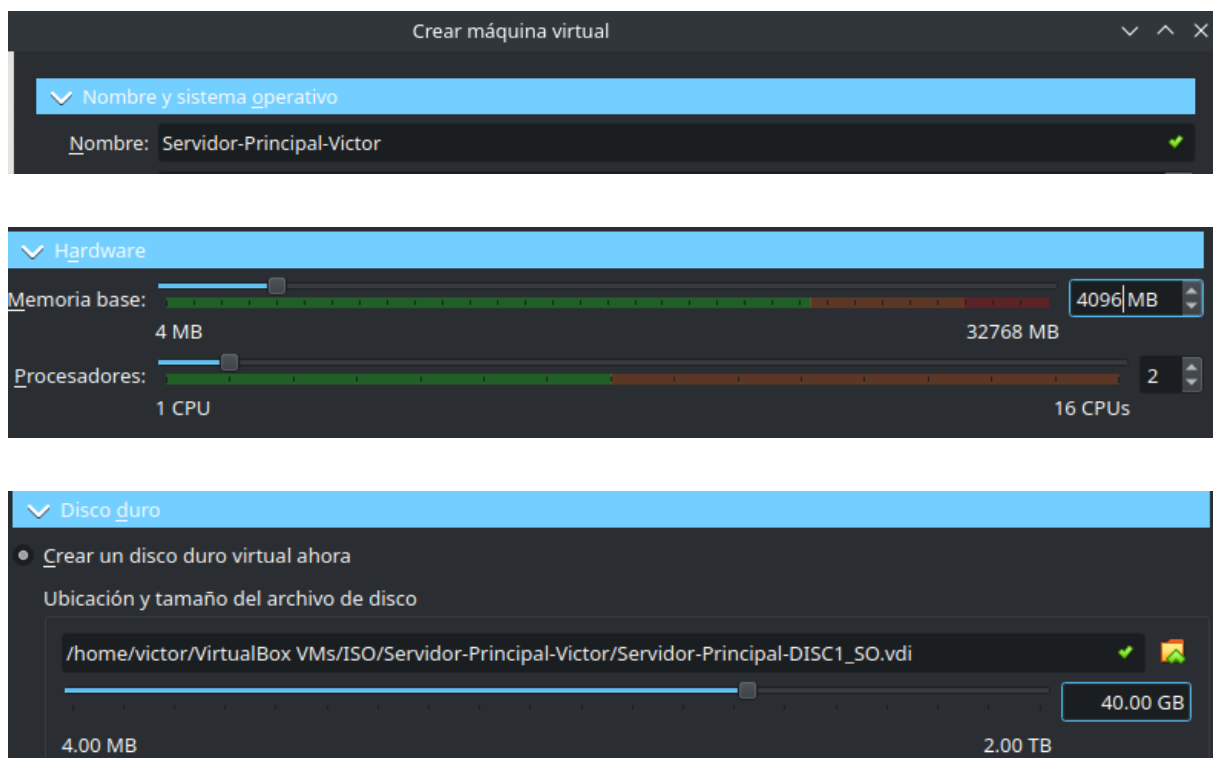
## 2. Configuració del sistema

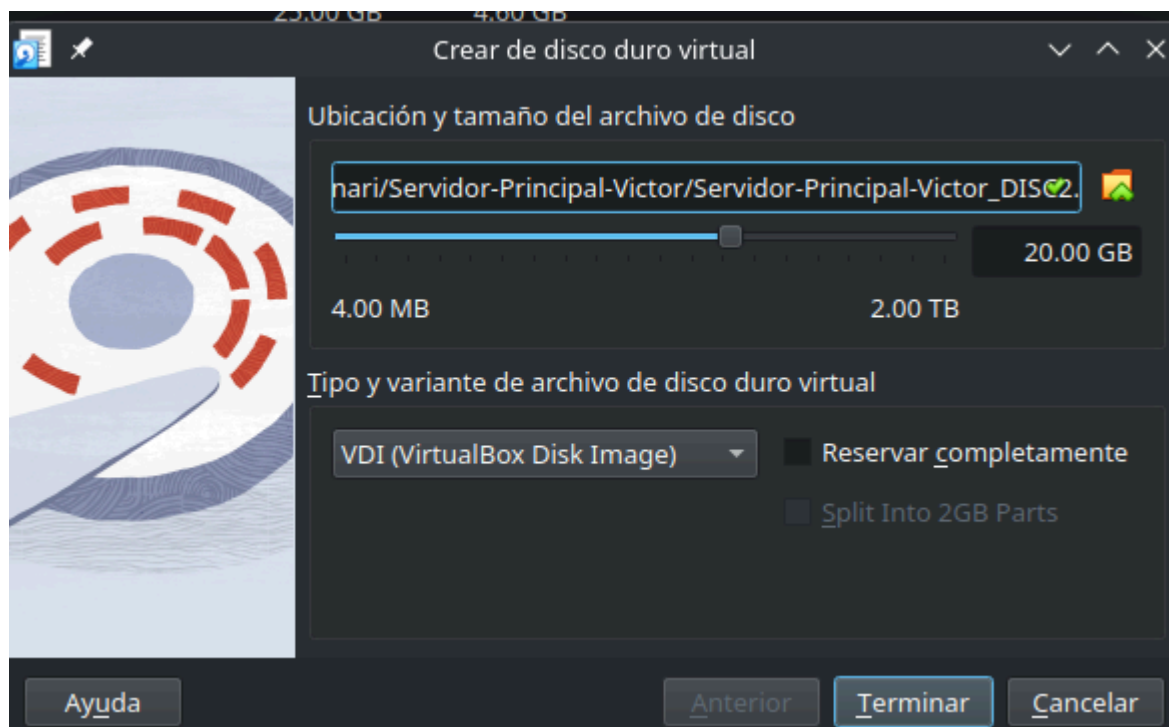
### 2.1 Crear dues màquines virtuals:

Primer crearem dues màquines virtuals amb els recursos especificats. El Servidor-Principal tindrà més potència perquè gestionarà el RAID i la sincronització, mentre que el Servidor-Backup serà més senzill i servirà com a còpia de seguretat.

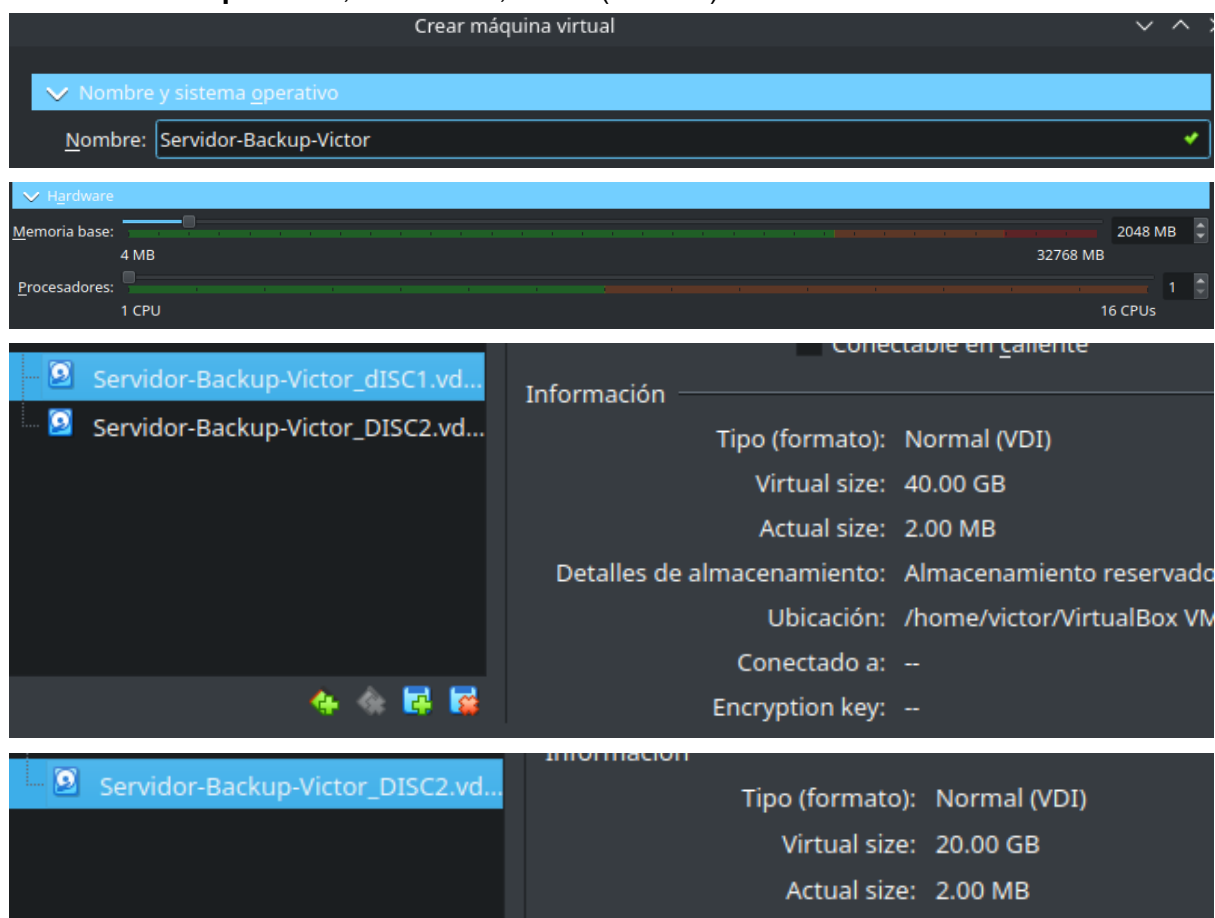
Les especificacions són les següents:

**Servidor-Principal:** 2 CPU, 4 GB RAM, 2 discos (40 GB SO, 100 GB dades).

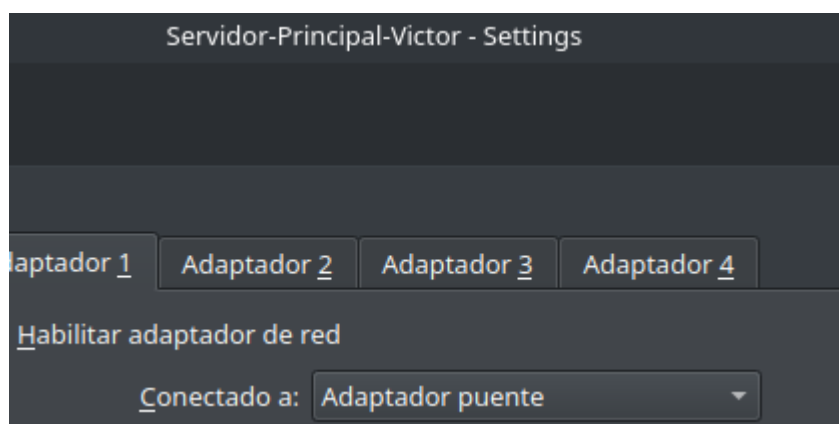
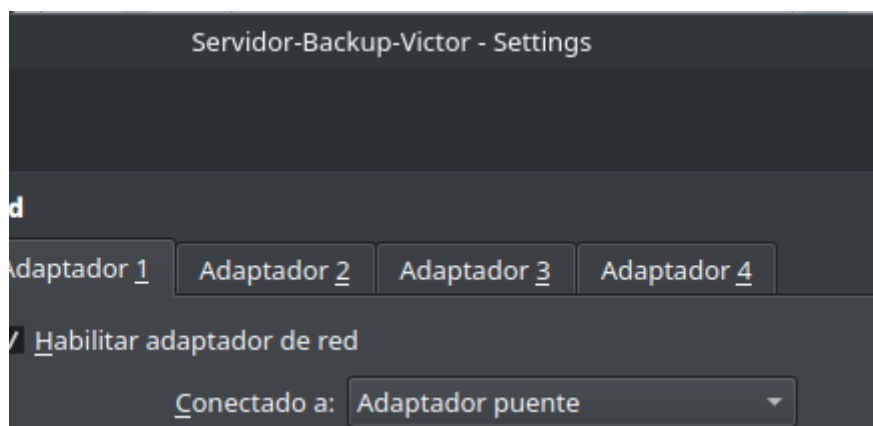




**Servidor-Backup:** 1 CPU, 2 GB RAM, 1 disc (100 GB).



Ara hem de configurar la xarxa en mode "**Bridged**" a les dues màquines per permetre la comunicació directa entre elles i la xarxa local.



## 2.2 Configura RAID 1 al Servidor-Principal:

Primer, es comprovaran els discos disponibles al sistema executant la comanda “lsblk”

```
server_principal-victor@victor1:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda                                 8:0      0   40G  0 disk
├─sda1                             8:1      0    1M  0 part
├─sda2                             8:2      0    2G  0 part /boot
├─sda3                             8:3      0   38G  0 part
└─ubuntu--vg-ubuntu--lv 252:0    0   19G  0 lvm /
sdb                                 8:16     0   20G  0 disk
sdc                                 8:32     0   20G  0 disk
sr0                                11:0     1 1024M  0 rom
```

Aquesta comanda mostrarà una llista dels dispositius d'emmagatzematge i les seves particions. Es buscaran els dos discos que es volen unir en RAID 1 (per exemple, /dev/sda i /dev/sdb).

Fet això, hem de descarregar el paquet mdadm amb la comanda “sudo apt install mdadm”

```
server_principal-victor@victor1:~$ sudo apt install mdadm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Paquetes sugeridos:
  default-mta | mail-transport-agent
Se actualizarán los siguientes paquetes:
  mdadm
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 130 no actualizados.
Se necesita descargar 464 kB de archivos.
Se utilizarán 0 B de espacio de disco adicional después de esta operación.
Des:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 mdadm amd64 4.3-1ubuntu2
Descargados 464 kB en 0s (1.556 kB/s)
Preconfigurando paquetes ...
(Leyendo la base de datos ... 80978 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../mdadm_4.3-1ubuntu2.1_amd64.deb ...
Desempaquetando mdadm (4.3-1ubuntu2.1) sobre (4.3-1ubuntu2) ...
```

Ara amb la comanda “sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc” crearem el raid 1 que ens demana a la pràctica, es level 1 perquè es raid 1 i els dos dispositius que volem raidear son sdb i sdc.

```
server_principal-victor@victor1:~$ sudo mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
Continue creating array?
Continue creating array? (y/n)
Continue creating array? (y/n) y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
server_principal-victor@victor1:~$
```

Un cop creat el RAID, cal formatar-lo amb el sistema de fitxers **ext4** i muntar-lo en un directori del sistema. Ho fem amb la comanda “sudo mkfs.ext4 /dev/md127”. md127 es el raid creat

```
server_principal-victor@victor1:~$ sudo mkfs.ext4 /dev/md127
mkfs 1.47.0 (5-Feb-2023)
Creating filesystem with 5238528 4k blocks and 1310720 inodes
Filesystem UUID: d824fed1-bfc7-4fff-b217-bd54c2a1ae70
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

server_principal-victor@victor1:~$
```

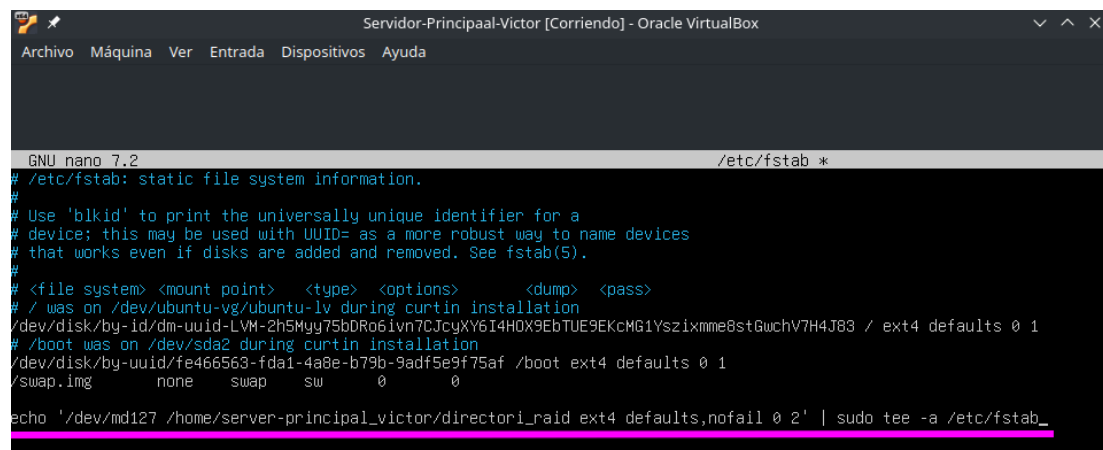
A continuació, crearem un directori per muntar-hi el RAID mab la comanda “sudo mkdir directori\_raid”

```
server_principal-victor@victor1:~$ sudo mkdir directori_raid
server_principal-victor@victor1:~$
```

I el muntem manualment amb “sudo mount /dev/md127 /home/server-principal\_victor/directori\_raid”

```
server_principal-victor@victor1:~$ sudo mount /dev/md127 /home/server_principal-victor/directori_raid/
server_principal-victor@victor1:~$
```

Ara ,per assegurar que el RAID es munti automàticament en cada reinici, cal afegir-lo al fitxer /etc/fstab. Això es pot fer amb la següent ordre “echo '/dev/md0 /mnt/dades ext4 defaults,nofail 0 2' | sudo tee -a /etc/fstab” al fitxer fstab



```
Servidor-Principaal-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /etc/fstab *
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/ubuntuv-g/ubuntuv-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-2h5Myg75bDRo6ivn7CJcyY6I4H0X9EbTUE9EKcMG1Yszixmme8stGuchV7H4J83 / ext4 defaults 0 1
# /boot was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/fe466563-fda1-4a8e-b79b-9adf5e9f75af /boot ext4 defaults 0 1
/swap.img none swap sw 0 0

echo '/dev/md127 /home/server-principal_victor/directori_raid ext4 defaults,nofail 0 2' | sudo tee -a /etc/fstab_
```



## 2.3 Configura sincronització automàtica amb rsync

L'objectiu és copiar /mnt/dades/ del Servidor-Principal al Servidor-Backup cada 6 hores.

Primer, ens hem d'assegurar que rsync està instal·lat en tots dos servidors amb “sudo apt install rsync -y”.

Primer vull aclarir que victor1 es el server principal i victor 2 es el server backup

```
server_principal-victor@victor1:~$ sudo apt install rsync -y
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
rsync ya está en su versión más reciente (3.2.7-1ubuntu1.2).
fijado rsync como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 128 no actualizados.
server_principal-victor@victor1:~$
```

PRINCIPAL

```
server_principal-victor@victor2:~$ sudo apt -y install rsync
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
rsync ya está en su versión más reciente (3.2.7-1ubuntu1.2).
fijado rsync como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 131 no actualizados.
server_principal-victor@victor2:~$
```

BACKUP

Com rsync farà servir ssh per transferir els fitxers, cal configurar l'accés sense contrasenya mitjançant claus SSH. Amb la següent comanda generarem una clau ssh “ssh-keygen -t rsa -b 4096”

```
server_principal-victor@victor2:/home$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/server_principal-victor/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/server_principal-victor/.ssh/id_rsa
Your public key has been saved in /home/server_principal-victor/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:KSnkHoJ9tVrgCn2FaH/bYWcANKj464HcoSlu5dbcRA4 server_principal-victor@victor2
The key's randomart image is:
+---[RSA 4096]-----+
|      o+          |
|     ... o        |
|    .o.+ o .      |
|   .=. E + o      |
|  o.+ o X S o     |
| .o=B.* B +      |
| o+=o= + .       |
| o..+ o .        |
| .oo             |
+---[SHA256]-----+
```

Ara copiem la clau al server backup amb la comanda “ssh-copy-id server\_principal-victor@servidor-backup”

Si no va ,abans s'ha de comprovar si esta instal·lat el server ssh.

```
server_principal-victor@victor2:~$ sudo systemctl status ssh
[sudo] password for server_principal-victor:
Unit ssh.service could not be found.
server_principal-victor@victor2:~$ sudo apt install openssh-server -y
```

Un cop instal·lat, iniciem el servei:

```
server_principal-victor@victor2:~$ sudo systemctl enable --now ssh
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh
server_principal-victor@victor2:~$ _
```

Comprovem que esta activat amb “sudo systemctl status ssh”

```
server_principal-victor@victor2:~$ sudo systemctl status ssh
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Thu 2025-03-13 12:02:59 UTC; 46s ago
 TriggeredBy: • ssh.socket
   Docs: man:sshd(8)
        man:sshd_config(5)
  Main PID: 2046 (sshd)
    Tasks: 1 (limit: 2272)
   Memory: 1.2M (peak: 1.5M)
      CPU: 17ms
   CGroup: /system.slice/ssh.service
           └─2046 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

mar 13 12:02:59 victor2 systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
mar 13 12:02:59 victor2 sshd[2046]: Server listening on :: port 22.
mar 13 12:02:59 victor2 systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
server_principal-victor@victor2:~$
```

Permetem els tallafocs al port 22 amb les següents comandes:

```
server_principal-victor@victor2:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
server_principal-victor@victor2:~$ sudo ufw reload
Firewall not enabled (skipping reload)
server_principal-victor@victor2:~$
```

Si us continua passant, és perquè hem d'esborrar la clau antiga i tornar a connectar. Executarem aquesta comanda al **Servidor-Principal** “ssh-keygen -R 192.168.1.24”

```
server_principal-victor@victor1:~$ ssh-keygen -R 192.168.1.24
Cannot stat /home/server_principal-victor/.ssh/known_hosts: No such file or directory
server_principal-victor@victor1:~$ mkdir -p ~/.ssh
```

Fet això, ara ja podem tornar a la màquina principal i provar de veure si ja connecta per ssh. Podem observar a la imatge que ja funciona:

```
server_principal-victor@victor2:~$ ssh -p 22 server_principal-victor@172.16.101.136
The authenticity of host '172.16.101.136 (172.16.101.136)' can't be established.
ED25519 key fingerprint is SHA256:gQu2kN0xByfXCBKhG7b1l8Z7D4A4DuSXbs7CWxX6UW8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.16.101.136' (ED25519) to the list of known hosts.
server_principal-victor@172.16.101.136's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-55-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of vie 14 mar 2025 19:56:16 UTC

System load:  0.06               Processes:            128
Usage of /:   34.0% of 18.5GB    Users logged in:     1
Memory usage: 5%                IPv4 address for enp0s3: 172.16.101.136
Swap usage:   0%

El mantenimiento de seguridad expandido para Applications está desactivado

Se pueden aplicar 122 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»
```

Ara hem de copiar la nostra clau pública al **Servidor Principal** perquè rsync pugui connectar-se sense demanar la contrasenya cada vegada. Utilitzarem la següent comanda “ssh-copy-id -p 2222 server\_principal-victor@172.16.101.136

```
server_principal-victor@victor1:~$ ssh-copy-id -p 22 server_principal-victor@172.16.101.136
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/server_principal-victor/.ssh/id_rsa.pub"
The authenticity of host '172.16.101.136 (172.16.101.136)' can't be established.
ED25519 key fingerprint is SHA256:gQu2kN0xByfXCBKhG7b1l8Z7D4A4DuSXbs7CWxX6UW8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
The authenticity of host '172.16.101.136 (172.16.101.136)' can't be established.
ED25519 key fingerprint is SHA256:gQu2kN0xByfXCBKhG7b1l8Z7D4A4DuSXbs7CWxX6UW8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
The authenticity of host '172.16.101.136 (172.16.101.136)' can't be established.
ED25519 key fingerprint is SHA256:gQu2kN0xByfXCBKhG7b1l8Z7D4A4DuSXbs7CWxX6UW8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Failed to add the host to the list of known hosts (/home/server_principal-victor/.ssh/known_hosts).
server_principal-victor@172.16.101.136's password:
hostfile_replace_entries: link /home/server_principal-victor/.ssh/known_hosts to /home/server_principal-victor/.ssh/known_hosts
update_known_hosts: hostfile_replace_entries failed for /home/server_principal-victor/.ssh/known_hosts: Operation not permitted
Number of key(s) added: 1

Now try logging into the machine, with:  "ssh -p 22 'server_principal-victor@172.16.101.136'"
and check to make sure that only the key(s) you wanted were added.
```

Ara hem de comprovar si la sincronització entre els servidors funciona correctament abans d'automatitzar-la. Hem d'executar la següent comanda al Servidor Backup “rsync -avz -e "ssh -p 2222" --delete server\_principal-victor@172.16.101.136:/mnt/dades/ /mnt/dades/”.

Es pot observar que no s'ha transeferit res, ja que el directori que he creat expresament està buit. Hem comprovat que la sincronització va correctament.

```
server_principal-victor@victor2:~$ sudo rsync -avz -e "ssh -p 22" --delete server_principal-victor@192.168.1.25:/mnt/dades/ /mnt/dades/
The authenticity of host '192.168.1.25 (192.168.1.25)' can't be established.
ED25519 key fingerprint is SHA256:gQu2kN0xBYfXCBKhG7b1l8Z7D4A40uSXbs7CHxX6UW8.
This host key is known by the following other names/addresses:
  ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.25' (ED25519) to the list of known hosts.
server_principal-victor@192.168.1.25's password:
receiving incremental file list
./
sent 27 bytes  received 54 bytes  3,06 bytes/sec
total size is 0  speedup is 0,00
server_principal-victor@victor2:~$ _
```

Si creem un arxiu podem veure que si es sincronitza:

```
server_principal-victor@victor2:~$ sudo touch /mnt/dades/holaa
server_principal-victor@victor2:~$ sudo rsync -avz -e "ssh -p 22" --delete server_principal-victor@192.168.1.25:/mnt/dades/ /mnt/dades/
server_principal-victor@192.168.1.25's password:
receiving incremental file list
deleting holaa
./
sent 27 bytes  received 58 bytes  24,29 bytes/sec
total size is 0  speedup is 0,00
server_principal-victor@victor2:~$ _
```

Ara hem d'automitzar tot això. Primer, hem d'editar el **crontab** de l'usuari per afegir la tasca programada. Això ho fem amb la comanda següent “crontab -e”.

Aquesta comanda obrirà l'editor de text per modificar les tasques cron de l'usuari actual. Si és la primera vegada que l'edites, potser et preguntarà quin editor vols utilitzar (per exemple, nano o vim). Utilitzaré nano:

```
server_principal-victor@victor2:~$ crontab -e
no crontab for server_principal-victor - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 1
```

Un cop dins de l'editor, afegeixo aquesta línia al final del fitxer:

```
0 */6 * * * rsync -avz -e "ssh -p 22" --delete
server_principal-victor@192.168.1.25:/mnt/dades/ /mnt/dades/
```

```

Servidor-Backup-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /tmp/crontab.0KnEqD/crontab *
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 */6 * * * rsync -avz -e "ssh -p 22" --delete server_principal-victor@192.168.1.25:/mnt/dades/ /mnt/dades/

```

Per assegurar-nos que la tasca s'ha guardat, executarem “crontab -l”

```

server_principal-victor@victor2:~$ crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 */6 * * * rsync -avz -e "ssh -p 22" --delete server_principal-victor@192.168.1.25:/mnt/dades/ /mnt/dades/

```

Per comprovar si realment funciona, executem aquesta comanda per forçar l'execució de les tasques horàries (si la tasca es trobés a /etc/cron.hourly/):

```

server_principal-victor@victor2:~$ sudo run-parts --report /etc/cron-hourly
run-parts: failed to open directory /etc/cron-hourly: No such file or directory

```

## 3. Seguretat i protecció de xarxa

### 3.1 Configuració del firewall

En aquest apartat configurarem el tallafor (ufw) per protegir els servidors i afegirem fail2ban per evitar atacs a SSH.

El primer pas és assegurar-nos que només es permet el tràfic necessari als nostres servidors mitjançant ufw .El fem als dos servers “sudo apt install -y ufw”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo apt install -y ufw
[sudo] password for server_principal-victor:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.36.2-6).
fijado ufw como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 128 no actualizados.
server_principal-victor@victor1:~$
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo apt install -y ufw
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ufw ya está en su versión más reciente (0.36.2-6).
fijado ufw como instalado manualmente.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 129 no actualizados.
server_principal-victor@victor2:~$ _
```

Ara configurarem ufw perquè per defecte bloquegi totes les connexions entrants i només permeti les sortints. Ho fem amb aquestes dues ordres “sudo ufw default deny incoming” “sudo ufw default allow outgoing”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
server_principal-victor@victor1:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
server_principal-victor@victor1:~$ _
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
server_principal-victor@victor2:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
server_principal-victor@victor2:~$
```

entrant

sortint

D'aquesta manera, cap connexió externa podrà accedir als nostres servidors tret que nosaltres l'hi permetem explícitament.

Ara volem que els nostres servidors es puguin comunicar entre ells dins de la nostra xarxa local (192.168.1.X). Per això afegim aquesta regla:

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo ufw allow from 192.168.1.0/24 to any
Rules updated
server_principal-victor@victor1:~$ _
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo ufw allow from 192.168.1.0/24 to any
Rules updated
server_principal-victor@victor2:~$ _
```

Amb això ens assegurem que els dispositius dins de la nostra xarxa poden parlar amb els servidors sense problemes.

Necessitem poder connectar-nos als servidors de forma remota mitjançant SSH. Per això, permetem les connexions al port **22** amb aquesta comanda “sudo ufw allow 22/tcp”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
server_principal-victor@victor1:~$
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo ufw allow 22/tcp
Skipping adding existing rule
Skipping adding existing rule (v6)
server_principal-victor@victor2:~$
```

Ara que ja tenim totes les regles configurades, activem el tallafoc “sudo ufw enable”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo ufw enable
Firewall is active and enabled on system startup
server_principal-victor@victor1:~$ _
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo ufw enable
Firewall is active and enabled on system startup
server_principal-victor@victor2:~$ _
```

## 3.2 Protecció contra atacs amb fail2ban

Aquests passos també s'han de fer tant al Servidor Principal com al Servidor Backup.

Ara instal·larem fail2ban, que ens ajudarà a protegir el servidor contra atacs ssh amb la comanda “sudo apt install -y fail2ban”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo apt install -y fail2ban
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
python3-pyasyncore python3-pyinotify whois
Paquetes sugeridos:
mailx msmtp esnail2 python3-pyinotify doc
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo apt install -y fail2ban
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
python3-pyasyncore python3-pyinotify whois
```

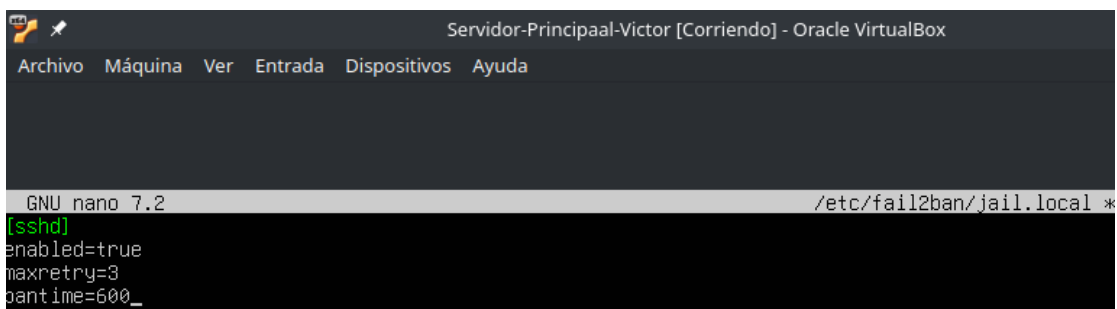
Amb fail2ban, si algú intenta accedir al nostre servidor repetidament amb contrasenyes incorrectes, es bloquejarà automàticament.

Ara hem de configurar fail2ban perquè funcioni correctament amb SSH. Editem el fitxer de configuració amb “sudo nano /etc/fail2ban/jail.local”

Afegim les línies següents:

```
[sshd]
enabled = true
maxretry = 3
bantime = 600
```

PRINCIPAL:



```
Servidor-Principaal-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /etc/fail2ban/jail.local *
[sshd]
enabled=true
maxretry=3
bantime=600_
```



BACKUP:

```
Servidor-Backup-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /etc/fail2ban/jail.local
[sshd]
enabled=true
maxretry=3
bantime=600
```

Aquí estem dient a fail2ban que, si algú falla 3 vegades intentant entrar per SSH, el bloquegi durant 10 minuts (600 segons).

Ara que ja hem configurat fail2ban, necessitem reiniciar el servei perquè els canvis tinguin efecte amb “sudo systemctl restart fail2ban”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo systemctl restart fail2ban
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo systemctl restart fail2ban
server_principal-victor@victor2:~$
```

Per veure si fail2ban està actiu i bloquejant correctament els intents fallits de connexió, executem “sudo fail2ban-client status sshd”

PRINCIPAL:

```
server_principal-victor@victor1:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|   |- Currently failed: 0
|   |- Total failed:    0
|   \- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
\-- Actions
    |- Currently banned: 0
    |- Total banned:    0
    \- Banned IP list:
```

BACKUP:

```
server_principal-victor@victor2:~$ sudo fail2ban-client status sshd
Status for the jail: sshd
|- Filter
|   |- Currently failed: 0
|   |- Total failed:    0
|   \- Journal matches: _SYSTEMD_UNIT=sshd.service + _COMM=sshd
\-- Actions
    |- Currently banned: 0
    |- Total banned:    0
    \- Banned IP list:
```

## 4 Instal·lació i configuració de SNMP

Ara configurarem SNMP per monitoritzar l'estat del nostre Servidor-Principal des del Servidor-Backup. Això ens permetrà obtenir informació com l'ús de CPU, memòria i espai en disc de forma remota.

### 4.1 Instal·lació i configuració de SNMP

Aquests passos es fan al Servidor-Principal.

El primer que farem és instal·lar el servei SNMP perquè el Servidor-Principal pugui enviar informació sobre el seu estat. Executarem la comanda “sudo apt update && sudo apt install -y snmpd”

```
server_principal-victor@victor1:~$ sudo apt install -y snmpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libsnmp-base libsnmp40t64
Paquetes sugeridos:
  snmp-mibs-downloader snmptrapd
Se instalarán los siguientes paquetes NUEVOS:
  libsnmp-base libsnmp40t64 snmpd
```

Això instal·larà snmpd, que és el dimoni de SNMP encarregat de respondre les consultes sobre el servidor.

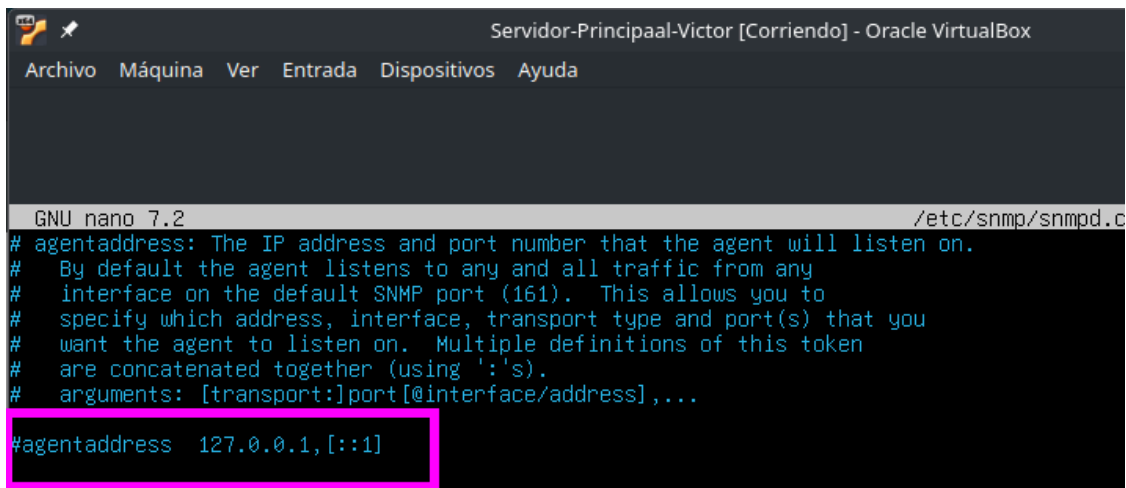
Ara configurarem SNMP perquè només pugui ser consultat des de la nostra xarxa interna i no des d'internet. Obrim el fitxer de configuració de SNMP amb “sudo nano /etc/snmp/snmpd.conf”

```
server_principal-victor@victor1:~$ sudo nano /etc/snmp/snmpd.conf
```

Modifiquem la configuració:

Busquem la següent línia i la comentem posant un # davant:

```
#agentAddress udp:127.0.0.1:161
```

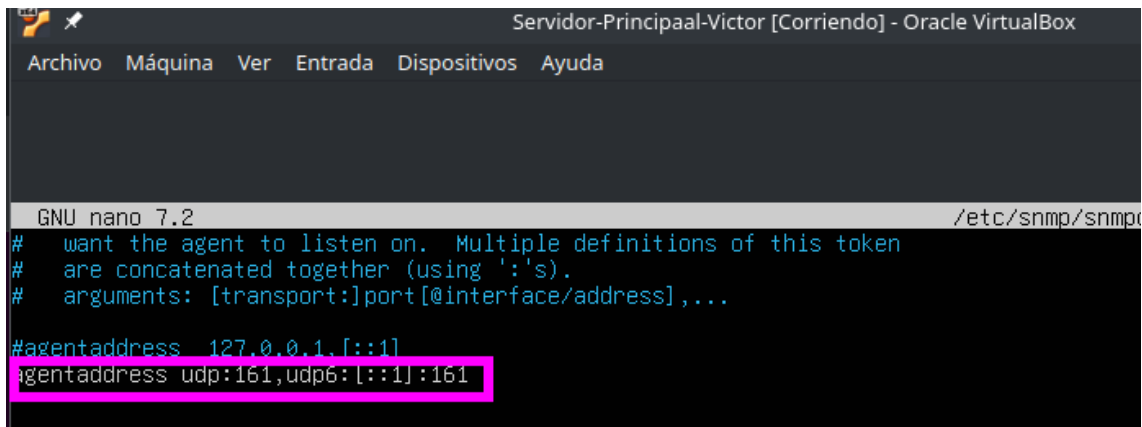


```
Servidor-Principaal-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /etc/snmp/snmpd.c
# agentaddress: The IP address and port number that the agent will listen on.
#   By default the agent listens to any and all traffic from any
#   interface on the default SNMP port (161). This allows you to
#   specify which address, interface, transport type and port(s) that you
#   want the agent to listen on. Multiple definitions of this token
#   are concatenated together (using ':'s).
#   arguments: [transport:]port[@interface/address],...
#agentaddress 127.0.0.1,[:1]
```

Afegim una nova línia per permetre peticions de qualsevol interfície del servidor:

agentAddress udp:161,udp6:[::1]:161



```
Servidor-Principaal-Victor [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda

GNU nano 7.2 /etc/snmp/snmpd.c
#   want the agent to listen on. Multiple definitions of this token
#   are concatenated together (using ':'s).
#   arguments: [transport:]port[@interface/address],...
#agentaddress 127.0.0.1,[:1]
agentaddress udp:161,udp6:[::1]:161
```

Afegim aquestes línies per definir una comunitat SNMP anomenada public només accessible des de la xarxa local

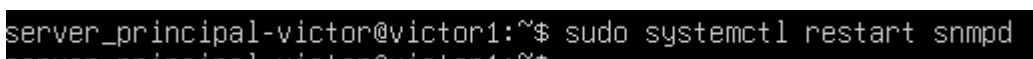
```
rocommunity public 192.168.1.0/24
sysLocation "CPD Empresa"
sysContact "admin@empresa.com"
```



```
rocommunity public 192.168.1.0/24
syslocation "CPD Empresa"
sysContact "admin@empresa.com"
^G Help      ^O Write Out  ^W Where Is
```

Amb això, només els dispositius dins de la nostra xarxa podran fer consultes SNMP al servidor.

4Reiniciem el servei perquè els canvis tinguin efecte:



```
server_principal-victor@victor1:~$ sudo systemctl restart snmpd
```

Per assegurar-nos que el servei SNMP està funcionant correctament:

```
server_principal-victor@victor1:~$ sudo systemctl status snmpd
● snmpd.service - Simple Network Management Protocol (SNMP) Daemon.
   Loaded: loaded (/usr/lib/systemd/system/snmpd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-03-18 10:56:56 UTC; 34s ago
     Main PID: 2889 (snmpd)
        Tasks: 1 (limit: 4609)
       Memory: 3.2M (peak: 3.6M)
          CPU: 21ms
      CGroup: /system.slice/snmpd.service
              └─2889 /usr/sbin/snmpd -L0w -u Debian-snmp -g Debian-snmp -I -smux mteTrigger mteTriggerConf -f

mar 18 10:56:56 victor1 systemd[1]: Starting snmpd.service - Simple Network Management Protocol (SNMP) Daemon....
mar 18 10:56:56 victor1 snmpd[2889]: systemstats_linux: unexpected header length in /proc/net/snmp. 237 != 224
mar 18 10:56:56 victor1 snmpd[2889]: systemstats_linux: unexpected header length in /proc/net/snmp. 237 != 224
mar 18 10:56:56 victor1 systemd[1]: Started snmpd.service - Simple Network Management Protocol (SNMP) Daemon..
server_principal-victor@victor1:~$
```

Podem fer una prova ràpida per veure si SNMP respon correctament amb la comanda “snmpwalk -v 2c -c public localhost | head -n 10”

```
server_principal-victor@victor1:~$ snmpwalk -v 2c -c public localhost | head -n 10
iso.3.6.1.2.1.1.1.0 = STRING: "Linux victor1 6.8.0-55-generic #57-Ubuntu SMP PREEMPT_DYNAMIC Wed Feb 12 23:42:21 UTC 202
iso.3.6.1.2.1.1.2.0 = OID: iso.3.6.1.4.1.8072.3.2.10
iso.3.6.1.2.1.1.3.0 = Timeticks: (11305) 0:01:53.05
iso.3.6.1.2.1.1.4.0 = STRING: "\"admin@empresa.com\""
iso.3.6.1.2.1.1.5.0 = STRING: "victor1"
iso.3.6.1.2.1.1.6.0 = STRING: "\"CPD Empresa\""
iso.3.6.1.2.1.1.7.0 = INTEGER: 72
iso.3.6.1.2.1.1.8.0 = Timeticks: (0) 0:00:00.00
iso.3.6.1.2.1.1.9.1.2.1 = OID: iso.3.6.1.6.3.10.3.1.1
iso.3.6.1.2.1.1.9.1.2.2 = OID: iso.3.6.1.6.3.11.3.1.1
```

## 4.2 Consulta d'informació del sistema

Primer, hem d'instal·lar les eines necessàries per fer consultes SNMP des del Servidor-Backup.Executem “sudo apt install -y snmp”

```
server_principal-victor@victor2:~$ sudo apt install -y snmp
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libsnmp-base libsnmp40t64
Paquetes sugeridos:
  snmp-gui
```

Ara provarem a obtenir dades de rendiment del Servidor-Principal des del Servidor-Backup.Exequeutem “ snmpget -v 2c -c public 192.168.1.X 1.3.6.1.4.1.2021.11.9.0” per a obtenir l'ús actual de CPU

```
server_principal-victor@victor2:~$ snmpget -v 2c -c public 192.168.1.25 1.3.6.1.4.1.2021.11.9.0
iso.3.6.1.4.1.2021.11.9.0 = INTEGER: 0
server_principal-victor@victor2:~$
```

Obtenir la memòria RAM disponible :“snmpget -v 2c -c public 192.168.1.25 1.3.6.1.4.1.2021.4.6.0”

```
server_principal-victor@victor2:~$ snmpget -v 2c -c public 192.168.1.25 1.3.6.1.4.1.2021.4.6.0
iso.3.6.1.4.1.2021.4.6.0 = INTEGER: 3162964
server_principal-victor@victor2:~$
```

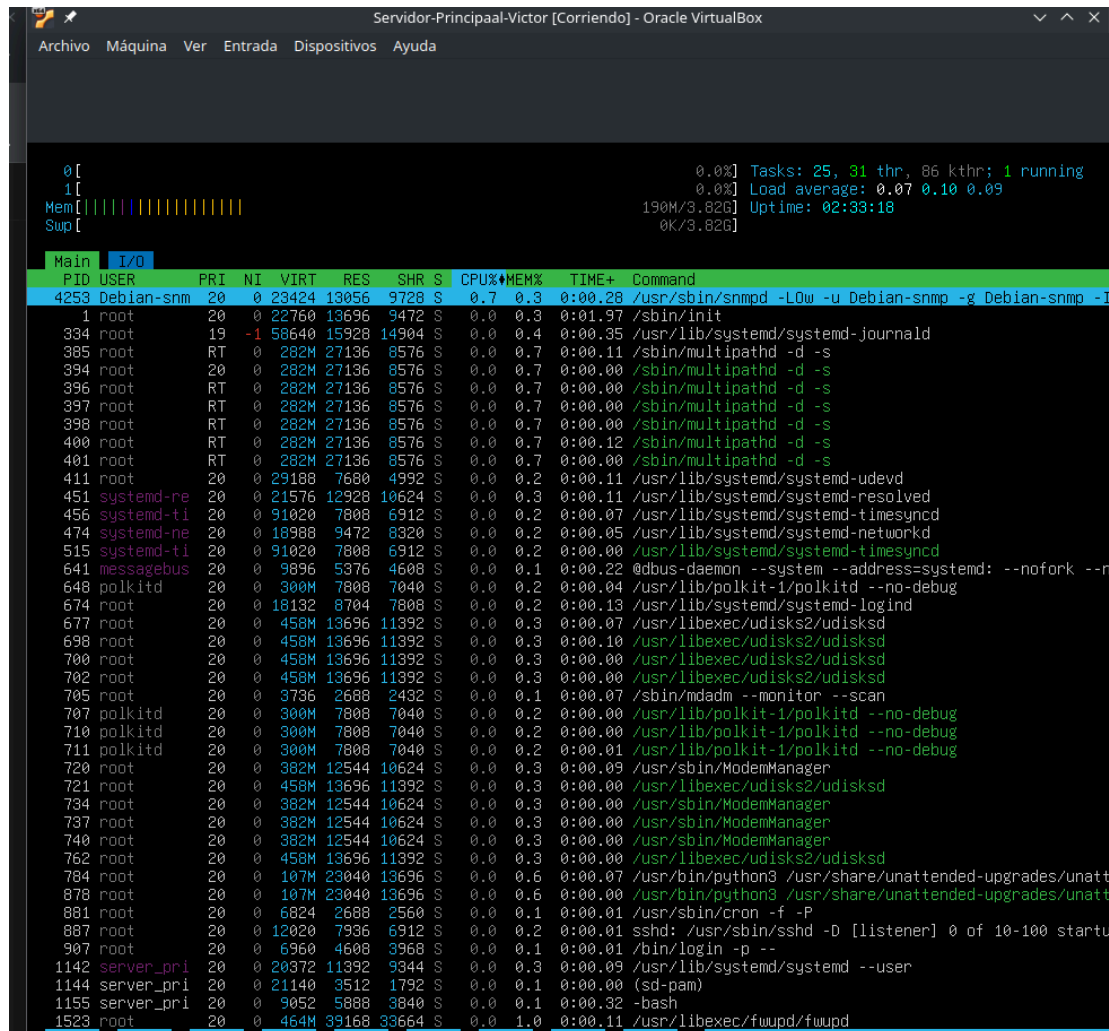
Obtenir l'espai lliure al sistema de fitxers:”snmpget -v 2c -c public 192.168.1.25 1.3.6.1.4.1.2021.9.1.7.1”

```
server_principal-victor@victor2:~$ snmpget -v 2c -c public 192.168.1.25 1.3.6.1.4.1.2021.9.1.7.1
iso.3.6.1.4.1.2021.9.1.7.1 = No Such Instance currently exists at this OID
server_principal-victor@victor2:~$
```

## 4.3 Validació de la monitorització

Ara comprovarem que les dades obtingudes via SNMP són correctes comparant-les amb les eines locals del servidor.

Executem al server-principal “htop”



Memòria RAM disponible

```
server_principal-victor@victor1:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           3,8Gi         413Mi         3,0Gi         1,1Mi         641Mi         3,4Gi
Swap:          3,8Gi           0B         3,8Gi
server_principal-victor@victor1:~$ _
```

Espai lliure en el disc

```
server_principal-victor@victor1:~$ df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     392M        1M  391M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 19G       6,4G   12G  37% /
tmpfs                     2,0G         0   2,0G   0% /dev/shm
tmpfs                     5,0M         0   5,0M   0% /run/lock
/dev/sda2                 2,0G       96M   1,7G   6% /boot
tmpfs                     392M       12K  392M   1% /run/user/1000
server_principal-victor@victor1:~$
```

## 5. Simulació de fallades i recuperació

Primer, comprovem l'estat del RAID amb la comanda “cat /proc/mdstat” per verificar en quin dispositiu està muntat el RAID.

Després, utilitzem “sudo mdadm --fail /dev/sdb” per marcar el disc /dev/sdb com a fallit. Aquesta acció simula una fallada de disc en el sistema RAID.

```
server_principal-victor@victor1:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
sda                                 8:0      0   40G  0 disk
├─sda1                             8:1      0    1M  0 part
├─sda2                             8:2      0    2G  0 part  /boot
├─sda3                             8:3      0   38G  0 part
└─ubuntu--vg-ubuntu--lv 252:0      0   19G  0 lvm    /
sdb                                 8:16     0   20G  0 disk
└─md0                              9:0      0   20G  0 raid1
sdc                                 8:32     0   20G  0 disk
└─md0                              9:0      0   20G  0 raid1
sr0                                 11:0     1 1024M  0 rom
server_principal-victor@victor1:~$ cat /proc/mdstat
Personalities : [raid1] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb[0] sdc[1]
      20954112 blocks super 1.2 [2/2] [UU]

unused devices: <none>
server_principal-victor@victor1:~$ sudo mdadm --fail /dev/md
md/ md0
server_principal-victor@victor1:~$ sudo mdadm --fail /dev/md
md/ md0
server_principal-victor@victor1:~$ sudo mdadm --fail /dev/md0
[sudo] password for server_principal-victor:
```

Comprovem l'estat del raid amb “cat /proc/mdstat”

```
server_principal-victor@victor1:~$ cat /proc/mdstat
Personalities : [raid1] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb[0] sdc[1]
      20954112 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

Ara amb la comanda sudo mdadm -remove /dev/md0 retirem el disc del raid que hem fet que falli i amb la mateixa però -add el tornem a afegir.

```
server_principal-victor@victor1:~$ sudo mdadm --remove /dev/md0
server_principal-victor@victor1:~$ sudo mdadm --add /dev/md0
server_principal-victor@victor1:~$ _
```

## 6. Conclusió

En aquesta pràctica, hem configurat un sistema de còpia de seguretat i monitorització entre dos servidors virtuals: el Servidor-Principal i el Servidor-Backup.

Primer, vam configurar les dues màquines virtuals amb els recursos que necessitaven. També vam configurar una xarxa en mode Bridged perquè els dos servidors poguessin comunicar-se entre ells i amb la xarxa local.

Després, vam crear un RAID 1 al Servidor-Principal, que consisteix en tenir dos discs perquè, si un falla, les dades continuïn estant disponibles. També vam configurar tot perquè el RAID es munti automàticament quan el servidor s'arrenca.

A continuació, vam configurar la sincronització automàtica de dades entre els dos servidors mitjançant rsync, de manera que cada 6 hores es copiïn les dades del Servidor-Principal al Servidor-Backup sense necessitat d'un humà.

Per millorar la seguretat, vam configurar el tallafoc ufw per restringir les connexions entrants només a les necessàries i vam instal·lar fail2ban per protegir els servidors contra intents d'accés no autoritzats per SSH.

A més, vam configurar SNMP per poder monitoritzar l'estat del Servidor-Principal des del Servidor-Backup, obtenint informació en temps real sobre l'ús de la CPU, la memòria i l'espai en disc.

Finalment, vam simular una fallada de disc al RAID per comprovar que el sistema es recupera correctament. Això ho vam fer manipulant els discs amb la comanda mdadm, comprovant que el sistema pot continuar funcionant malgrat els errors.

## 7. Webgrafia

<https://github.com/HectorPascuallesCarlesVallbona/ASX01-fonaments-maquinari/blob/main/01-RAs/RA04/02-practica/02-implementacio-servidor-alt-rendiment.md>