

Práctica SI Seguridad - RAID por Software: mdadm

ESCENARIO

Máquinas virtuales ou físicas:

RAM \leq 2048MB CPU \leq 2 PAE/NX habilitado

Rede: 192.168.120.0/24

BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB

Máquina A:

Rede Interna(eth0) e NAT(eth1)

IP/MS: 192.168.120.100/24

Servidor SSH

SO: Debian amd64 xfce instalado

sda: SO instalado

sdb, sdc, sdd, sde: array de discos

Tamaño sd[bcde]: 10GB

User: usuario

Passwd: abc123.

Groups usuario: usuario sudo

usuario \rightarrow sudo su \rightarrow root

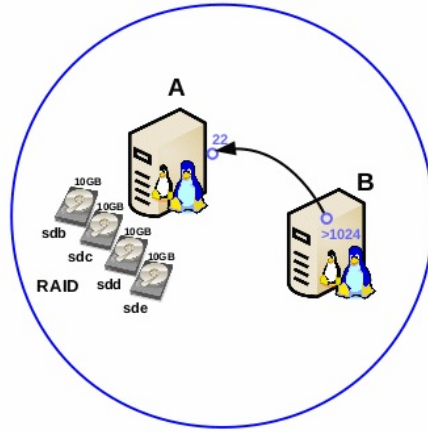
Máquina B:

Rede Interna(eth0)

IP/MS: 192.168.120.101/24

Cliente SSH

SO: Live GNU/Linux



LIMITACIÓN DE RESPONSABILIDADE O autor do presente documento declina calquera responsabilidade asociada ao uso incorrecto e/ou malicioso que puidese realizarse coa información exposta no mesmo. Por tanto, non se fai responsable en ningún caso, nin pode ser considerado legalmente responsable en ningún caso, das consecuencias que poidan derivarse da información contida nel ou que esté enlazada dende ou hacia el, incluíndo os posibles erros e información incorrecta existentes, información difamatoria, así como das consecuencias que se poidan derivar sobre a súa aplicación en sistemas de información reais e/ou virtuais. Este documento foi xerado para uso didáctico e debe ser empregado en contornas privadas e virtuais controladas co permiso correspondente do administrador desas contornas.

NOTA:

■ Prerrequisitos:

- **RAID por Software: mdadm**
- **Práctica Seguridad Informática - Allow Boot GRUB disco duro - GNU/Linux**

■ Documentación de interese:

- **RAID por Software (debian-handbook)**
- Paquete mdadm (# apt update && apt -y install mdadm).
\$ man mdadm
\$ man mdadm.conf (/etc/mdadm/mdadm.conf)
\$ man md

md - Controlador de dispositivo múltiple tamén coñecido como RAID de software Linux:

- Proporciona dispositivos virtuais creados como:

`/dev/mdn`
`/dev/md/n`
`/dev/md/nome`

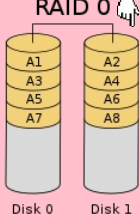
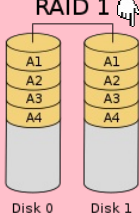
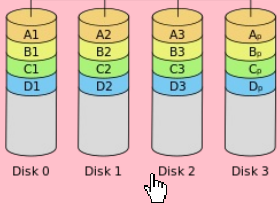
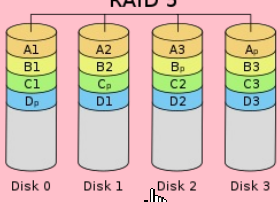
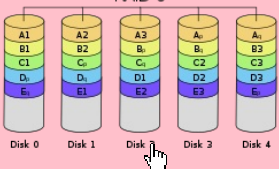
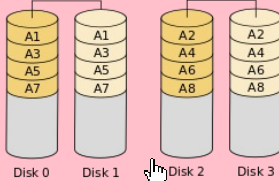
- Admite os niveis: RAID 0, 1, 4, 5, 6 e 10.

- Se non é quen de identificar o dispositivo array creado comezará a dar os seguintes nomes para os arrays de disco: `/dev/md127`, `/dev/md126`, `/dev/md125`... Podemos solucionar isto identificando no ficheiro `/etc/fstab` o dispositivo array por UUID e non por disco/partición `/dev/sdXY`. Unha vez realizado o cambio no `/etc/fstab`, hai que actualizar o arquivo `initrd` (# update-initramfs -u) para que o cambio sexa efectivo dende o arranque do sistema operativo.

```
$ cat /proc/mdstat
$ cat /etc/fstab
$ man update-initramfs
```

- **Comandos GNU/Linux e SHELL BASH (/bin/bash) (1)**
- **losetup (dispositivos de bloques virtual)**
- **LVM: physical volume(pvX), volume group (vgX), logical volume(lvX)**
- **Cheat Sheet Vagrant**

Niveis RAID:

Array	Espazo útil	Redundancia	Tolerancia a fallos (Datos disponibles)
RAID0	$n \geq 2, \sum_{i=1}^n sd_n$ <p>Mínimo 2 discos</p>	<p>NON</p> <p>RAID 0</p>  <p>Disk 0 Disk 1</p>	Non dispoñibles se falla 1 disco
RAID1	$\sum_{i=0}^n \frac{sd_{2n+1} + sd_{2n+2}}{2}$ <p>Mínimo 2 discos</p>	<p>SI (Mirror)</p> <p>RAID 1</p>  <p>Disk 0 Disk 1</p>	Datos dispoñibles se fallan n-1 discos
RAID4	$n \geq 3, \sum_{i=1}^n (sd_n) - 1$ <p>Mínimo 3 discos</p>	<p>SI (1 disco para paridade)</p> <p>RAID 4</p>  <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	Datos dispoñibles se falla 1 disco
RAID5	$n \geq 3, \sum_{i=1}^n (sd_n) - 1$ <p>Mínimo 3 discos</p>	<p>SI (redundancia de paridade repartida en todos os discos)</p> <p>RAID 5</p>  <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	Datos dispoñibles se falla 1 disco
RAID6	$n \geq 4, \sum_{i=1}^n (sd_n) - 2$ <p>Mínimo 4 discos</p>	<p>SI (doble redundancia de paridade repartida en todos os discos)</p> <p>RAID 6</p>  <p>Disk 0 Disk 1 Disk 2 Disk 3 Disk 4</p>	Datos dispoñibles se fallan 2 discos
RAID-1+0 (RAID10)	$\sum_{i=0}^n \left(\frac{(sd_{4n+1} + sd_{4n+2})}{2} + \frac{(sd_{4n+3} + sd_{4n+4})}{2} \right)$ <p>Mínimo 4 discos</p>	<p>SI (Mirror)</p> <p>RAID 1+0</p> <p>RAID 0</p>  <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	Datos dispoñibles se falla 1 disco en cada par de discos RAID1

Resumo Práticas Exemplos

RAID0

- No **Exemplo1. Criar RAID 0** imos crear un array de discos RAID0 con 4 discos: sdb, sdc, sdd e sde.
- No **Exemplo2. Degradar RAID 0 e Recuperar** imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID0 con 4 discos: sdb, sdc, sdd e sde.
- No **Exemplo3. Eliminar e destruír o RAID 0** imos eliminar e destruír o RAID0 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

RAID1

- No **Exemplo4. Criar RAID 1** imos crear un array de discos RAID1 con 3 discos: 2 discos en espello (sdb e sdc) e un disco libre de respaldo (sdd).
- No **Exemplo5. Degradar RAID 1 e Recuperar** imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID1 con 3 discos: 2 discos en espello (sdb e sdc) e un disco libre de respaldo (sdd).
- No **Exemplo6. Eliminar e destruír o RAID 1** imos eliminar e destruír o RAID1 para poder voltar a empregar os 3 discos SATA: sdb, sdc e sdd.

RAID5

- No **Exemplo7. Criar RAID 5** imos crear un array de discos RAID5 con 4 discos: 3 discos RAID5 (sdb, sdc e sdd) + 1 disco de respaldo (sde).
- No **Exemplo8. Degradar RAID 5 e Recuperar** imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID5 con 4 discos: 3 discos RAID5 (sdb, sdc e sdd) + 1 disco de respaldo (sde).
- No **Exemplo9. Eliminar e destruír o RAID 5** imos eliminar e destruír o RAID5 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

RAID-1+0 (RAID10)

- No **Exemplo10. Criar RAID-1+0** imos crear un array de discos RAID10 con 4 discos: 2 discos RAID1 (sdb e sdc) + 2 discos RAID1 (sdd e sde) + 1 volume RAID0 de 4 discos (sdb, sdc, sdd e sde). Así, imos empregar os discos liberados: sdb, sdc, sdd e sde.
- No **Exemplo11. Degradar RAID-1+0 e Recuperar** imos ver que acontece e como recuperar cando se degrada 1 dos discos dentro do array de discos RAID1 (sdb).
- No **Exemplo12. Eliminar e destruír o RAID-1+0** imos eliminar e destruír o RAID10 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

Máquina virtual A: Debian amd64

1. Na contorna gráfica abrir un terminal e executar:

```
usuario@debian:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
usuario@debian:~$ passwd usuario #Cambiar o contrasinal do usuario usuario. Por como contrasinal abc123. (Ollo que o contrasinal ten un caracter punto final).
```

2. Cambiar hostname da máquina virtual A. Por debianA como hostname:

```
usuario@debian:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debian:~# echo 'debianA' > /etc/hostname #Indicar ao sistema o valor do hostname.
```

```
root@debian:~# echo 'kernel.hostname=debianA' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
```

```
root@debian:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
```

```
root@debian:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de usuario.
```

```
usuario@debian:~$ exit #Pechar o terminal saíndo da consola local do usuario usuario.
```

3. Configurar a rede:

Na contorna gráfica abrir un terminal e executar:

```
usuario@debianA:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
```

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# /etc/init.d/avahi-daemon stop || systemctl stop avahi-daemon #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
```

```
root@debianA:~# systemctl disable avahi-daemon #Impide que o servizo avahi-daemon sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d)
```

```
root@debianA:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar doutro xeito (co comando ip(ifconfig) de forma manual ou mediante networking (ficheiros /etc/init.d/networking, /etc/init.d/networking.d) a configuración de rede e non ter conflito con este xestor.
```

```
root@debianA:~# systemctl disable network-manager || systemctl disable NetworkManager #Impide que o servizo network-manager sexa iniciado no arranque xerando os links K* nos runlevels (/etc/rcX.d) ou que sexa arrancado mediante systemd
```

```
root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).
```

```
$ man interfaces #Ver ás páxinas de manual referente ao ficheiro de configuración de rede /etc/network/interfaces
$ cat /etc/network/interfaces #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces
$ ls -l /etc/network/interfaces.d #Listar de forma extendida o contido do directorio /etc/network/interfaces/setup
$ cat /etc/network/interfaces.d/setup #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces/setup
```

```
root@debianA:~# cat > /etc/network/interfaces.d/setup <<EOF #Comezo do ficheiro a crear /etc/network/interfaces.d/setup
auto lo
```

```
iface lo inet loopback
```

```
auto enp0s3
```

```
iface enp0s3 inet static
```

```
address 192.168.120.100/24
```

```
auto enp0s8
```

```
iface enp0s8 inet dhcp
```

```
EOF #Fin do ficheiro a crear /etc/network/interfaces.d/setup
```

```
root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# /etc/init.d/networking start #Arrancar o demo networking, é dicir, activar a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# /etc/init.d/networking status #Comprobar o estado do demo networking, é dicir, comprobar se está activa a configuración de rede en /etc/network/interfaces (/etc/network/interfaces.d).
```

```
root@debianA:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), interna(enp0s3) e NAT(enp0s8).
```

```
root@debianA:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local enp0s3
```

4. Comprobar estado do Servidor SSH:

```
# apt update #Actualizar o listado de paquetes dos repositorios (/etc/apt/sources.list, /etc/apt/sources.list.d)
# apt -y install netcat #Instalar o paquete netcat, é dicir, instalar o paquete que integra o comando nc. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
# dpkg -l net-tools ; [ $(echo $?)-eq '1' ] && apt update && apt -y install net-tools #Verificar se o paquete net-tools está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase. O paquete net-tools é necesario para poder empregar comandos coma: ifconfig, netstat, route e arp.
# dpkg -l openssh-server ; [ $(echo $?)-eq '1' ] && apt update && apt -y install openssh-server #Verificar se o paquete openssh-server está instalado. Se non está instalado, actualízase a lista de paquetes dos repositorios e instálase.
```

```
root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, por defecto non está arrancado.
```

```
root@debianA:~# nc -vz localhost 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
```

root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# netstat -natp | grep 22 #Mediante o comando netstat comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# ss -natp | grep 22 #Mediante o comando ss comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -n permite non resolver nomes amosando así soamente as IPs e o comando ser máis rápido na execución. A opción -a equivale á opción all o que permite amosar todos os sockets (conectores) á escoita no servidor. A opción -t equivale a tcp o que permite buscar soamente información sobre o protocolo TCP. A opción -p equivale a program e amosa o PID e nome do programa ao cal pertence o socket.

root@debianA:~# /etc/init.d/ssh start #Arrancar o servidor SSH.

root@debianA:~# /etc/init.d/ssh status #Comprobar o estado do servidor SSH, agora debe estar arrancado.

root@debianA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafolios /etc/rc*

root@debianA:~# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)

root@debianA:~# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafolios /etc/rc*

root@debianA:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled

root@debianA:~# nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar se o porto 22 do servidor ssh está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.

root@debianA:~# ssh -v usuario@localhost #Comprobar se o servidor SSH está activo e podemos conectarnos a el dende localhost co usuario **usuario** e o seu contrasinal. Se é a primeira vez que nos conectamos o servidor avísanos se estamos de acordo coa autenticación. Respostamos yes e pulsamos Enter. A opción -v (modo verbose) aporta información máis detallada da conexión.

usuario@debianA:~\$ exit #Saír da consola remota ssh a que acabamos de acceder, para voltar á consola local de **root**.

root@debianA:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de **kali**.

usuario@debianA:~\$

Máquina virtual B: Kali amd64

5. Configuración da rede. Na contorna gráfica abrir un terminal e executar:

```
kali@kali:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
kali@kali:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kali:~# /etc/init.d/avahi-daemon stop #Parar o demo avahi-daemon(control resolución de nomes) para poder configurar de forma manual a configuración de rede e non ter conflito con este demo.
root@kali:~# /etc/init.d/network-manager stop || pkill NetworkManager #Parar o demo network-manager(xestor de rede) ou o script NetworkManager (executado sen ser demo) para poder configurar de forma manual a configuración de rede e non ter conflito con este xestor.
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
root@kali:~# ip addr add 192.168.120.101/24 dev eth0 #Configurar a tarxeta de rede interna eth0, coa IP: 192.168.120.101 e máscara de subrede: 255.255.255.0.
root@kali:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina B as tarxetas de redes: loopback(lo) e interna(eth0).
root@kali:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
root@kali:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
root@kali:~# echo '192.168.120.100 debianA' >> /etc/hosts #Engadir no ficheiro /etc/hosts, é dicir, na táboa estática de búsqueda para nomes de host (DNS) o nome debianA, para que atenda á IP 192.168.120.100
root@kali:~# ping -c4 debianA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

6. Cambiar hostname da máquina virtual B. Por kaliB como hostname:

```
root@kali:~# echo 'kaliB' > /etc/hostname #Indicar ao sistema o valor do hostname.
root@kali:~# echo 'kernel.hostname=kaliB' >> /etc/sysctl.conf #Indicar ao kernel o valor do hostname.
root@kali:~# sysctl -p #Activar o cambio de hostname sen ter que pechar sesión nin reiniciar
root@kali:~# exit #Saír da consola local sudo na que estabamos a traballar para voltar á consola local de kali.
kali@kali:~$ exit #Pechar o terminal saíndo da consola local do usuario kali.
```

7. ^{SSH} **B → A** Acceder mediante SSH dende a máquina virtual B á máquina virtual A. Dende agora executaremos sempre os comandos dende a máquina virtual B, a través da consola SSH:

Na contorna gráfica abrir un terminal e executar:

```
kali@kaliB:~$ setxkbmap es #Cambiar o mapa de teclado ao idioma español.
kali@kaliB:~$ nc -vz 192.168.120.100 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
kali@kaliB:~$ nc -vz debianA 22 #Mediante o comando nc(netcat) comprobar que o porto 22 do servidor SSH está en estado escoita(listen), esperando conexións. A opción -v corresponde á opción verbose, o que permite amosar información máis detallada na saída do comando. A opción -z permite devolver PROMPT do sistema e de igual xeito facer o escaneo ao/s porto/s solicitados. O número 22 é o porto TCP a escanear.
kali@kaliB:~$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el. Agora accedemos como o usuario usuario a través da conexión cifrada SSH.
usuario@debianA:~$
```


Xestionar arrays: Configurar no servidor

8. Preparación discos duros: Particionamento/formateo

Temos que agregar na máquina virtual A (debianA) 4 discos duros virtuais SATA de 10GB cada un: sdb, sdc, sdd e sde. Unha vez agregados:

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# apt update || apt-get update #Actualizar repositorios declarados no ficheiro /etc/apt/sources.list e nos ficheiros existentes no directorio /etc/apt/sources.list.d. Así, unha vez realizada a consulta dos ficheiros existentes nas rutas anteriores, descárganse uns ficheiros coas listas de paquetes posibles a instalar. Estes ficheiros son gardados en /var/lib/apt/lists
```

```
root@debianA:~# apt -y install parted || apt-get -y install parted #Instalar o paquete parted. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

```
root@debianA:~# for i in sdb sdc sdd sde; do fdisk -l /dev/$i;done #Amosar a táboa de particións dos 4 discos agregados
```

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
Disco /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
Disco /dev/sdd: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
Disco /dev/sde: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
```

```
root@debianA:~# for i in sdb sdc sdd sde; do parted --script /dev/${i} mklabel msdos;done #Crear a etiqueta de disco (táboa de particións) aos dispositivos /dev/sdb, /dev/sdc, /dev/sdd, /dev/sde sen ter que acceder ao prompt de parted
```

```
root@debianA:~# for i in sdb sdc sdd sde; do parted --script /dev/${i} mkpart primary 0 50% -a cylinder;done #Crear unha partición primaria en cada disco cos primeiros 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
root@debianA:~# for i in sdb sdc sdd sde; do parted --script /dev/${i} mkpart primary 50% 70% -a cylinder;done #Crear unha partición primaria en cada disco de 2GB a continuación das particións de 5GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
root@debianA:~# for i in sdb sdc sdd sde; do parted --script /dev/${i} mkpart primary 70% 100% -a cylinder;done #Crear unha partición primaria en cada disco de 3GB a continuación das particións de 2GB, alineando a cilindros, sen ter que acceder ao prompt de parted
```

```
root@debianA:~# for i in sdb sdc sdd sde; do fdisk -l /dev/$i;done #Amosar a táboa de particións dos 4 discos agregados
```

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0xc0f1200a
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		63	10490444	10490382	5G	83	Linux
/dev/sdb2		10490445	14683409	4192965	2G	83	Linux
/dev/sdb3		14683410	20964824	6281415	3G	83	Linux

```
Disco /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x5df10001
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1		63	10490444	10490382	5G	83	Linux
/dev/sdc2		10490445	14683409	4192965	2G	83	Linux
/dev/sdc3		14683410	20964824	6281415	3G	83	Linux

```
Disco /dev/sdd: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x9e0e7d9d
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdd1		63	10490444	10490382	5G	83	Linux
/dev/sdd2		10490445	14683409	4192965	2G	83	Linux
/dev/sdd3		14683410	20964824	6281415	3G	83	Linux

```
Disco /dev/sde: 10 GiB, 10737418240 bytes, 20971520 sectores
Unidades: sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico/físico): 512 bytes / 512 bytes
Tamaño de E/S (mínimo/óptimo): 512 bytes / 512 bytes
Tipo de etiqueta de disco: dos
Identificador del disco: 0x5df10001
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sde1		63	10490444	10490382	5G	83	Linux
/dev/sde2		10490445	14683409	4192965	2G	83	Linux
/dev/sde3		14683410	20964824	6281415	3G	83	Linux

9. Instalar mdadm

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# apt update || apt-get update #Actualizar repositorios declarados no ficheiro /etc/apt/sources.list e nos ficheiros existentes no directorio /etc/apt/sources.list.d. Así, unha vez realizada a consulta dos ficheiros existentes nas rutas anteriores, descárganse uns ficheiros coas listas de paquetes posibles a instalar. Estes ficheiros son gardados en /var/lib/apt/lists
```

```
root@debianA:~# apt -y install mdadm || apt-get -y install mdadm #Instalar o paquete mdadm. Co parámetro -y automaticamente asumimos yes a calquera pregunta que ocorra na instalación do paquete.
```

```
root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

10. Exemplo1. Crear RAID 0

Imos crear un array de discos RAID0 con 4 discos: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo
(/etc/sudoers, visudo)
root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

root@debianA:~# yes | mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Xerar RAID 0 coas primeiras particións de cada disco

root@debianA:~# mdadm --examine --scan >> /etc/mdadm/mdadm.conf #Volcar información do estado actual do volume RAID e gardar esa información

root@debianA:~# mkdir /mnt/md0 #Xerar o cartafol /mnt/md0

root@debianA:~# mkfs.ext4 -F -L 'RAID0' /dev/md0 #Formatear en ext4 coa etiqueta RAID0 o array /dev/md0

root@debianA:~# lsblk #Listar dispositivos de bloques. Consegue a información do sistema de ficheiros sysfs e a base de datos udev.

root@debianA:~# lsblk -o +UUID #Listar dispositivos de bloques cos seus correspondentes UUID.

root@debianA:~# lsblk -o +UUID | grep md0 #Listar dispositivos de bloques cos seus correspondentes UUID, e filtrar esa saída co patrón md0.

root@debianA:~# UUID_MD0=$(lsblk -o +UUID | grep md0 | awk '{print $NF}' | sort -u) #Declarar unha variable de nome UUID_MD0 co valor do UUID correspondente ao dispositivo /dev/md0

root@debianA:~# echo "UUID=${UUID_MD0} /mnt/md0 ext4 defaults 0 2" >> /etc/fstab #Montar automaticamente os array

root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo

root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [raid0]
md0 : active raid0 sde1[3] sdd1[2] sdc1[1] sdb1[0]
      20959232 blocks super 1.2 512k chunks

unused devices: <none>

root@debianA:~# mdadm --detail /dev/md0 #Amosa información extendida sobre o volume RAID /dev/md0

/dev/md0:
   Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid0
    Array Size : 20959232 (19.99 GiB 21.46 GB)
   Raid Devices : 4
  Total Devices : 4
 Persistence : Superblock is persistent

   Update Time : Sat Jan  2 17:08:24 2021
     State : clean
  Active Devices : 4
 Working Devices : 4
 Failed Devices : 0
  Spare Devices : 0

   Chunk Size : 512K

Consistency Policy : none

           Name : debianA:0 (local to host debianA)
          UUID : 0987486c:c1d78b78:7a2e31ff:f67ecc46
         Events : 0

   Number   Major   Minor   RaidDevice State
     0         8       17         0     active sync   /dev/sdb1
     1         8       33         1     active sync   /dev/sdc1
     2         8       49         2     active sync   /dev/sdd1
     3         8       65         3     active sync   /dev/sde1

root@debianA:~# update-initramfs -u #Actualizar a imaxe initrd

root@debianA:~# reboot #Reiniciar para montar xa no arranque os arrays de disco, os cambios de configuración de rede e a activación no arranque do servizo SSH do servidor
```

Unha vez que reinicie o servidor debianA:

kali@kaliB:~\$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el a través da IP configurada en /etc/network/interfaces.d/setup (192.168.120.100). Agora accedemos como o usuario **usuario** a través da conexión cifrada SSH.

usuario@debianA:~\$ cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

```
Personalities : [raid0] [linear] [multipath] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid0 sdb1[0] sde1[3] sdc1[1] sdd1[2]
      20959232 blocks super 1.2 512k chunks
```

unused devices: <none>

usuario@debianA:~\$ sudo cp -pv /etc/passwd /mnt/md0/ #Copiar o ficheiro /etc/passwd en /mnt/md0 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

IMPORTANTE: Acabamos de comprobar que dende o arranque:

1. A IP de enps03 está configurada como 192.168.120.100
2. Pódese acceder ao servidor SSH
3. Está activo o array de discos RAID0 /dev/md0.

11. Exemplo2. Degradar RAID 0 e Recuperar

Imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID0 con 4 discos: sdb, sdc, sdd e sde. Así, apagamos a máquina virtual A (debianA) e retiramos 1 disco duro virtual SATA de 10GB: sde.

Unha vez retirado o disco sde e iniciado o servidor debianA, teremos un problema no arranque do sistema operativo:

```
You are in emergency mode. After logging in, type "journalctl -xb" to view
system logs, "systemctl reboot" to reboot, "systemctl default" or "exit"
to boot into default mode.

Cannot open access to console, the root account is locked.
See sulogin(8) man page for more details.

Press Enter to continue.
```

Entón, temos que acceder a unha consola de **root** a través do xestor do arranque ou arrancando debianA cunha GNU/Linux Live amd64 para comentar a liña relativa ao RAID0 no ficheiro /etc/fstab e así poder arrancar o sistema operativo e intentar recuperar o array de discos, ou ben, xa na propia consola de root sen chegar a arrancar o sistema operativo por completo intentamos arranxar o array de discos. Entón, imos intentar arranxar o array de discos con esta segunda opción, polo que arrancamos co sistema operativo GNU/Linux instalado no disco duro:

- i. O xestor de arranque: **GRUB versión 2 ou GRUB 2** arranca por defecto na súa primeira opción en 5 segundos. Entón, parar o arranque deste primeira opción premendo as teclas frechas abaixo ↓, arriba ↑.
- ii. Seleccionar a primeira opción de arranque.
- iii. Premer a tecla **e** (edit) para poder editar os parámetros de arranque do kernel.
- iv. Moverse coa tecla frecha abaixo ↓ ata chegar á liña onde aparecen os parámetros **ro quiet splash**
- v. Substituír os parámetros **ro quiet splash** polos parámetros **rw init=/bin/bash**. e premer as teclas **<Ctrl> + x**, é dicir, ^x, para arrancar a opción escollida con novos parámetros do kernel. Agora no arranque veremos que non chegamos a arrancar o sistema operativo porque o primeiro proceso a chamar (init ou systemd) está modificado a /bin/bash, co cal en vez de facer unha chamada ao arranque do sistema operativo facemos unha chamada a unha consola de comandos, polo que, accedemos a unha consola onde temos permisos de root (administrador). **Ollol! Non está cargado completamente o sistema operativo, pero si está recoñecido o hardware.**

```

Versión do GNU GRUB 2.02+dfsg1-20

if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1\
--hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 be4c3037-85b1-4f58\
-a404-54cc95ee1042
else
  search --no-floppy --fs-uuid --set=root be4c3037-85b1-4f58-a40\
4-54cc95ee1042
fi
echo          'Cargando Linux 4.19.0-6-amd64 ...'
linux         /boot/vmlinuz-4.19.0-6-amd64 root=UUID=be4c3037-85b\
1-4f58-a404-54cc95ee1042 rw init=/bin/bash resume=UUID=869fefb2-0fcf-448\
b-a9ea-d2b42f404eb5
echo          'Cargando o ramdisk inicial ...'
initrd        /boot/initrd.img-4.19.0-6-amd64

É compatíbel co modo mínimo de edición en pantalla estilo Emacs.
Completado de listas con TAB. Prema Ctrl-x ou F10 para iniciar,
Ctrl-c ou F2 para a liña de ordes ou ESC para descartar cambios
e volver ao menú do GRUB.
```

vi. Executar:

mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.

cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

```
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : inactive sdc1[1](S) sdd1[2](S) sdb1[0](S)
      15720213 blocks super 1.2

unused devices: <none>
```

mdadm --detail /dev/md0 #Amosa información extendida sobre o volume RAID /dev/md0

```
/dev/md0:
  Version : 1.2
  Raid Level : raid0
```

```

Total Devices : 3
Persistence : Superblock is persistent

State : inactive
Working Devices : 3

Name : debianA:0 (local to host debianA)
UUID : 0987486c:c1d78b78:7a2e31ff:f67ecc46
Events : 0

```

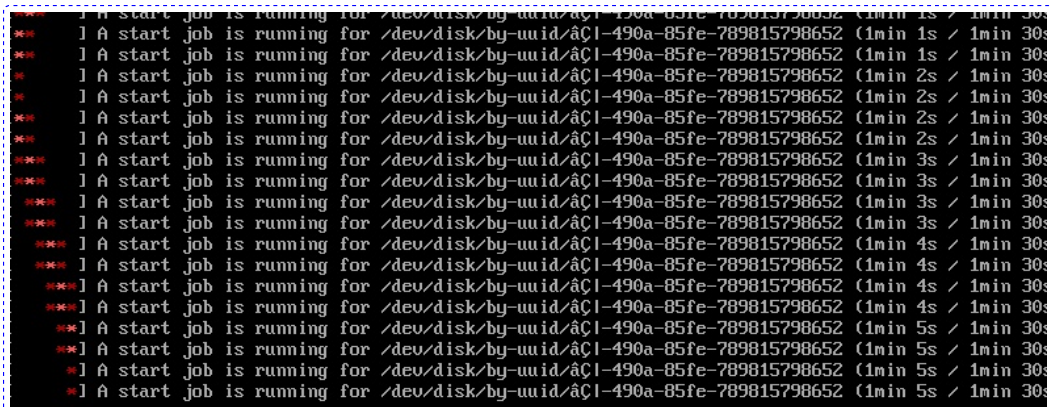
```

Number   Major   Minor   RaidDevice
-         8       49      -          /dev/sdd1
-         8       33      -          /dev/sdc1
-         8       17      -          /dev/sdb1

```

```
# halt -f #Apagar de forma forzosa.
```

- vii. Acabamos de comprobar que o RAID0 está inactivo e non ten presente a partición /dev/sde1. Polo tanto apagamos de novo debianA, insertamos de novo o disco sde e imos ver que acontece.



```

cat /proc/mdstat
md0 : active raid0 sde1[3] sdd1[2] sdc1[1] sdb1[0]
      20959232 blocks super 1.2 512k chunks

unused devices: <none>

```

- viii. Observamos que non podemos recuperar o arranque do sistema operativo, polo que procedemos como antes: editar GRUB e conseguir unha consola de root, na cal executamos os seguintes comandos:

```
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
# mdadm --stop /dev/md0 #Desactivar volume RAID0, liberando todos os recursos.
```

```
md: md0 stopped.
```

```
mdadm: stopped /dev/md0
```

```
# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

```

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>

```

```
# yes | mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Xerar RAID 0 coas primeiras particións de cada disco
```

```
# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

```

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid0 sde1[3] sdd1[2] sdc1[1] sdb1[0]
      20959232 blocks super 1.2 512k chunks

```

```
unused devices: <none>
```

```
# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo
```

```
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
# ls -l /mnt/md0/ #Listar de forma extendida o contido do directorio /mnt/md0/
```

```

total 20
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
-rw-r--r-- 1 root root 2225 Jan 2 16:50 passwd

```

```
# reboot -f #Reiniciar de forma forzosa.
```

ix. **RAID0 recuperado e o sistema operativo arranca correctamente.**

12. Exemplo3. Eliminar e destruír o RAID 0

Imos eliminar e destruír o RAID0 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# umount /mnt/md0 #Desmontar /mnt/md0
```

```
root@debianA:~# mdadm --stop /dev/md0 #Desactivar volume RAID0, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Liberar a asociación de dispositivos ao volume RAID0
```

Se voltamos a xerar o RAID0 cos dispositivos anteriores xa non teriamos os datos gardados nese array de discos posto que o RAID0 (ver niveis RAID) non posúe redundancia e polo tanto non ten tolerancia a fallos:

```
root@kaliA:~# yes | mdadm --create /dev/md0 --level=0 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1  
#Xerar RAID 0 coas primeiras particións de cada disco
```

```
root@kaliA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo
```

```
root@kaliA:~# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
root@kaliA:~# ls -l /mnt/md0/ #Listar de forma extendida o contido do directorio /mnt/md0/  
  
total 0
```

```
root@debianA:~# A=$(grep -n '/mnt/md0' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md0) no ficheiro /etc/fstab e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md0 en /etc/fstab. O número de liña onde aparece /mnt/md0 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/0' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/0) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/mdadm/mdadm.conf #Eliminar a liña correspondente a /dev/md/0 en /etc/mdadm/mdadm.conf. O número de liña onde aparece /dev/md/0 está gardado na variable A
```


13. Exemplo4. Crear RAID 1

Realizado o Exemplo3 imos crear un array de discos RAID1 con 3 discos: 2 discos en espello (sdb e sdc) e un disco libre de respaldo (sdd). Así, imos empregar os discos liberados: sdb, sdc e sdd.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>

root@debianA:~# yes | mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb2 /dev/sdc2 --spare-devices=1 /dev/sdd2 #Xerar RAID 1 coas segundas particións dos discos sdb e sdc e un disco de reposto empregando como reposto a súa segunda partición /dev/sdd2

root@debianA:~# mdadm --examine --scan >> /etc/mdadm/mdadm.conf #Volcar información do estado actual do volume RAID e gardar esa información

root@debianA:~# mkdir /mnt/md1 #Xerar o cartafol /mnt/md1

root@debianA:~# mkfs.ext4 -F -L 'RAID1' /dev/md1 #Formatear en ext4 coa etiqueta RAID1 o array /dev/md1

root@debianA:~# lsblk #Listar dispositivos de bloques. Consegue a información do sistema de ficheiros sysfs e a base de datos udev.

root@debianA:~# lsblk -o +UUID #Listar dispositivos de bloques cos seus correspondentes UUID.

root@debianA:~# lsblk -o +UUID | grep md1 #Listar dispositivos de bloques cos seus correspondentes UUID, e filtrar esa saída co patrón md1.

root@debianA:~# UUID_MD1=$(lsblk -o +UUID | grep md1 | awk '{print $NF}' | sort -u) #Declarar unha variable de nome UUID_MD1 co valor do UUID correspondente ao dispositivo /dev/md1

root@debianA:~# echo "UUID=${UUID_MD1} /mnt/md1 ext4 defaults 0 2" >> /etc/fstab #Montar automáticamente os array

root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo

root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdd2[2](S) sdc2[1] sdb2[0]
      2094400 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

```
root@debianA:~# mdadm --detail /dev/md1 #Amosa información extendida sobre o volume RAID /dev/md1
```

```
/dev/md1:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid1
    Array Size : 2094400 (2045.31 MiB 2144.67 MB)
  Used Dev Size : 2094400 (2045.31 MiB 2144.67 MB)
    Raid Devices : 2
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean
  Active Devices : 2
 Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1


Consistency Policy : resync

    Name : debianA:1 (local to host debianA)
   UUID : 0b5d0644:8cd23c31:5ad1a9ee:aefc0264
   Events : 17

   Number   Major   Minor   RaidDevice State
     0         8       18         0   active sync   /dev/sdb2
     1         8       34         1   active sync   /dev/sdc2

     2         8       50         -    spare   /dev/sdd2
```

```
root@debianA:~# update-initramfs -u #Actualizar a imaxe initrd
```

```
root@debianA:~# reboot #Reiniciar para montar xa no arranque os arrays de disco.
```

Unha vez que reinicie o servidor debianA:

kali@kaliB:~\$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el a través da IP configurada en /etc/network/interfaces.d/setup (192.168.120.100). Agora accedemos como o usuario **usuario** a través da conexión cifrada SSH.

```
usuario@debianA:~$ cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

```
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdd2[2](S) sdc2[1] sdb2[0]
      2094400 blocks super 1.2 [2/2] [UU]
```

```
unused devices: <none>
```

```
usuario@debianA:~$ sudo cp -pv /etc/group /mnt/md1/ #Copiar o ficheiro /etc/group en /mnt/md1 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

IMPORTANTE: Acabamos de comprobar que dende o arranque:

1. A IP de enps03 está configurada como 192.168.120.100
2. Pódese acceder ao servidor SSH
3. Está activo o array de discos RAID1 /dev/md1.

14. Exemplo5. Degradar RAID 1 e Recuperar

Imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID1 con 3 discos: 2 discos en espello (sdb e sdc) e un disco libre de respaldo (sdd).

Entón, imos provocar o fallo de /dev/sdb2, de tal xeito que como temos de respaldo /dev/sdd2 automaticamente este toma o lugar de /dev/sdb2 e o array de discos RAID1 /dev/md1 segue funcionando:

i. Executar:

```
# mdadm --fail /dev/md1 /dev/sdb2 #Amosa información extendida sobre o volume RAID /dev/md1

mdadm: set /dev/sdb2 faulty in /dev/md1

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID1 xa que
entra como activo no array /dev/sdd2

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdd2[2] sdc2[1] sdb2[0](F)
      2094400 blocks super 1.2 [2/1] [_U]
      [=====>.....]  recovery = 57.7% (1209152/2094400) finish=0.0min speed=201525K/sec

unused devices: <none>

# mdadm --detail /dev/md1 #Amosa información extendida sobre o volume RAID /dev/md1

/dev/md1:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid1
    Array Size : 2094400 (2045.31 MiB 2144.67 MB)
  Used Dev Size : 2094400 (2045.31 MiB 2144.67 MB)
    Raid Devices : 2
    Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 1
  Spare Devices : 0

Consistency Policy : resync

        Name : debianA:1 (local to host debianA)
        UUID : 0b5d0644:8cd23c31:5ad1a9ee:ae0264
        Events : 36

   Number   Major   Minor   RaidDevice State
     2         8        50           0  active sync  /dev/sdd2
     1         8        34           1  active sync  /dev/sdc2

     0         8        18           -    faulty    /dev/sdb2
```

Como podemos observar agora /dev/sdb2 está fallando no array RAID1, estando este agora formado por /dev/sdd2 e /dev/sdc2

```
# mdadm --remove /dev/md1 /dev/sdb2 #Quitar o dispositivo fallido /dev/sdb2 do RAID

mdadm: hot removed /dev/sdb2 from /dev/md1

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso amosa soamente 2 discos no array xa
que acabamos de quitar 1 (/dev/sdb2)

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdc2[1] sdd2[2]
      2094400 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

```
# mdadm --detail /dev/md1 #Amosa información extendida sobre o volume RAID /dev/md1
```

```
/dev/md1:
```

```
Version : 1.2
Creation Time : Sat Jan  2 17:08:24 2021
Raid Level : raid1
Array Size : 2094400 (2045.31 MiB 2144.67 MB)
Used Dev Size : 2094400 (2045.31 MiB 2144.67 MB)
Raid Devices : 2
Total Devices : 2
Persistence : Superblock is persistent

Update Time : Sat Jan  2 17:08:24 2021
State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0
```

```
Consistency Policy : resync
```

```
Name : debianA:1 (local to host debianA)
UUID : 0b5d0644:8cd23c31:5ad1a9ee:aefc0264
Events : 39
```

Number	Major	Minor	RaidDevice	State	
2	8	50	0	active sync	/dev/sdd2
1	8	34	1	active sync	/dev/sdc2

Como podemos observar agora /dev/sdb2 xa non forma parte do array RAID1, estando este agora formado por /dev/sdd2 e /dev/sdc2

```
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
# ls -l /mnt/md1/ #Listar de forma extendida o contido do directorio /mnt/md1/
```

```
total 20
-rw-r--r-- 1 root root  980 Jan 2 16:50 group
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
```

- ii. **RAID1 segue funcionando aínda que falle 1 dos discos que forma parte do array. E ademais como actúa o disco de respaldo este sincroniza co array e o RAID1 é recuperado de forma automática.**

15. Exemplo6. Eliminar e destruír o RAID 1

Imos eliminar e destruír o RAID1 para poder voltar a empregar os 3 discos SATA: sdb, sdc e sdd.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# umount /mnt/md1 #Desmontar /mnt/md1
```

```
root@debianA:~# mdadm --stop /dev/md1 #Desactivar volume RAID1, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb2 /dev/sdc2 /dev/sdd2 #Liberar a asociación de dispositivos ao volume RAID1
```

```
root@debianA:~# A=$(grep -n '/mnt/md1' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md1) no ficheiro /etc/fstab e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md1 en /etc/fstab. O número de liña onde aparece /mnt/md1 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/1' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/1) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A},$(( ${A} + 1 ))d" /etc/mdadm/mdadm.conf #Eliminar as liñas correspondentes a /dev/md/1 en /etc/mdadm/mdadm.conf. O número de liña onde aparece /dev/md/1 está gardado na variable A
```

```
root@debianA:~# reboot #Reiniciar para comprobar que no próximo arranque xa non existen referencias aos arrays de disco.
```

Se voltamos a xerar o RAID1 cos dispositivos anteriores segueriamos tendo os datos gardados nese array de discos aínda que empregamos o comando **--zero-superblock** e eliminamos o gardado do referente ao array nos ficheiros `/etc/fstab` e `/etc/mdadm/mdadm.conf`:

```
root@kaliA:~# yes | mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb2 /dev/sdc2 --spare-devices=1 /dev/sdd2 #Xerar RAID 1 coas segundas particións dos discos sdb e sdc e un disco de reposto empregando como reposto a súa segunda partición /dev/sdd2
```

```
root@kaliA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo. Pero podemos observar que non se monta /dev/md1 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado. Podemos observar que non se monta /dev/md1 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# mount /dev/md1 /mnt/md1 #Montar o array de discos RAID1 /mnt/md1 no directorio /mnt/md1. Podemos observar que non se monta /dev/md1 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# ls -l /mnt/md1/ #Listar de forma extendida o contido do directorio /mnt/md1/
```

```
total 20
-rw-r--r-- 1 root root 980 Jan 2 16:50 group
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
```

Entón, para poder destruír este array completamente temos que voltar a formatealo.

```
root@debianA:~# umount /mnt/md1 #Desmontar /mnt/md1
```

```
root@debianA:~# mkfs.ext4 -F -L 'RAID1' /dev/md1 #Formatear en ext4 coa etiqueta RAID1 o array /dev/md1
```

```
root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo. Pero podemos observar que non se monta /dev/md1 xa que agora o UUID do array de discos /dev/md1 non é o mesmo.
```

```
mount: /mnt/md1: can't find UUID=75a85210-5599-4212-b124-80e0ca738ac0.
```

```
root@debianA:~# mount /dev/md1 /mnt/md1 #Montar o array de discos RAID1 /mnt/md1 no directorio /mnt/md1.
```

```
root@debianA:~# ls -l /mnt/md1/ #Listar de forma extendida o contido do directorio /mnt/md1/
```

```
total 20
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
```

16. Exemplo7. Crear RAID 5

Realizado o Exemplo6 imos crear un array de discos RAID5 con 4 discos: 3 discos RAID5 (sdb, sdc e sdd) + 1 disco de respaldo (sde). Así, imos empregar os discos liberados: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo
(/etc/sudoers, visudo)
root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>

root@debianA:~# yes | mdadm --create /dev/md5 --level=5 --raid-devices=3 /dev/sdb3 /dev/sdc3 /dev/sdd3 --
spare-devices=1 /dev/sde3 #Xerar RAID 5 coas últimas particións dos discos sdb, sdc e sdd e un disco de reposto empregando como
reposto a súa terceira partición /dev/sde3

root@debianA:~# mdadm --examine --scan >> /etc/mdadm/mdadm.conf #Volcar información do estado actual do
volume RAID e gardar esa información

root@debianA:~# mkdir /mnt/md5 #Xerar o cartafol /mnt/md5

root@debianA:~# mkfs.ext4 -F -L 'RAID5' /dev/md5 #Formatear en ext4 coa etiqueta RAID5 o array /dev/md5

root@debianA:~# lsblk #Listar dispositivos de bloques. Consegue a información do sistema de ficheiros sysfs e a base de datos udev.

root@debianA:~# lsblk -o +UUID #Listar dispositivos de bloques cos seus correspondentes UUID.

root@debianA:~# lsblk -o +UUID | grep md5 #Listar dispositivos de bloques cos seus correspondentes UUID, e filtrar esa
saída co patrón md5.

root@debianA:~# UUID_MD5=$(lsblk -o +UUID | grep md5 | awk '{print $NF}' | sort -u) #Declarar unha variable
de nome UUID_MD5 co valor do UUID correspondente ao dispositivo /dev/md5

root@debianA:~# echo "UUID=${UUID_MD5} /mnt/md5 ext4 defaults 0 2" >> /etc/fstab #Montar
automáticamente os array

root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo

root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md5 : active raid5 sdd3[4] sde3[3](S) sdc3[1] sdb3[0]
      6275072 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

```
root@debianA:~# mdadm --detail /dev/md5 #Amosa información extendida sobre o volume RAID /dev/md5
```

```
/dev/md5:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid5
    Array Size : 6275072 (5.98 GiB 6.43 GB)
  Used Dev Size : 3137536 (2.99 GiB 3.21 GB)
    Raid Devices : 3
  Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean
  Active Devices : 3
 Working Devices : 4
 Failed Devices : 0
  Spare Devices : 1


    Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

        Name : debianA:5 (local to host debianA)
        UUID : c01f4a3b:f9b45d17:1d29d28a:6cf64a11
        Events : 18

   Number   Major   Minor   RaidDevice State
     0         8       19         0   active sync   /dev/sdb3
     1         8       35         1   active sync   /dev/sdc3
     4         8       51         2   active sync   /dev/sdd3

     3         8       67         -    spare    /dev/sde3
```

```
root@debianA:~# update-initramfs -u #Actualizar a imaxe initrd
```

```
root@debianA:~# reboot #Reiniciar para montar xa no arranque os arrays de disco.
```

Unha vez que reinicie o servidor debianA:

```
kali@kaliB:~$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el a través da IP configurada en /etc/network/interfaces.d/setup (192.168.120.100). Agora accedemos como o usuario usuario a través da conexión cifrada SSH.
```

```
usuario@debianA:~$ cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

```
Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md5 : active raid5 sdb3[0] sdd3[4] sdc3[1] sde3[3](S)
      6275072 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
```

```
unused devices: <none>
```

```
usuario@debianA:~$ sudo cp -pv /etc/shadow /mnt/md5/ #Copiar o ficheiro /etc/shadow en /mnt/md5 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

IMPORTANTE: Acabamos de comprobar que dende o arranque:

1. A IP de enps03 está configurada como 192.168.120.100
2. Pódese acceder ao servidor SSH
3. Está activo o array de discos RAID1 /dev/md5.

17. Exemplo8. Degradar RAID 5 e Recuperar

Imos ver que acontece e como recuperar cando se degrada 1 dos discos do array de discos RAID5 con 4 discos: 3 discos RAID5 (sdb, sdc e sdd) e 1 disco de respaldo (sde).

Entón, imos provocar o fallo de /dev/sdb3, de tal xeito que como temos de respaldo /dev/sde3 automaticamente este toma o lugar de /dev/sdb3 e o array de discos RAID1 /dev/md5 segue funcionando:

i. Executar:

```
# mdadm --fail /dev/md5 /dev/sdb3 #Amosa información extendida sobre o volume RAID /dev/md5

mdadm: set /dev/sdb3 faulty in /dev/md5

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID5 xa que
entra como activo no array /dev/sde3

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md5 : active raid5 sdb3[0](F) sde3[3] sdd3[4] sdc3[1]
      6275072 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [==>.....] recovery = 12.8% (402748/3137536) finish=0.5min speed=80549K/sec

unused devices: <none>

# mdadm --detail /dev/md5 #Amosa información extendida sobre o volume RAID /dev/md5

/dev/md5:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid5
    Array Size : 6275072 (5.98 GiB 6.43 GB)
  Used Dev Size : 3137536 (2.99 GiB 3.21 GB)
    Raid Devices : 3
    Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean, degraded, recovering
  Active Devices : 2
 Working Devices : 3
  Failed Devices : 1
   Spare Devices : 1


    Layout : left-symmetric
   Chunk Size : 512K

Consistency Policy : resync

Rebuild Status : 41% complete

    Name : debianA:5 (local to host debianA)
   UUID : c01f4a3b:f9b45d17:1d29d28a:6cf64a11
  Events : 34

   Number   Major   Minor   RaidDevice State
     3         8        67           0  spare rebuilding   /dev/sde3
     1         8        35           1  active sync        /dev/sdc3
     4         8        51           2  active sync        /dev/sdd3

     0         8        19           -  faulty             /dev/sdb3

    Como podemos observar agora /dev/sdb3 está fallando no array RAID5, estando este agora formado por /dev/sde3,
    /dev/sdd3 e /dev/sdc3

# mdadm --remove /dev/md5 /dev/sdb3 #Quitar o dispositivo fallido /dev/sdb3 do RAID

mdadm: hot removed /dev/sdb3 from /dev/md5

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso amosa soamente 2 discos no array xa
que acabamos de quitar 1 (/dev/sdb3)

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md5 : active raid5 sde3[3] sdd3[4] sdc3[1]
      6275072 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

```
# mdadm --detail /dev/md5 #Amosa información extendida sobre o volume RAID /dev/md5
```

```
/dev/md5:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid5
    Array Size : 6275072 (5.98 GiB 6.43 GB)
  Used Dev Size : 3137536 (2.99 GiB 3.21 GB)
    Raid Devices : 3
    Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean
    Active Devices : 3
    Working Devices : 3
    Failed Devices : 0
    Spare Devices : 0


    Layout : left-symmetric
    Chunk Size : 512K

Consistency Policy : resync

        Name : debianA:5 (local to host debianA)
        UUID : c01f4a3b:f9b45d17:1d29d28a:6cf64a11
        Events : 46

   Number   Major   Minor   RaidDevice State
     3         8       67           0  active sync  /dev/sde3
     1         8       35           1  active sync  /dev/sdc3
     4         8       51           2  active sync  /dev/sdd3
```

Como podemos observar agora /dev/sdb3 xa non forma parte do array RAID5, estando este agora formado por /dev/sde3, /dev/sdd3 e /dev/sdc3

```
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
# ls -l /mnt/md5/ #Listar de forma extendida o contido do directorio /mnt/md5/
```

```
total 20
drwx----- 2 root root 16384 Xan  4 23:18 lost+found
-rw-r----- 1 root shadow 1216 Xan  2 16:50 shadow
```

ii. RAID5 recuperado. O RAID5 segue funcionando grazas ao disco de respaldo.

RAID5 segue funcionando aínda que falle 1 dos discos que forma parte do array. E ademais como actúa o disco de respaldo este sincroniza co array e o RAID5 é recuperado de forma automática.

18. Exemplo9. Eliminar e destruír o RAID 5

Imos eliminar e destruír o RAID5 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# umount /mnt/md5 #Desmontar /mnt/md5
```

```
root@debianA:~# mdadm --stop /dev/md5 #Desactivar volume RAID5, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb3 /dev/sdc3 /dev/sdd3 /dev/sde3 #Liberar a asociación de dispositivos ao volume RAID5
```

```
root@debianA:~# A=$(grep -n '/mnt/md5' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md5) no ficheiro /etc/fstab e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md5 en /etc/fstab. O número de liña onde aparece /mnt/md5 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md5' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md5) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A},$((${A}+1))d" /etc/mdadm/mdadm.conf #Eliminar as liñas correspondentes a /dev/md5 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md5 está gardado na variable A
```

```
root@debianA:~# reboot #Reiniciar para comprobar que no próximo arranque xa non existen referencias aos arrays de disco.
```

Se voltamos a xerar o RAID5 cos dispositivos anteriores seguiríamos tendo os datos gardados nese array de discos aínda que empregamos o comando **--zero-superblock** e eliminamos o gardado do referente ao array nos ficheiros `/etc/fstab` e `/etc/mdadm/mdadm.conf`:

```
root@kaliA:~# yes | mdadm --create /dev/md5 --level=5 --raid-devices=3 /dev/sdb3 /dev/sdc3 /dev/sdd3 --spare-devices=1 /dev/sde3 #Xerar RAID 5 coas últimas particións dos discos sdb, sdc e sdd e un disco de reposto empregando como reposto a súa terceira partición /dev/sde3
```

```
root@kaliA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo. Pero podemos observar que non se monta /dev/md5 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado. Podemos observar que non se monta /dev/md5 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# mount /dev/md5 /mnt/md5 #Montar o array de discos RAID5 /mnt/md5 no directorio /mnt/md5. Podemos observar que non se monta /dev/md5 xa que agora os dispositivos activos do array cambiaron.
```

```
root@kaliA:~# ls -l /mnt/md5/ #Listar de forma extendida o contido do directorio /mnt/md5/
```

Entón, para poder destruír este array completamente temos que voltar a formatealo.

```
root@debianA:~# umount /mnt/md5 #Desmontar /mnt/md5
```

```
root@debianA:~# mkfs.ext4 -F -L 'RAID5' /dev/md5 #Formatear en ext4 coa etiqueta RAID5 o array /dev/md5
```

```
root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo. Pero podemos observar que non se monta /dev/md5 xa que agora o UUID do array de discos /dev/md5 non é o mesmo.
```

```
mount: /mnt/md5: can't find UUID=75a85210-5599-4212-b124-80e0ca738ac0.
```

```
root@debianA:~# mount /dev/md5 /mnt/md5 #Montar o array de discos RAID5 /mnt/md5 no directorio /mnt/md5.
```

```
root@debianA:~# ls -l /mnt/md5/ #Listar de forma extendida o contido do directorio /mnt/md5/
```

```
total 20
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
```

19. Exemplo10. Crear RAID-1+0

Realizado o Exemplo9 imos crear un array de discos RAID10 con 4 discos: 2 discos RAID1 (sdb e sdc) + 2 discos RAID1 (sdd e sde) + 1 volume RAID0 de 4 discos (sdb, sdc, sdd e sde). Así, imos empregar os discos liberados: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>

root@debianA:~# yes | mdadm --create /dev/md10 --level=10 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Xerar RAID 10 coas primeiras particións dos discos sdb, sdc, sdd e sde

root@debianA:~# mdadm --examine --scan >> /etc/mdadm/mdadm.conf #Volcar información do estado actual do volume RAID e gardar esa información

root@debianA:~# mkdir /mnt/md10 #Xerar o cartafol /mnt/md10

root@debianA:~# mkfs.ext4 -F -L 'RAID10' /dev/md10 #Formatear en ext4 coa etiqueta RAID10 o array /dev/md10

root@debianA:~# lsblk #Listar dispositivos de bloques. Consegue a información do sistema de ficheiros sysfs e a base de datos udev.

root@debianA:~# lsblk -o +UUID #Listar dispositivos de bloques cos seus correspondentes UUID.

root@debianA:~# lsblk -o +UUID | grep md10 #Listar dispositivos de bloques cos seus correspondentes UUID, e filtrar esa saída co patrón md10.

root@debianA:~# UUID_MD10=$(lsblk -o +UUID | grep md10 | awk '{print $NF}' | sort -u) #Declarar unha variable de nome UUID_MD10 co valor do UUID correspondente ao dispositivo /dev/md10

root@debianA:~# echo "UUID=${UUID_MD10} /mnt/md10 ext4 defaults 0 2" >> /etc/fstab #Montar automáticamente os array

root@debianA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo

root@debianA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID

Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md10 : active raid10 sde1[3] sdd1[2] sdc1[1] sdb1[0]
      10479616 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]

unused devices: <none>
```

```
root@debianA:~# mdadm --detail /dev/md10 #Amosa información extendida sobre o volume RAID /dev/md10
```

```
/dev/md10:
    Version : 1.2
    Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid10
    Array Size : 10479616 (9.99 GiB 10.73 GB)
    Used Dev Size : 5239808 (5.00 GiB 5.37 GB)
    Raid Devices : 4
    Total Devices : 4
    Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
    State : clean
    Active Devices : 4
    Working Devices : 4
    Failed Devices : 0
    Spare Devices : 0


    Layout : near=2
    Chunk Size : 512K

    Consistency Policy : resync

    Name : debianA:10 (local to host debianA)
    UUID : c0295f50:10a93c58:c37c1c36:bca7297d
    Events : 19

    Number Major Minor RaidDevice State
     0       8      17        0  active sync set-A  /dev/sdb1
     1       8      33        1  active sync set-B  /dev/sdc1
     2       8      49        2  active sync set-A  /dev/sdd1
     3       8      65        3  active sync set-B  /dev/sde1
```

```
root@debianA:~# update-initramfs -u #Actualizar a imaxe initrd
```

```
root@debianA:~# reboot #Reiniciar para montar xa no arranque os arrays de disco.
```

Unha vez que reinicie o servidor debianA:

kali@kaliB:~\$ ssh -v usuario@192.168.120.100 #Comprobar se o servidor SSH está activo e podemos conectarnos a el a través da IP configurada en /etc/network/interfaces.d/setup (192.168.120.100). Agora accedemos como o usuario **usuario** a través da conexión cifrada SSH.

```
usuario@debianA:~$ cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID
```

```
Personalities : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md10 : active raid10 sdd1[2] sdb1[0] sdc1[1] sde1[3]
      10479616 blocks super 1.2 512K chunks 2 near-copies [4/4] [UUUU]
```

```
unused devices: <none>
```

```
usuario@debianA:~$ sudo cp -pv /etc/passwd /etc/group /etc/shadow /mnt/md10/ #Copiar os ficheiros /etc/passwd, /etc/group e /etc/shadow en /mnt/md10 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

IMPORTANTE: Acabamos de comprobar que dende o arranque:

1. A IP de enps03 está configurada como 192.168.120.100
2. Pódese acceder ao servidor SSH
3. Está activo o array de discos RAID10 /dev/md10.

20. Exemplo11. Degradar RAID-1+0 e Recuperar

Imos ver que acontece e como recuperar cando se degrada 1 dos discos dentro do array de discos RAID1 (sdb).

Entón, imos provocar o fallo de /dev/sdb1, de tal xeito quevo array de discos RAID1 dentro de /dev/md10 segue funcionando:

i. Executar:

```
# mdadm --fail /dev/md10 /dev/sdb1 #Amosa información extendida sobre o volume RAID /dev/md10

mdadm: set /dev/sdb1 faulty in /dev/md10

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID10 xa que
fallou o dispositivo /dev/sdb1

Personalities : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md10 : active raid10 sdd1[2] sdb1[0](F) sdc1[1] sde1[3]
      10479616 blocks super 1.2 512K chunks 2 near-copies [4/3] [_UUU]

unused devices: <none>
```

```
# mdadm --detail /dev/md10 #Amosa información extendida sobre o volume RAID /dev/md10
```

```
/dev/md10:
    Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
    Raid Level : raid10
    Array Size : 10479616 (9.99 GiB 10.73 GB)
  Used Dev Size : 5239808 (5.00 GiB 5.37 GB)
    Raid Devices : 4
    Total Devices : 4
 Persistence : Superblock is persistent

    Update Time : Sat Jan  2 17:08:24 2021
      State : clean, degraded
    Active Devices : 3
   Working Devices : 3
    Failed Devices : 1
     Spare Devices : 0

    Layout : near=2
   Chunk Size : 512K

Consistency Policy : resync

        Name : debianA:10 (local to host debianA)
        UUID : c0295f50:10a93c58:c37c1c36:bca7297d
        Events : 23

   Number   Major   Minor   RaidDevice State
    -       -       -       -       -
     0        0        0         0   removed
     1        8       33         1   active sync set-B /dev/sdc1
     2        8       49         2   active sync set-A /dev/sdd1
     3        8       65         3   active sync set-B /dev/sde1

     0        8       17         -   faulty /dev/sdb1
```

Como podemos observar agora /dev/sdb1 está fallando no array RAID10, estando este agora formado por /dev/sde1, /dev/sdd1 e /dev/sdc1

```
# mdadm --remove /dev/md10 /dev/sdb1 #Quitar o dispositivo fallido /dev/sdb1 do RAID
```

```
mdadm: hot removed /dev/sdb1 from /dev/md10
```

```
# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso amosa soamente 2 discos no array xa
que acabamos de quitar 1 (/dev/sdb1)
```

```
Personalities : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md10 : active raid10 sdd1[2] sdc1[1] sde1[3]
      10479616 blocks super 1.2 512K chunks 2 near-copies [4/3] [_UUU]

unused devices: <none>
```

```
# mdadm --detail /dev/md10 #Amosa información extendida sobre o volume RAID /dev/md10
```

```
/dev/md10:
  Version : 1.2
  Creation Time : Sat Jan  2 17:08:24 2021
  Raid Level : raid10
  Array Size : 10479616 (9.99 GiB 10.73 GB)
  Used Dev Size : 5239808 (5.00 GiB 5.37 GB)
  Raid Devices : 4
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Sat Jan  2 17:08:24 2021
  State : clean, degraded
  Active Devices : 3
  Working Devices : 3
  Failed Devices : 0
  Spare Devices : 0


  Layout : near=2
  Chunk Size : 512K

Consistency Policy : resync

        Name : debianA:10 (local to host debianA)
        UUID : c0295f50:10a93c58:c37c1c36:bca7297d
        Events : 24
```

Number	Major	Minor	RaidDevice	State	
-	0	0	0	removed	
1	8	33	1	active sync	set-B /dev/sdc1
2	8	49	2	active sync	set-A /dev/sdd1
3	8	65	3	active sync	set-B /dev/sde1

Como podemos observar agora /dev/sdb1 xa non forma parte do array RAID10, estando este agora formado por /dev/sde1, /dev/sdd1 e /dev/sdc1

```
# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado.
```

```
# ls -l /mnt/md10/ #Listar de forma extendida o contido do directorio /mnt/md10/
```

```
total 28
-rw-r--r-- 1 root root    980 Xan  2 16:50 group
drwx----- 2 root root  16384 Xan  5 02:07 lost+found
-rw-r--r-- 1 root root   2225 Xan  2 16:50 passwd
-rw-r----- 1 root shadow 1216 Xan  2 16:50 shadow
```

ii. **RAID10 segue funcionando aínda que falle 1 dos discos que forma parte do array.**

21. Exemplo12. Eliminar e destruír o RAID-1+0

Imos eliminar e destruír o RAID10 para poder voltar a empregar os 4 discos SATA: sdb, sdc, sdd e sde.

```
usuario@debianA:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
```

```
root@debianA:~# umount /mnt/md10 #Desmontar /mnt/md10
```

```
root@debianA:~# mdadm --stop /dev/md10 #Desactivar volume RAID10, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Liberar a asociación de dispositivos ao volume RAID10
```

```
root@debianA:~# A=$(grep -n '/mnt/md10' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md10) no ficheiro /etc/fstab e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md10 en /etc/fstab. O número de liña onde aparece /mnt/md10 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/10' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/10) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/mdadm/mdadm.conf #Eliminar a liña correspondente a /dev/md/10 en /etc/mdadm/mdadm.conf. O número de liña onde aparece /dev/md/10 está gardado na variable A
```

```
root@debianA:~# reboot #Reiniciar para comprobar que no próximo arranque xa non existen referencias aos arrays de disco.
```

Se voltamos a xerar o RAID10 cos dispositivos anteriores seguiríamos tendo os datos gardados nese array de discos aínda que empregamos o comando **--zero-superblock** e eliminamos o gardado do referente ao array nos ficheiros /etc/fstab e /etc/mdadm/mdadm.conf:

```
root@kaliA:~# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Neste caso non amosa ningunha referencia.
```

```
root@kaliA:~# yes | mdadm --create /dev/md10 --level=10 --raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Xerar RAID 10 coas primeiras particións dos discos sdb, sdc, sdd e sde
```

```
root@kaliA:~# mount -a #Provocar a montaxe dos arrays sen ter que reiniciar o equipo. Pero podemos observar que non se monta /dev/md10 xa que agora non existe referencia en /etc/fstab sobre o dispositivo /dev/md10
```

```
root@kaliA:~# mount #Amosar os sistemas de ficheiros montados, é dicir, os que está a usar e podemos empregar neste sistema operativo instalado. Podemos observar que non se monta /dev/md10
```

```
root@kaliA:~# mount /dev/md10 /mnt/md10 #Montar o array de discos RAID10 /mnt/md10 no directorio /mnt/md10.
```

```
root@kaliA:~# ls -l /mnt/md10/ #Listar de forma extendida o contido do directorio /mnt/md10/
```

```
total 16
-rw-r--r-- 1 root root      0 Xan  2 16:50 group
drwx----- 2 root root 16384 Xan  6 23:00 lost+found
-rw-r--r-- 1 root root      0 Xan  2 16:50 passwd
-rw-r----- 1 root shadow  0 Xan  2 16:50 shadow
```

Entón, para poder destruír este array completamente temos que voltar a formatealo.

```
root@debianA:~# umount /mnt/md10 #Desmontar /mnt/md10
```

```
root@debianA:~# mkfs.ext4 -F -L 'RAID10' /dev/md10 #Formatear en ext4 coa etiqueta RAID10 o array /dev/md10
```

```
root@debianA:~# mount /dev/md10 /mnt/md10 #Montar o array de discos RAID10 /mnt/md10 no directorio /mnt/md10.
```

```
root@debianA:~# ls -l /mnt/md10/ #Listar de forma extendida o contido do directorio /mnt/md10/
```

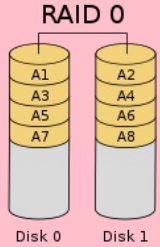
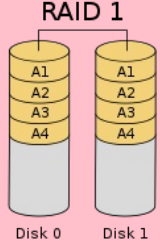
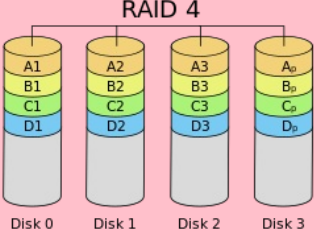
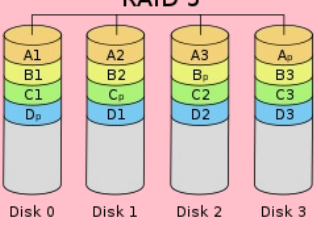
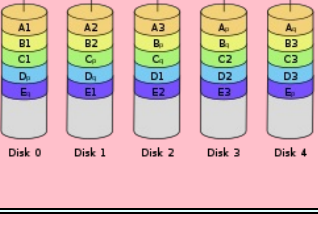
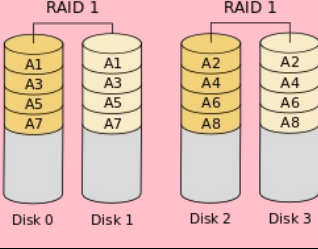
```
total 20
drwx----- 2 root root 16384 Jan 2 20:39 lost+found
```




```
root@debianA:~# umount /mnt/md10 #Desmontar /mnt/md10
```

```
root@debianA:~# mdadm --stop /dev/md10 #Desactivar volume RAID10, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1 #Liberar a asociación de dispositivos ao volume RAID10
```



Imaxe Wikimedia Commons	Licenza
 <p>RAID 0</p> <p>Disk 0 Disk 1</p>	<p>en:User:Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>
 <p>RAID 1</p> <p>Disk 0 Disk 1</p>	<p>en:User:Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>
 <p>RAID 4</p> <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	<p>en:User:Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>
 <p>RAID 5</p> <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	<p>en:User:Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>
 <p>RAID 6</p> <p>Disk 0 Disk 1 Disk 2 Disk 3 Disk 4</p>	<p>en:User:Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>
 <p>RAID 1+0</p> <p>RAID 0</p> <p>RAID 1 RAID 1</p> <p>Disk 0 Disk 1 Disk 2 Disk 3</p>	<p>Wheart, based on image File:RAID 0.svg by Cburnett, CC BY-SA 3.0 <http://creativecommons.org/licenses/by-sa/3.0/>, via Wikimedia Commons</p>

Mathjax	Licenza	
https://www.mathjax.org/ 	https://github.com/mathjax/MathJax/blob/master/LICENSE Apache License 2.0 	

Anexo Vagrant: Creación do Escenario e Configuración Inicial



O escenario da práctica e as configuracións das máquinas xa estarían realizadas no ficheiro **Vagrantfile** seguinte, tendo en conta que:

- No Vagrantfile:
 1. En debianA: eth0(NAT) e eth1(Rede Interna, IP:192.168.120.100)
 2. En kaliB: eth0(NAT) e eth1(Rede Interna, IP:192.168.120.101)
 3. debianA recibe o nome Practica-SI-RAID-debianA en VirtualBox
 4. kaliB recibe o nome Practica-SI-RAID-kaliB en VirtualBox
- Débese ter instalado Vagrant
- Débese posuír conexión a Internet para descargar o ficheiro Vagrantfile
- Débense executar os seguintes comandos. Lembrar que os comandos Vagrant, deben executarse na mesma ruta onde existe o ficheiro Vagrantfile:

```
$ wget https://gist.githubusercontent.com/ricardofc/df16041f117a409f19930720b2eb8048/raw/3b5545d1fbaf57217e8d94eb950839439cc4b208/Vagrantfile-Practica-SI-RAID \
-O Vagrantfile
$ mkdir RAID && cp -pv Vagrantfile RAID && cd RAID
$ vagrant up
```

```
1 Vagrant.configure("2") do |config|
2
3   # Máquina virtual debianA
4   config.vm.define "debianA" do |debian|
5     # Caixa virtual
6     debian.vm.box = "debian/bookworm64"
7
8     # Nome do hostname da máquina virtual
9     debian.vm.hostname = "debianA"
10
11    # Tempo de espera para o arranque da máquina virtual
12    debian.vm.boot_timeout = 1800
13
14    # Fornecedor VirtualBox
15    debian.vm.provider "virtualbox" do |vb|
16
17      # Habilitar interface gráfica do VirtualBox
18      vb.gui = true
19
20      # Definir a memoria RAM da máquina virtual (2048 MB)
21      vb.memory = "2048"
22
23      # Definir o número de CPUs da máquina virtual (2)
24      vb.cpus = 2
25
26      # Nome da máquina virtual no VirtualBox
27      vb.name = "Practica-SI-RAID-debianA"
28
29      # Definir a orde de arranque do BIOS
30      vb.customize ["modifyvm", :id, "--boot1", "dvd", "--boot2", "disk", "--boot3", "none", "--boot4", "none"]
31
32    end
33
34    # Engadir 4 discos duros dinámicos de 10 GB cada un (executar soamente unha vez)
35    (1..4).each do |i|
36      debian.vm.provider "virtualbox" do |vb|
37
38        # Arquivo do disco virtual
39        file_for_disk = "./disk#{i}.vdi"
40
41        # Verificar se o arquivo do disco xa existe
42        unless File.exist?(file_for_disk)
43
44          # Crear o disco duro dinámico
45          vb.customize ["createhd", "--filename", "disk#{i}.vdi", "--size", 10 * 1024]
46
47          # Conectar o disco duro dinámico ao controlador SATA
48          vb.customize ["storageattach", :id, "--storagectl", "SATA Controller", "--port", i, "--device", 0, "--type", "hdd", "--medium", "disk#{i}.vdi"]
49
50        end
51      end
52    end
53
54    # Configurar a rede interna (eth1)
55    debian.vm.network "private_network", ip: "192.168.120.100", virtualbox__intnet: "raid", adapter: 2
56
57    # Aproveisionamento:
58
59    # Instalar o ambiente de desktop XFCE e paquetes necesarios
60    debian.vm.provision "shell", inline: <<-SHELL
61      # Actualizar o sistema
62      apt update
63      apt upgrade -y
64
65      # Instalar o ambiente de desktop XFCE e outros paquetes
66      apt -y install xfce4 xfce4-terminal vim terminator
67
68      # Configurar o XFCE como ambiente de desktop predeterminado
69      echo "startxfce4" >> ~/.xsession
70
71      # Crear o usuario "usuario"
72      apt -y install whois && useradd -m -d /home/usuario -s /bin/bash -p $(mkpasswd abc123.) usuario && usermod -aG sudo usuario
73
74      # Configurar o teclado español
75      echo "setxkbmap es" >> /home/usuario/.bashrc
76
77      # Habilitar o acceso por contrasinal e a root mediante SSH
78      sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd_config
79      sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
80
81      # Reiniciar a máquina virtual
82      reboot
83    SHELL
84  end
```

```
85
86
87 # Máquina virtual KaliB
88 config.vm.define "kaliB" do |kali|
89   kali.vm.box = "kalilinux/rolling"
90   kali.vm.hostname = "kaliB"
91   kali.vm.boot_timeout = 1800
92
93   kali.vm.provider "virtualbox" do |vb|
94     vb.gui = true
95     vb.memory = "2048"
96     vb.cpus = 2
97     vb.name = "Practica-SI-RAID-kaliB"
98   end
99
100   kali.vm.network "private_network", ip: "192.168.120.101", virtualbox____intnet: "raid", adapter: 2
101
102   kali.vm.provision "shell", inline: <<-SHELL
103     apt update
104     apt -y install whois && useradd -m -d /home/kali -s /usr/bin/zsh -p $(mkpasswd kali) kali && usermod -aG kali-trusted kali
105     echo "setxkbmap es" >> /home/kali/.zshrc
106   SHELL
107 end
108 end
```