

Práctica BRS - RAID por Software: mdadm

ESCENARIO (realizado con Vagrant: Vagrantfile → vagrant up)

Máquinas virtuales ou físicas:

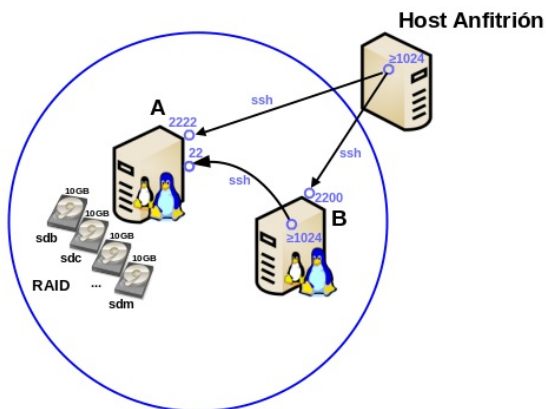
RAM ≥ 2048MB CPU ≥ 2 PAE/NX habilitado
Rede: 192.168.120.0/24
BIOS: Permite arranque dispositivo extraíble: CD/DVD, USB
NAT(eth0) e Rede Interna(eth1)
sda: SO instalado

Máquina A:

IP/MS: 192.168.120.100/24
Servidor SSH
SO: Debian amd64 xfce
sd[b,m]: array de discos
Tamaño sd[bcdefghijklm]: 10GB
User: usuario
Passwd: abc123.
Groups usuario: usuario sudo
usuario → sudo su - → root

Máquina B:

IP/MS: 192.168.120.101/24
Cliente SSH
SO: Kali Live GNU/Linux
User: kali
Passwd: kali
Groups kali: kali sudo
kali → sudo su - → root



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** e **1-Taller-BRS-Practica-Allow-Boot-GRUB-HD-GNU-Linux.pdf**

■ 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

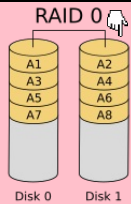
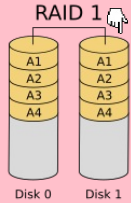
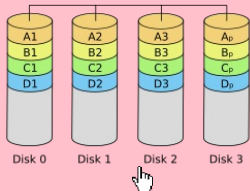
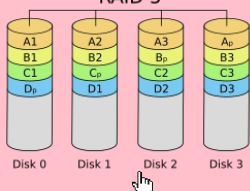
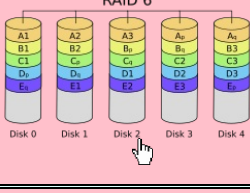
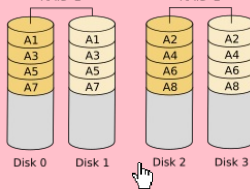
md - Controlador de dispositivo múltiple tamén coñecido como RAID de software Linux:

- Proporciona dispositivos virtuais creados como:
/dev/mdn
/dev/md/n
/dev/md/nome
- Admite os niveis: RAID 0, 1, 4, 5, 6 e 10. Pero tamén podemos xerar outros niveis como RAID50 e RAID60, xerando primeiro os RAID5 e RAID6 e logo combinándoos nun RAID0. Podemos traballar con particións ou discos duros.
- 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 dispoñibles)
RAID0	$n \geq 2, \sum_{i=1}^n sd_n$ <p>Mínimo 2 discos</p>	<p>NON</p>  <p>RAID 0</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>	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> 	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> 	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> 	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> 	Datos dispoñibles se falla 1 disco en cada par de discos RAID1

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-BRS-RAID-debianA en VirtualBox
 4. kaliB recibe o nome Practica-BRS-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/3b604d729991844652401f01f676a9ba/raw/157ef17f84012174e1d12b5b931d74f82304fbfd/Vagrantfile-BRS-RAID -O Vagrantfile
$ mkdir RAID && cp -pv Vagrantfile RAID && cd RAID
$ vagrant up
```
- No caso de problemas sempre se pode voltar a realizar o proceso anterior logo de eliminar a configuración realizada en Vagrant:

```
$ vagrant destroy -f
$ rm -rf .vagrant
```

```
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-BRS-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      # Asignar está máquina virtual ao grupo de máquinas virtuais BRS-RAID
33      vb.customize ["modifyvm", :id, "--groups", "/BRS/RAID"]
34
35    end
36  end
37
38  # Engadir 12 discos duros dinámicos de 10 GB cada un (executar soamente unha vez)
39  (1..12).each do |i|
40    debian.vm.provider "virtualbox" do |vb|
41
42      # Arquivo do disco virtual
43      file_for_disk = ".disk#{i}.vdi"
44
45      # Verificar se o arquivo do disco xa existe
46      unless File.exist?(file_for_disk)
47
48        # Crear o disco duro dinámico
49        vb.customize ["createhd", "--filename", "disk#{i}.vdi", "--size", 10 * 1024]
50
51        # Conectar o disco duro dinámico ao controlador SATA
52        vb.customize ["storageattach", :id, "--storagectl", "SATA Controller", "--port", i, "--device", 0, "--type", "hdd", "--medium", "disk#{i}.vdi"]
53
54      end
55    end
56  end
57
58  # Configurar a rede interna (eth1)
59  debian.vm.network "private_network", ip: "192.168.120.100", virtualbox__intnet: "raid", adapter: 2
60
61  # Aproveitamento:
62
63  # Instalar o ambiente de desktop XFCE e paquetes necesarios
64  debian.vm.provision "shell", inline: <<-SHELL
65  # Actualizar o sistema
66  apt update
67  apt upgrade -y
68
69  # Instalar o ambiente de desktop XFCE e outros paquetes
70  apt -y install xfce4 xfce4-terminal vim terminator
71
72  # Configurar o XFCE como ambiente de desktop predeterminado
73  echo "startxfce4" >> ~/.xsession
74
75  # Crear o usuario "usuario"
76  apt -y install whois && useradd -m -d /home/usuario -s /bin/bash -p $(mkpasswd abc123.) usuario && usermod -aG sudo usuario
77
78  # Configurar o teclado español
79  echo "setxkbmap es" >> /home/usuario/.bashrc
80
81  # Habilitar o acceso por contrasinal e a root mediante SSH
82  sed -i 's/PasswordAuthentication no/PasswordAuthentication yes/' /etc/ssh/sshd_config
83  sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config
84
85  # Reiniciar a máquina virtual
86  reboot
87  SHELL
88 end
89
90 # Máquina virtual KaliB
91 config.vm.define "kaliB" do |kali|
92   kali.vm.box = "kalilinux/rolling"
93   kali.vm.hostname = "kaliB"
94   kali.vm.boot_timeout = 1800
95
96   kali.vm.provider "virtualbox" do |vb|
97     vb.gui = true
98     vb.memory = "2048"
99     vb.cpus = 2
100     vb.name = "Practica-BRS-RAID-kaliB"
101     vb.customize ["modifyvm", :id, "--groups", "/BRS/RAID"]
102   end
103
104   kali.vm.network "private_network", ip: "192.168.120.101", virtualbox__intnet: "raid", adapter: 2
105
106   kali.vm.provision "shell", inline: <<-SHELL
107   apt update
108   apt -y install whois && useradd -m -d /home/kali -s /usr/bin/zsh -p $(mkpasswd kali) kali && usermod -aG kali-trusted kali
109   echo "setxkbmap es" >> /home/kali/.zshrc
110   SHELL
111 end
112 end
113
```

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: sdc, sdd e sde.

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.

RAID6

- No **Exemplo10. Criar RAID 6** imos crear un array de discos RAID6 con 6 discos: 4 discos RAID6 (sdb, sdc, sdd, sde) + 2 discos de respaldo (sdf, sdg).
- No **Exemplo11. Degradar RAID 6 e Recuperar** imos ver que acontece e como recuperar cando se degradan 2 dos discos do array de discos RAID6 con 6 discos: 4 discos RAID6 (sdb, sdc, sdd, sde) + 2 discos de respaldo (sdf, sdg).
- No **Exemplo12. Eliminar e destruír o RAID 6** imos eliminar e destruír o RAID6 para poder voltar a empregar os 6 discos SATA: sdb, sdc, sdd, sde, sdf e sdg.

RAID-1+0 (RAID10)

- No **Exemplo13. 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 **Exemplo14. 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 **Exemplo15. 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.

RAID50

- No **Exemplo16. Criar RAID50** imos crear 2 arrays independentes de discos RAID5 e combinándoos cun RAID0, tal que:
 - md50=RAID0[md51+md52]
 - md51=RAID5[sdb, sdc, sdd, sde], onde sde=spare-device
 - md52=RAID5[sdf, sdg, sdh, sdi], onde sdi=spare-device
- No **Exemplo17. Degradar RAID50 e Recuperar** imos ver que acontece e como recuperar cando se degrada 1 dos discos dentro do array de discos RAID50 (sdb).
- No **Exemplo18. Eliminar e destruír o RAID50** imos eliminar e destruír o RAID50 para poder voltar a empregar os 8 discos SATA: sdb, sdc, sdd, sde, sdf, sdg, sdh e sdi.

RAID60

- No **Exemplo19. Criar RAID60** imos crear 2 arrays independentes de discos RAID6 e combinándoos cun RAID0, tal que:
 - md60=RAID0[md61+md62]
 - md61=RAID6[sdb, sdc, sdd, sde, sdf, sdg], onde sdf e sdg son spare-device
 - md62=RAID6[sdh, sdi, sdj, sdk, sdl, sdm], onde sdl e sdm son spare-device
- No **Exemplo20. Degradar RAID60 e Recuperar** imos ver que acontece e como recuperar cando se degradan 2 dos discos dentro do array de discos RAID60 (sdb e sdc).
- No **Exemplo21. Eliminar e destruír o RAID60** imos eliminar e destruír o RAID60 para poder voltar a empregar os 12 discos SATA: sdb, sdc, sdd, sde, sdf, sdg, sdh, sdi, sdj, sdk, sdl, sdm.

Máquina virtual A: Debian amd64

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

usuario@debianA:~\$ ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), NAT(eth0) e interna(eth1).

```
$ 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

# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
auto eth1
iface eth1 inet static
    address 192.168.120.100
    netmask 255.255.255.0
#VAGRANT-END

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

usuario@debianA:~\$ ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth1

2. 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-traditional #Instalar o paquete netcat-traditional, 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:~# systemctl is-enabled ssh.service #Amosa se o servizo ssh está enabled ou disabled

```
# find /etc/rc* -name "*ssh*" #Busca polas links runlevels nos cartafoles /etc/rc*
# systemctl enable ssh #Permite que o servizo ssh sexa iniciado no arranque xerando os links nos runlevels (/etc/rcX.d)
```

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:~$
```

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

```
kali@kaliB:~$ sudo su - #Acceder á consola de root(administrador) a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)
root@kaliB:~# /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@kaliB:~# /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@kaliB:~# ip addr show #Amosar a configuración de todas as tarxetas de rede. Nesta caso, na máquina A, as tarxetas de redes: loopback(lo), NAT(eth0) e
interna(eth1).
root@kaliB:~# cat /etc/network/interfaces #Amosar o contido do ficheiro configuración de rede /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
#VAGRANT-BEGIN
# The contents below are automatically generated by Vagrant. Do not modify.
auto eth1
iface eth1 inet static
    address 192.168.120.101
    netmask 255.255.255.0
#VAGRANT-END

root@kaliB:~# ping -c4 192.168.120.101 #Comprobar mediante o comando ping a conectividade coa interface de rede local eth0
root@kaliB:~# ping -c4 192.168.120.100 #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
root@kaliB:~# 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@kaliB:~# ping -c4 debianA #Comprobar mediante o comando ping a conectividade coa interface de rede da máquina virtual A
```

SSH

4. **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:~$ 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 usuario@debianA #Agora accedemos a debianA como o usuario usuario a través da conexión cifrada SSH.
usuario@debianA:~$
```

Xestionar arrays: Configurar no servidor (debianA)

5. Preparación discos duros: Particionamento/formateo

A través do vagrant xa temos agregados na máquina virtual A (debianA) 12 discos duros virtuais SATA de 10GB cada un: sdb, sdc, sdd, sde, sdf, sdg, sdh, sdi, sdj, sdk, sdl, sdm. Unha vez agregados:

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

```
# 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
# 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 sd{b..m}; do fdisk -l /dev/\$i;done #Amosar a táboa de particións dos 12 discos agregados

```
Disco /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectores
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disco /dev/sdc: 10 GiB, 10737418240 bytes, 20971520 sectores
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

...

```
Disco /dev/sdm: 10 GiB, 10737418240 bytes, 20971520 sectores
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

6. 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
```


7. 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/sdb /dev/sdc /dev/sdd /dev/sde #Xerar RAID 0 cos discos sdb,
sdc, sdc e sdd

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 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 sde[3] sdd[2] sdc[1] sdb[0]
      41906176 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 : Mon Oct 14 19:59:49 2024
    Raid Level : raid0
   Array Size : 41906176 (39.96 GiB 42.91 GB)
  Raid Devices : 4
 Total Devices : 4
 Persistence : Superblock is persistent

   Update Time : Mon Oct 14 19:59:49 2024
     State : clean
   Active Devices : 4
 Working Devices : 4
 Failed Devices : 0
  Spare Devices : 0

    Layout : -unknown-
   Chunk Size : 512K

Consistency Policy : none

           Name : debianA:0 (local to host debianA)
          UUID : eb88eae0:d3bdd10d:de56227d:4bbaa915
         Events : 0

   Number   Major   Minor   RaidDevice State
    0         8       16         0      active sync /dev/sdb
    1         8       32         1      active sync /dev/sdc
    2         8       48         2      active sync /dev/sdd
    3         8       64         3      active sync /dev/sde

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 usuario@debianA #Accedemos a debianA 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 sde[3] sdc[1] sdb[0] sdd[2]
      41906176 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 está activo o array de discos RAID0 /dev/md0.

8. 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 (disk4.vdi).

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.
Give root password for maintenance
```

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 introducindo (ver imaxe anterior) o contrasinal de root (vagrant):

i. 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 sdc[1](S) sdbb[0](S) sdbh[2](S)
      31429632 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 : eb88eae0:d3bdd10d:de56227d:4bbaa915
  Events : 0

  Number Major Minor RaidDevice
    -     8     32      -     /dev/sdc
    -     8    112      -     /dev/sdh
    -     8     16      -     /dev/sdb
```

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

Acabamos de comprobar que o RAID0 está inactivo e non ten presente o disco /dev/sde. Polo tanto apagamos de novo debianA, insertamos de novo o disco sde(disk4.vdi) e imos ver que acontece.

```
# 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 sdb[1] sdc[0] sdd[2] sde[3]
      41906176 blocks super 1.2 512k chunks
```

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

```
# 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 Oct 14 20:02 lost+found
-rw-r--r-- 1 root root 1758 Oct 14 19:58 passwd
```

ii. RAID0 recuperado e o sistema operativo arranca correctamente.

9. 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/sdb /dev/sdc /dev/sdd /dev/sde #Liberar a asociación de dispositivos ao volume RAID0

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

Se voltamos a xerar o RAID0 cos dispositivos anteriores xa non teríamos 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/sdb /dev/sdc /dev/sdd /dev/sde #Xerar RAID 0 cos discos sdb, sdc, sdc e sdd
```

```
root@kaliA:~# mount /dev/md0 /mnt/md0 #Montar o array de discos RAID0 /dev/md0 no directorio /mnt/md0.
```

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

```
total 0
```

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

10. 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/sdb /dev/sdc --spare-devices=1 /dev/sdd #Xerar RAID 1 cos discos sdb e sdc e un disco de reposto empregando como reposto /dev/sdd
```

```
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 sdd[2](S) sdc[1] sdb[0]
      10476544 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 : Mon Oct 14 21:12:08 2024
  Raid Level : raid1
  Array Size : 10476544 (9.99 GiB 10.73 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 2
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Mon Oct 14 21:15:03 2024
  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 : df1ae9fa:d92d36c1:0dcfe03f:47aa9559
           Events : 19

   Number   Major   Minor   RaidDevice State   /dev/sdb
     0         8       16         0     active sync   /dev/sdb
     1         8       32         1     active sync   /dev/sdc
     2         8       48          -     spare        /dev/sdd
```

```
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 usuario@debianA #Accedemos a debianA 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 sdc[1] sdb[0] sdd[2](S)
      10476544 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 está activo o array de discos RAID1 /dev/md1.

11. 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/sdb, de tal xeito que como temos de respaldo /dev/sdd automaticamente este toma o lugar de /dev/sdb e o array de discos RAID1 /dev/md1 segue funcionando. Agora en vez de quitar o disco dende a interface de VirtualBox imos facelo co propio comando mdadm:

i. Executar:

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

mdadm: set /dev/sdb 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/sdd

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdd[2] sdc[1] sdb[0](F)
      10476544 blocks super 1.2 [2/1] [_U]
      [=====>.....]   recovery = 50.0% (5238272/10476544) 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 : Mon Oct 14 21:12:08 2024
    Raid Level : raid1
    Array Size : 10476544 (9.99 GiB 10.73 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
    Raid Devices : 2
   Total Devices : 3
 Persistence : Superblock is persistent

   Update Time : Mon Oct 14 21:23:55 2024
     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 : df1ae9fa:d92d36c1:0dcfe03f:47aa9559
         Events : 42

   Number   Major   Minor   RaidDevice State
     2         8       48         0     active sync   /dev/sdd
     1         8       32         1     active sync   /dev/sdc
     0         8       16         -     faulty    /dev/sdb
```

Como podemos observar agora /dev/sdb está fallando no array RAID1, estando este agora formado por /dev/sdd e /dev/sdc

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

mdadm: hot removed /dev/sdb 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/sdb)

```
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md1 : active raid1 sdc[1] sdd[2]
      10476544 blocks super 1.2 [2/2] [UU]

unused devices: <none>
```

mdadm --add /dev/md1 /dev/sde #Engadir o dispositivo /dev/sde ao RAID

```
mdadm: added /dev/sde
```

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

```
/dev/md1:
  Version : 1.2
  Creation Time : Mon Oct 14 21:12:08 2024
  Raid Level : raid1
  Array Size : 10476544 (9.99 GiB 10.73 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 2
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Mon Oct 14 21:31:24 2024
  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 : df1ae9fa:d92d36c1:0dcfe03f:47aa9559
           Events : 44

Number   Major   Minor   RaidDevice State
  2         8       48         0   active sync   /dev/sdd
  1         8       32         1   active sync   /dev/sdc
  3         8       64        -   spare        /dev/sde
```

Como podemos observar agora /dev/sde forma parte do RAID1 como disco en espera.

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 826 Oct 14 19:58 group
drwx----- 2 root root 16384 Oct 14 21:12 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.

12. Exemplo6. Eliminar e destruir o RAID 1

Imos eliminar e destruir o RAID1 para poder voltar a empregar os 3 discos SATA: 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/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:~# mdadm --stop /dev/md1 #Desactivar volume RAID1, liberando todos os recursos.

root@debianA:~# mdadm --zero-superblock /dev/sdc /dev/sdd /dev/sde #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

Se non formateamos e 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/sdc /dev/sdd --spare-devices=1 /dev/sde #Xerar RAID 1 cos discos sdc, sdd e un disco de reposto /dev/sde

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

root@kaliA:~# ls -l /mnt/md1/ #Listar de forma extendida o contido do directorio /mnt/md5/

```
total 20
-rw-r--r-- 1 root root 826 Oct 14 19:58 group
drwx----- 2 root root 16384 Oct 14 21:12 lost+found
```

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

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

13. 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/sdb /dev/sdc /dev/sdd --spare-devices=1 /dev/sde #Xerar RAID 5 cos discos sdb, sdc e sdd e un disco de repostro /dev/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/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 sdd[4] sde[3](S) sdc[1] sdb[0]
      20953088 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 : Tue Oct 15 14:37:45 2024
    Raid Level : raid5
   Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
   Raid Devices : 3
  Total Devices : 4
 Persistence : Superblock is persistent

   Update Time : Tue Oct 15 14:39:04 2024
     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 : 3f9bd6c1:a30c65cd:95dfdb24:13802f59
         Events : 20

   Number   Major   Minor   RaidDevice State
     0         8       16         0   active sync   /dev/sdb
     1         8       32         1   active sync   /dev/sdc
     4         8       48         2   active sync   /dev/sdd

     3         8       64         -   spare        /dev/sde
```

```
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 usuario@debianA #Agora accedemos a debianA 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 sdb[0] sdd[4] sde[3](S) sdc[1]
      20953088 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 está activo o array de discos RAID5 /dev/md5.

14. 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/sdb, de tal xeito que como temos de respaldo /dev/sde automaticamente este toma o lugar de /dev/sdb e o array de discos RAID5 /dev/md5 segue funcionando:

i. Executar:

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

mdadm: set /dev/sdb 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/sde

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md5 : active raid5 sdb[0](F) sdd[4] sde[3] sdc[1]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [==>.....]   recovery = 12.8% (1344028/10476544) finish=0.9min speed=168003K/sec

unused devices: <none>

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

/dev/md5:
  Version : 1.2
  Creation Time : Tue Oct 15 14:37:45 2024
  Raid Level : raid5
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 3
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 14:50:17 2024
  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 : 90% complete

    Name : debianA:5 (local to host debianA)
    UUID : 3f9bd6c1:a30c65cd:95dfdb24:13802f59
    Events : 38

Number   Major   Minor   RaidDevice State
 3         8       64         0   spare rebuilding /dev/sde
 1         8       32         1   active sync /dev/sdc
 4         8       48         2   active sync /dev/sdd
 0         8       16         -   faulty /dev/sdb
```

Como podemos observar agora /dev/sdb está fallando no array RAID5, estando este agora formado por /dev/sde, /dev/sdd e /dev/sdc

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

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

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso amosa soamente 3 discos no array xa que acabamos de quitar 1 (/dev/sdb)

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md5 : active raid5 sdd[4] sde[3] sdc[1]
      20953088 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 : Tue Oct 15 14:37:45 2024
  Raid Level : raid5
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 14:53:49 2024
  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 : 3f9bd6c1:a30c65cd:95dfdb24:13802f59
    Events : 42

Number   Major   Minor   RaidDevice State   sync   /dev/sde
  3         8       64         0   active sync   /dev/sde
  1         8       32         1   active sync   /dev/sdc
  4         8       48         2   active sync   /dev/sdd
```

Como podemos observar agora /dev/sdb xa non forma parte do array RAID5, estando este agora formado por /dev/sde, /dev/sdd e /dev/sdc

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 Oct 15 14:38 lost+found
-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow
```

- ii. **RAID5 recuperado. O 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.**

15. Exemplo9. Eliminar e destruir o RAID 5

Imos eliminar e destruir 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:~# mkfs.ext4 -F -L 'RAID5' /dev/md5 #Formatear en ext4 coa etiqueta RAID5 o array /dev/md5

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

root@debianA:~# mdadm --zero-superblock /dev/sdb /dev/sdc /dev/sdd /dev/sde #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

Se non formateamos e voltamos a xerar o RAID5 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/md5 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd --spare-devices=1 /dev/sde #Xerar RAID 5 cos discos sdb, sdc e sdd e un disco de reposto /dev/sde

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

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

```
total 20
drwx----- 2 root root 16384 Oct 15 14:38 lost+found
-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow
```

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

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

16. Exemplo10. Crear RAID 6

Realizado o Exemplo9 imos crear un array de discos RAID6 con 6 discos: 4 discos RAID6 (sdb, sdc, sdd, sde) + 2 discos de respaldo (sdf, sdg). Así, imos empregar os discos liberados: sdb, sdc, sdd e sde e a maiores os discos sdf e sdg.

```
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/md6 --level=6 --raid-devices=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde \
--spare-devices=2 /dev/sdf /dev/sdg #Xerar RAID 6 cos discos sdb, sdc, sdd, sde e 2 discos de reposto /dev/sdf e /dev/sdg
OLLO!: Como o comando non cabe nunha soa liña emprégase o carácter multiliña "\n". Unha vez escrito este carácter premer Intro para continuar escribindo o comando.

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/md6 #Xerar o cartafol /mnt/md6

root@debianA:~# mkfs.ext4 -F -L 'RAID6' /dev/md6 #Formatear en ext4 coa etiqueta RAID6 o array /dev/md6

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 md6 #Listar dispositivos de bloques cos seus correspondentes UUID, e filtrar esa saída co patrón md6.

root@debianA:~# UUID_MD6=$(lsblk -o +UUID | grep md6 | awk '{print $NF}' | sort -u) #Declarar unha variable de nome UUID_MD6 co valor do UUID
correspondente ao dispositivo /dev/md6

root@debianA:~# echo "UUID=${UUID_MD6} /mnt/md6 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 : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md6 : active raid6 sdg[5](S) sdf[4](S) sde[3] sdd[2] sdc[1] sdb[0]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUUU]

unused devices: <none>

root@debianA:~# mdadm --detail /dev/md6 #Amosa información extendida sobre o volume RAID /dev/md6

/dev/md6:
    Version : 1.2
  Creation Time : Tue Oct 15 16:51:03 2024
    Raid Level : raid6
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 4
  Total Devices : 6
 Persistence : Superblock is persistent

    Update Time : Tue Oct 15 16:54:56 2024
      State : active
  Active Devices : 4
 Working Devices : 6
  Failed Devices : 0
   Spare Devices : 2


    Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

           Name : debianA:6 (local to host debianA)
          UUID : 267df377:1ebbaaa4:85cf3dc3:f9e8d55d
         Events : 18

   Number   Major   Minor   RaidDevice State
     0         8        16           0  active sync   /dev/sdb
     1         8        32           1  active sync   /dev/sdc
     2         8        48           2  active sync   /dev/sdd
     3         8        64           3  active sync   /dev/sde
     4         8        80            -  spare        /dev/sdf
     5         8        96            -  spare        /dev/sdg

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 usuario@debianA #Agora accedemos a debianA 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]
md6 : active raid6 sdg[5](S) sdf[4](S) sdd[2] sdc[1] sde[3] sdb[0]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]
```

unused devices: <none>

usuario@debianA:~\$ sudo cp -pv /etc/shadow /mnt/md6/ #Copiar o ficheiro /etc/shadow en /mnt/md6 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

IMPORTANTE: Acabamos de comprobar que dende o arranque está activo o array de discos RAID4 /dev/md6.

17. Exemplo11. Degradar RAID 6 e Recuperar

Imos ver que acontece e como recuperar cando se degradan 2 dos discos do array de discos RAID6 con 6 discos: 4 discos RAID6 (sdb, sdc, sdd e sde) e 2 discos de respaldo (sdf e sdg).

Entón, imos provocar o fallo dos discos /dev/sdb e /dev/sdc, de tal xeito que como temos de respaldo /dev/sdf e /dev/sdg automaticamente estes toman o lugar de /dev/sdb e /dev/sdc e o array de discos RAID6 /dev/md6 segue funcionando:

i. Executar:

```
# mdadm --fail /dev/md6 /dev/sdb /dev/sdc #Amosa información extendida sobre o volume RAID /dev/md6

mdadm: set /dev/sdb faulty in /dev/md6
mdadm: set /dev/sdc faulty in /dev/md6

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID6 xa que entran como activos no array
/dev/sdf e /dev/sdg

Personalities : [raid6] [raid5] [raid4] [linear] [multipath] [raid0] [raid1] [raid10]
md6 : active raid6 sdg[5] sdf[4] sdd[2] sdc[1](F) sde[3] sdb[0](F)
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/2] [_UU]
      [>.....] recovery = 1.4% (152796/10476544) finish=2.2min speed=76398K/sec

unused devices: <none>

# mdadm --detail /dev/md6 #Amosa información extendida sobre o volume RAID /dev/md6

/dev/md6:
   Version : 1.2
  Creation Time : Tue Oct 15 16:51:03 2024
   Raid Level : raid6
   Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
   Raid Devices : 4
  Total Devices : 6
 Persistence : Superblock is persistent

   Update Time : Tue Oct 15 17:05:14 2024
     State : clean, degraded, recovering
  Active Devices : 2
 Working Devices : 4
 Failed Devices : 2
 Spare Devices : 2

   Layout : left-symmetric
  Chunk Size : 512K

Consistency Policy : resync

Rebuild Status : 4% complete

       Name : debianA:6 (local to host debianA)
       UUID : 267df377:1ebb4aa4:85cf3dc3:f9e8d55d
       Events : 26

Number   Major   Minor   RaidDevice State
   5         8        96           0 spare rebuilding /dev/sdg
   4         8        80           1 spare rebuilding /dev/sdf
   2         8        48           2 active sync /dev/sdd
   3         8        64           3 active sync /dev/sde

   0         8        16           - faulty /dev/sdb
   1         8        32           - faulty /dev/sdc
```

Como podemos observar agora /dev/sdb e /dev/sdc están fallando no array RAID6, estando este agora formado por /dev/sdd, /dev/sde, /dev/sdf e /dev/sdg

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

```
# ls -l /mnt/md6/ #Listar de forma extendida o contido do directorio /mnt/md6/
```

```
total 20
drwx----- 2 root root 16384 Oct 15 14:38 lost+found
-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow
```

ii. **RAID6 recuperado: O RAID6 segue funcionando aínda que falle 2 dos discos que forma parte do array. E ademais como actúan os discos de respaldo estes sincronizan co array e o RAID6 é recuperado de forma automática.**

18. Exemplo12. Eliminar e destruir o RAID 6

Imos eliminar e destruir o RAID6 para poder voltar a empregar os 6 discos SATA: sdb, sdc, sdd, sde, sdf e sdg.

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/md6 #Desmontar /mnt/md6

root@debianA:~# mkfs.ext4 -F -L 'RAID6' /dev/md6 #Formatear en ext4 coa etiqueta RAID6 o array /dev/md6

root@debianA:~# mdadm --stop /dev/md6 #Desactivar volume RAID6, liberando todos os recursos.

root@debianA:~# mdadm --zero-superblock /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg #Liberar a asociación de dispositivos ao volume RAID6

root@debianA:~# A=\$(grep -n '/mnt/md6' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md6) no ficheiro /etc/fstab e gardalo na variable A

root@debianA:~# sed -i "\${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md6 en /etc/fstab. O número de liña onde aparece /mnt/md6 está gardado na variable A

root@debianA:~# A=\$(grep -n '/dev/md/6' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/6) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A

root@debianA:~# sed -i "\${A},\${A}d" /etc/mdadm/mdadm.conf #Eliminar as liñas correspondentes a /dev/md/6 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md/6 está gardado na variable A

Se non formateamos e voltamos a xerar o RAID6 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/md6 --level=6 --raid-devices=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde--spare-devices=2 /dev/sdf /dev/sdg #Xerar RAID 6 cos discos sdb, sdc, sdd, sde e 2 discos de reposto /dev/sdf e /dev/sdg

root@kaliA:~# mount /dev/md6 /mnt/md6 #Montar o array de discos RAID6 /dev/md6 no directorio /mnt/md6.

root@kaliA:~# ls -l /mnt/md6/ #Listar de forma extendida o contido do directorio /mnt/md6/

total 20

drwx----- 2 root root 16384 Oct 15 16:52 lost+found

-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow

root@debianA:~# ls -l /mnt/md6/ #Listar de forma extendida o contido do directorio /mnt/md6/

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

19. Exemplo13. 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/sdb /dev/sdc /dev/sdd /dev/sde #Xerar RAID 10 cos 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 sde[3] sdd[2] sdc[1] sdb[0]
      20953088 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 : Tue Oct 15 15:38:16 2024
  Raid Level : raid10
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 15:41:14 2024
  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 : 86f52874:fbeb8db1:53a01b5c:72098610
      Events : 19

  Number Major Minor RaidDevice State
    0         8        16         0  active sync set-A  /dev/sdb
    1         8        32         1  active sync set-B  /dev/sdc
    2         8        48         2  active sync set-A  /dev/sdd
    3         8        64         3  active sync set-B  /dev/sde

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 usuario@debianA #Agora accedemos a debianA 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 sdb[0] sdc[1] sdd[2] sde[3]
      20953088 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 está activo o array de discos RAID10 /dev/md10.

20. Exemplo14. 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/sdb, de tal xeito que o array de discos RAID1 dentro de /dev/md10 segue funcionando:

i. Executar:

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

mdadm: set /dev/sdb 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/sdb

Personalities : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md10 : active raid10 sdb[0](F) sdc[1] sdd[2] sde[3]
      20953088 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 : Tue Oct 15 15:38:16 2024
  Raid Level : raid10
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 4
  Total Devices : 4
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 15:53:50 2024
  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 : 86f52874:fbeb8db1:53a01b5c:72098610
      Events : 25

Number   Major   Minor   RaidDevice State
-       -       -       -
  1        8        32        1    active sync set-B /dev/sdc
  2        8        48        2    active sync set-A /dev/sdd
  3        8        80        3    active sync set-B /dev/sde
  0        8        16        -    faulty /dev/sdb
```

Como podemos observar agora /dev/sdb está fallando no array RAID10, estando este agora formado por /dev/sde, /dev/sdd e /dev/sdc

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

mdadm: hot removed /dev/sdb 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/sdb)

Personalities : [raid10] [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
md10 : active raid10 sdc[1] sdd[2] sde[3]
      20953088 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 : Tue Oct 15 15:38:16 2024
  Raid Level : raid10
  Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 10476544 (9.99 GiB 10.73 GB)
  Raid Devices : 4
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 16:00:19 2024
  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 : 86f52874:fbeb8db1:53a01b5c:72098610
        Events : 26

Number   Major   Minor   RaidDevice State
  -   -   -   -   -
  0         0         0         0   removed
  1         8         32         1   active sync set-B /dev/sdc
  2         8         48         2   active sync set-A /dev/sdd
  3         8        80         3   active sync set-B /dev/sde
```

Como podemos observar agora /dev/sdb xa non forma parte do array RAID10, estando este agora formado por /dev/sde, /dev/sdd e /dev/sdc

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 826 Oct 14 19:58 group
drwx----- 2 root root 16384 Oct 15 15:38 lost+found
-rw-r--r-- 1 root root 1758 Oct 14 19:58 passwd
-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow
```

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

21. Exemplo15. Eliminar e destruir o RAID-1+0

Imos eliminar e destruir 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:~# mkfs.ext4 -F -L 'RAID10' /dev/md10 #Formatear en ext4 coa etiqueta RAID10 o array /dev/md10
```

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

```
root@debianA:~# mdadm --zero-superblock /dev/sdb /dev/sdc /dev/sdd /dev/sde #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:~# umount /mnt/md10 #Desmontar /mnt/md10
```

Se non formateamos e 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:~# yes | mdadm --create /dev/md10 --level=10 --raid-devices=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde #Xerar RAID 10 cos discos sdb, sdc, sdd e sde
```

```
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 28
-rw-r--r-- 1 root root    826 Oct 14 19:58 group
drwx----- 2 root root 16384 Oct 15 15:38 lost+found
-rw-r--r-- 1 root root   1758 Oct 14 19:58 passwd
-rw-r----- 1 root shadow  977 Oct 14 19:58 shadow
```

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

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

22. Exemplo16. Crear RAID 50

Imos realizar o Exemplo7 2 veces e logo combinalo co Exemplo1. Así, imos crear 2 arrays independentes de discos RAID5 e combinándoos cun RAID0, tal que:

- md50=RAID0[md51+md52]
- md51=RAID5[sdb, sdc, sdd, sde], onde sde=spare-device
- md52=RAID5[sdf, sdg, sdh, sdi], onde sdi=spare-device

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/md51 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd --spare-devices=1 /dev/sde #Xerar RAID 5 cos discos sdb, sdc e sdd e un disco de reposto /dev/sde

root@debianA:~# yes | mdadm --create /dev/md52 --level=5 --raid-devices=3 /dev/sdf /dev/sdg /dev/sdh --spare-devices=1 /dev/sdi #Xerar RAID 5 cos discos sdf, sdg e sdh e un disco de reposto /dev/sdi

root@debianA:~# yes | mdadm --create /dev/md50 --level=0 --raid-devices=2 /dev/md51 /dev/md52 #Xerar RAID 0 cos RAID md51 e md52

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/md50 #Xerar o cartafol /mnt/md50

root@debianA:~# mkfs.ext4 -F -L 'RAID50' /dev/md50 #Formatear en ext4 coa etiqueta RAID50 o array /dev/md50

root@debianA:~# lsblk -o +UUID | grep md50 #Listar dispositivos de bloques cos seus UUID, e filtrar esa saída co patrón md50.

root@debianA:~# UUID_MD50=\$(lsblk -o +UUID | grep md50 | awk '{print \$NF}' | sort -u) #Declarar unha variable de nome UUID_MD50 co valor do UUID correspondente ao dispositivo /dev/md50

root@debianA:~# echo "UUID=\${UUID_MD50} /mnt/md50 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]
md50 : active raid0 md52[1] md51[0]
      41871360 blocks super 1.2 512k chunks

md52 : active raid5 sdh[4] sdi[3](S) sdg[1] sdf[0]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

md51 : active raid5 sdd[4] sde[3](S) sdc[1] sdb[0]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

root@debianA:~# mdadm --detail /dev/md50 #Amosa información extendida sobre o volume RAID /dev/md50

```
/dev/md50:
  Version : 1.2
  Creation Time : Tue Oct 15 18:45:54 2024
    Raid Level : raid0
    Array Size : 41871360 (39.93 GiB 42.88 GB)
    Raid Devices : 2
    Total Devices : 2
  Persistence : Superblock is persistent

    Update Time : Tue Oct 15 18:45:54 2024
      State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0


    Layout : -unknown-
    Chunk Size : 512K

Consistency Policy : none

           Name : debianA:50 (local to host debianA)
           UUID : d28bee89:ee27886c:ebab481f:f77cc055
           Events : 0

   Number   Major   Minor   RaidDevice State
    0         9       51         0     active sync    /dev/md51
    1         9       52         1     active sync    /dev/md52
```

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 usuario@debianA #Agora accedemos a debianA 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] [raid0] [linear] [multipath] [raid1] [raid10]
md50 : active raid0 md51[0] md52[1]
      41871360 blocks super 1.2 512k chunks

md52 : active (auto-read-only) raid5 sdg[1] sdf[0] sdi[3](S) sdh[4]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

md51 : active raid5 sdb[0] sdc[1] sde[3](S) sdd[4]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

usuario@debianA:~\$ sudo cp -pv /etc/shadow /mnt/md50/ #Copiar o ficheiro /etc/shadow en /mnt/md50 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

IMPORTANTE: Acabamos de comprobar que dende o arranque está activo o array de discos RAID50 /dev/md50.

23. Exemplo17. Degradar RAID50 e Recuperar

Imos ver que acontece e como recuperar cando se degradan 2 discos dentro do array de discos RAID50: 1 disco en md51(sdb) e outro en md52(sdg).
Entón, imos provocar o fallo de /dev/sdb e /dev/sdg, de tal xeito que o array de discos RAID50 segue funcionando, xa que seguen funcionando RAID51 e RAID52:

i. Executar:

```
# mdadm --fail /dev/md51 /dev/sdb #Amosa información extendida sobre o volume RAID /dev/md51
mdadm: set /dev/sdb faulty in /dev/md51

# mdadm --fail /dev/md52 /dev/sdg #Amosa información extendida sobre o volume RAID /dev/md52
mdadm: set /dev/sdg faulty in /dev/md52

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID50 xa que fallaron os dispositivos /dev/sdb e /dev/sdg
Personalities : [raid6] [raid5] [raid4] [raid0] [linear] [multipath] [raid1] [raid10]
md50 : active raid0 md51[0] md52[1]
      41871360 blocks super 1.2 512k chunks

md52 : active (auto-read-only) raid5 sdg[1](F) sdf[0] sdi[3](S) sdh[4]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [U_U]

md51 : active raid5 sdb[0](F) sdc[1] sde[3] sdd[4]
      20953088 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
[==>.....] recovery = 17.7% (1856604/10476544) finish=1.0min speed=142815K/sec
unused devices: <none>

# mdadm --detail /dev/md50 #Amosa información extendida sobre o volume RAID /dev/md50
/dev/md50:
  Version : 1.2
  Creation Time : Tue Oct 15 18:45:54 2024
  Raid Level : raid0
  Array Size : 41871360 (39.93 GiB 42.88 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 18:45:54 2024
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Layout : -unknown-
  Chunk Size : 512K

Consistency Policy : none

           Name : debianA:50 (local to host debianA)
           UUID : d28bee89:ee27886c:ebab481f:f77cc055
           Events : 0

Number   Major   Minor   RaidDevice State   /dev/md/51
  0         9       51         0   active sync    /dev/md/51
  1         9       52         1   active sync    /dev/md/52
```

Como podemos observar agora /dev/sdb está fallando no array RAID51, estando este agora formado por /dev/sde, /dev/sdd e /dev/sdc
Como podemos observar agora /dev/sdg está fallando no array RAID52, estando este agora formado por /dev/sdi, /dev/sdh e /dev/sdf

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

ls -l /mnt/md50/ #Listar de forma extendida o contido do directorio /mnt/md50/

```
total 20
drwx----- 2 root root 16384 Oct 15 18:46 lost+found
-rw-r----- 1 root shadow 977 Oct 14 19:58 shadow
```

ii. RAID50 segue funcionando aínda que fallen 2 discos que forma parte do array.

24. Exemplo18. Eliminar e destruir o RAID50

Imos eliminar e destruir o RAID50 para poder voltar a empregar os 8 discos SATA: sdb, sdc, sdd, sde, sdf, sdg, sdh e sdi.

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/md50 #Desmontar /mnt/md50

root@debianA:~# mkfs.ext4 -F -L 'RAID50' /dev/md50 #Formatear en ext4 coa etiqueta RAID50 o array /dev/md50

root@debianA:~# mdadm --stop /dev/md50 #Desactivar volume RAID50, liberando todos os recursos.

root@debianA:~# mdadm --stop /dev/md51 #Desactivar volume RAID51, liberando todos os recursos.

root@debianA:~# mdadm --stop /dev/md52 #Desactivar volume RAID52, liberando todos os recursos.

root@debianA:~# mdadm --zero-superblock /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh /dev/sdi #Liberar a asociación de dispositivos ao volume RAID50

root@debianA:~# A=\$(grep -n '/mnt/md50' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md50) no ficheiro /etc/fstab e gardalo na variable A

root@debianA:~# sed -i "\${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md50 en /etc/fstab. O número de liña onde aparece /mnt/md50 está gardado na variable A

root@debianA:~# A=\$(grep -n '/dev/md/50' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/50) 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/50 está gardado na variable A

root@debianA:~# A=\$(grep -n '/dev/md/51' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/51) 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/51 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md/51 está gardado na variable A

root@debianA:~# A=\$(grep -n '/dev/md/52' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/52) 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/52 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md/52 está gardado na variable A

Se non formateamos e voltamos a xerar o RAID50 cos dispositivos anteriores seguiriamos 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/md51 --level=5 --raid-devices=3 /dev/sdb /dev/sdc /dev/sdd --spare-devices=1 /dev/sde #Xerar RAID 5 cos discos sdb, sdc e sdd e un disco de reposto /dev/sde
```

```
root@kaliA:~# yes | mdadm --create /dev/md52 --level=5 --raid-devices=3 /dev/sdf /dev/sdg /dev/sdh --spare-devices=1 /dev/sdi #Xerar RAID 5 cos discos sdf, sdg e sdh e un disco de reposto /dev/sdi
```

```
root@kaliA:~# yes | mdadm --create /dev/md50 --level=0 --raid-devices=2 /dev/md51 /dev/md52 #Xerar RAID 0 cos RAID md51 e md52
```

```
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 20
drwx----- 2 root root 16384 Oct 15 20:36 lost+found
-rw-r----- 1 root root 977 Oct 15 20:39 shadow
```

root@debianA:~# ls -l /mnt/md50/ #Listar de forma extendida o contido do directorio /mnt/md50/

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

25. Exemplo19. Crear RAID 60

Imos realizar o Exemplo10 2 veces e logo combinalo co Exemplo1. Así, imos crear 2 arrays independentes de discos RAID6 e combinándoos cun RAID0, tal que:

- md60=RAID0[md61+md62]
- md61=RAID6[sdb, sdc, sdd, sde, sdf, sdg], onde sdf e sdg son spare-device
- md62=RAID6[sdh, sdi, sdj, sdk, sdl, sdm], onde sdl e sdm son spare-device

```
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/md61 --level=6 --raid-devices=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde --spare-devices=2 /dev/sdf /dev/sdg
#Xerar RAID 6 cos discos sdb, sdc, sdd e sde e 2 discos de reposto /dev/sdf e /dev/sdg

root@debianA:~# yes | mdadm --create /dev/md62 --level=6 --raid-devices=4 /dev/sdh /dev/sdi /dev/sdj /dev/sdk --spare-devices=2 /dev/sdl /dev/sdm
#Xerar RAID 6 cos discos sdh, sdi, sdj e sdk e 2 discos de reposto /dev/sdl e /dev/sdm

root@debianA:~# yes | mdadm --create /dev/md60 --level=0 --raid-devices=2 /dev/md61 /dev/md62 #Xerar RAID 0 cos RAID md61 e md62

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/md60 #Xerar o cartafol /mnt/md60

root@debianA:~# mkfs.ext4 -F -L 'RAID60' /dev/md60 #Formatear en ext4 coa etiqueta RAID60 o array /dev/md60

root@debianA:~# lsblk -o +UUID | grep md60 #Listar dispositivos de bloques cos seus UUID, e filtrar esa saída co patrón md60.

root@debianA:~# UUID_MD60=$(lsblk -o +UUID | grep md60 | awk '{print $NF}' | sort -u) #Declarar unha variable de nome UUID_MD60 co valor do UUID
correspondente ao dispositivo /dev/md60

root@debianA:~# echo "UUID=${UUID_MD60} /mnt/md60 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

md60 : active raid0 md62[1] md61[0]
      41871360 blocks super 1.2 512k chunks

md62 : active raid6 sdm[5](S) sdl[4](S) sdk[3] sdj[2] sdi[1] sdh[0]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]
      [=>.....] resync = 6.6% (694204/10476544) finish=1.8min speed=86775K/sec

md61 : active raid6 sdg[5](S) sdf[4](S) sde[3] sdd[2] sdc[1] sdb[0]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]
      [=====>.....] resync = 32.3% (3387932/10476544) finish=0.9min speed=116825K/sec
unused devices: <none>

root@debianA:~# mdadm --detail /dev/md60 #Amosa información extendida sobre o volume RAID /dev/md60

/dev/md60:
  Version : 1.2
  Creation Time : Tue Oct 15 20:34:14 2024
  Raid Level : raid0
  Array Size : 41871360 (39.93 GiB 42.88 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Tue Oct 15 20:34:14 2024
  State : clean
  Active Devices : 2
  Working Devices : 2
  Failed Devices : 0
  Spare Devices : 0

  Layout : -unknown-
  Chunk Size : 512K

Consistency Policy : none

  Name : debianA:60 (local to host debianA)
  UUID : 4d3bd825:4ae73a92:14300b07:7248bff2
  Events : 0

  Number Major Minor RaidDevice State
    0      9     61      0     active sync  /dev/md61
    1      9     62      1     active sync  /dev/md62

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 usuario@debianA #Agora accedemos a debianA 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

```
md60 : active raid0 md62[1] md61[0]
      41871360 blocks super 1.2 512k chunks

md62 : active raid6 sdk[3] sdl[4](S) sdj[2] sdm[5](S) sdh[0] sdi[1]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]

md61 : active raid6 sdc[1] sdf[4](S) sde[3] sdd[2] sdb[0] sdg[5](S)
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/4] [UUUU]

unused devices: <none>
```

usuario@debianA:~\$ sudo cp -pv /etc/shadow /mnt/md60/ #Copiar o ficheiro /etc/shadow en /mnt/md60 a través dos permisos configurados co comando sudo (/etc/sudoers, visudo)

IMPORTANTE: Acabamos de comprobar que dende o arranque está activo o array de discos RAID60 /dev/md60.

26. Exemplo20. Degradar RAID60 e Recuperar

Imos ver que acontece e como recuperar cando se degradan 4 discos dentro do array de discos RAID60: 2 discos en md61(sdb e sdc) e outros 2 en md62(sdh,sdi).

Entón, imos provocar o fallo de /dev/sdb, /dev/sdc, /dev/sdh e /dev/sdi, de tal xeito que o array de discos RAID60 segue funcionando, xa que seguen funcionando RAID61 e RAID62:

i. Executar:

```
# mdadm --fail /dev/md61 /dev/sdb /dev/sdc #Amosa información extendida sobre o volume RAID /dev/md61

mdadm: set /dev/sdb faulty in /dev/md61
mdadm: set /dev/sdc faulty in /dev/md61

# mdadm --fail /dev/md62 /dev/sdh /dev/sdi #Amosa información extendida sobre o volume RAID /dev/md62

mdadm: set /dev/sdh faulty in /dev/md62
mdadm: set /dev/sdi faulty in /dev/md62

# cat /proc/mdstat #Amosa información sobre o estado actual do/s volume/s RAID. Nesta caso está sincronizando o RAID60 xa que fallaron os dispositivo /dev/sdb,
/dev/sdc, /dev/sdh e /dev/sdi

Personalities : [raid6] [raid5] [raid4] [raid0] [linear] [multipath] [raid1] [raid10]
md60 : active raid0 md62[1] md61[0]
      41871360 blocks super 1.2 512k chunks

md62 : active raid6 sdc[3] sdl[4] sdj[2] sdm[5] sdh[0](F) sdi[1](F)
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/2] [__UU]
      [>.....] recovery = 2.0% (219900/10476544) finish=2.3min speed=73300K/sec

md61 : active raid6 sdc[1](F) sdf[4] sde[3] sdd[2] sdb[0](F) sdg[5]
      20953088 blocks super 1.2 level 6, 512k chunk, algorithm 2 [4/2] [__UU]
      [==>.....] recovery = 10.4% (1093852/10476544) finish=1.2min speed=121539K/sec

unused devices: <none>

# mdadm --detail /dev/md60 #Amosa información extendida sobre o volume RAID /dev/md60

/dev/md60:
   Version : 1.2
  Creation Time : Tue Oct 15 20:34:14 2024
    Raid Level : raid0
   Array Size : 41871360 (39.93 GiB 42.88 GB)
  Raid Devices : 2
 Total Devices : 2
 Persistence : Superblock is persistent

   Update Time : Tue Oct 15 20:34:14 2024
     State : clean
  Active Devices : 2
 Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

   Layout : -unknown-
 Chunk Size : 512K

Consistency Policy : none

           Name : debianA:60 (local to host debianA)
          UUID : 4d3bd825:4ae73a92:14300b07:7248bff2
         Events : 0

   Number   Major   Minor   RaidDevice State
    0         9        61         0   active sync   /dev/md/61
    1         9        62         1   active sync   /dev/md/62
```

Como podemos observar agora /dev/sdb e /dev/sdc están fallando no array RAID61, estando este agora formado por /dev/sdg, /dev/sdf, /dev/sde e /dev/sdd
Como podemos observar agora /dev/sdh e /dev/sdi están fallando no array RAID62, estando este agora formado por /dev/sdm, /dev/sdl, /dev/sdk e /dev/sdj

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

```
# ls -l /mnt/md50/ #Listar de forma extendida o contido do directorio /mnt/md50/
```

```
total 20
drwx----- 2 root root 16384 Oct 15 20:36 lost+found
-rw-r----- 1 root root 977 Oct 15 20:39 shadow
```

ii. RAID60 segue funcionando aínda que fallen 4 discos que forma parte do array.

27. Exemplo21. Eliminar e destruir o RAID60

Imos eliminar e destruir o RAID60 para poder voltar a empregar os 12 discos SATA: sdb, sdc, sdd, sde, sdf, sdg, sdh, sdi, sdj, sdk, sdl e sdm.

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/md60 #Desmontar /mnt/md60
```

```
root@debianA:~# mkfs.ext4 -F -L 'RAID60' /dev/md60 #Formatear en ext4 coa etiqueta RAID60 o array /dev/md60
```

```
root@debianA:~# mdadm --stop /dev/md60 #Desactivar volume RAID60, liberando todos os recursos.
```

```
root@debianA:~# mdadm --stop /dev/md61 #Desactivar volume RAID61, liberando todos os recursos.
```

```
root@debianA:~# mdadm --stop /dev/md62 #Desactivar volume RAID62, liberando todos os recursos.
```

```
root@debianA:~# mdadm --zero-superblock /dev/sdb /dev/sdc /dev/sdd /dev/sde /dev/sdf /dev/sdg /dev/sdh /dev/sdi /dev/sdj /dev/sdk /dev/sdl /dev/sdm #Liberar a asociación de dispositivos ao volume RAID60
```

```
root@debianA:~# A=$(grep -n '/mnt/md60' /etc/fstab | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/mnt/md60) no ficheiro /etc/fstab e gardalo na variable A
```

```
root@debianA:~# sed -i "${A}d" /etc/fstab #Eliminar a liña correspondente a /mnt/md60 en /etc/fstab. O número de liña onde aparece /mnt/md60 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/60' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/60) 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/60 en /etc/mdadm/mdadm.conf. O número de liña onde aparece /dev/md/60 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/61' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/61) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A},${A}d" /etc/mdadm/mdadm.conf #Eliminar as liñas correspondentes a /dev/md/61 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md/61 está gardado na variable A
```

```
root@debianA:~# A=$(grep -n '/dev/md/62' /etc/mdadm/mdadm.conf | cut -d':' -f1) #Atopar a liña onde aparece o patrón buscado (/dev/md/62) no ficheiro /etc/mdadm/mdadm.conf e gardalo na variable A
```

```
root@debianA:~# sed -i "${A},${A}d" /etc/mdadm/mdadm.conf #Eliminar as liñas correspondentes a /dev/md/62 en /etc/mdadm/mdadm.conf. Os números de liña onde aparece /dev/md/62 está gardado na variable A
```

Se non formateamos e voltamos a xerar o RAID60 cos dispositivos anteriores seguiriamos 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/md61 --level=6 --raid-devices=4 /dev/sdb /dev/sdc /dev/sdd /dev/sde --spare-devices=2 /dev/sdf /dev/sdg #Xerar RAID 6 cos discos sdb, sdc, sdd e sde e 2 discos de reposto /dev/sdf e /dev/sdg
```

```
root@kaliA:~# yes | mdadm --create /dev/md62 --level=6 --raid-devices=4 /dev/sdh /dev/sdi /dev/sdj /dev/sdk --spare-devices=2 /dev/sdl /dev/sdm #Xerar RAID 6 cos discos sdh, sdi, sdj e sdk e 2 discos de reposto /dev/sdl e /dev/sdm
```

```
root@kaliA:~# yes | mdadm --create /dev/md60 --level=0 --raid-devices=2 /dev/md61 /dev/md62 #Xerar RAID 0 cos RAID md61 e md62
```

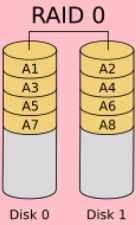
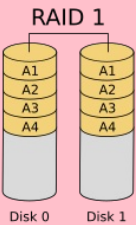
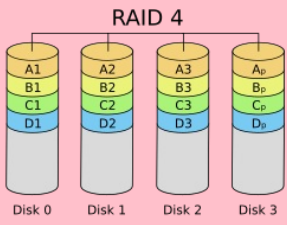
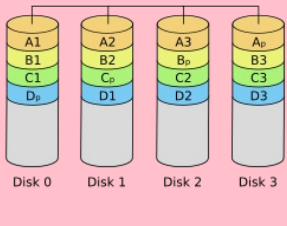
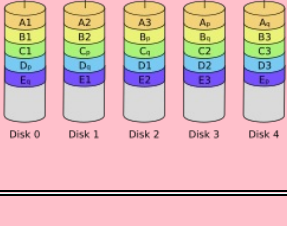
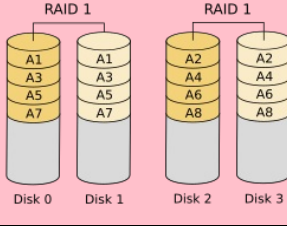
```
root@kaliA:~# mount /dev/md60 /mnt/md60 #Montar o array de discos RAID60 /mnt/md60 no directorio /mnt/md60.
```

```
root@kaliA:~# ls -l /mnt/md60/ #Listar de forma extendida o contido do directorio /mnt/md60/
```

```
total 20
drwx----- 2 root root 16384 Oct 15 20:36 lost+found
-rw-r----- 1 root root 977 Oct 15 20:39 shadow
```

```
root@debianA:~# ls -l /mnt/md60/ #Listar de forma extendida o contido do directorio /mnt/md60/
```

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

Imaxe Wikimedia Commons	Licenza
	en>User:Cburnett, CC BY-SA 3.0 < http://creativecommons.org/licenses/by-sa/3.0/ >, via Wikimedia Commons
	en>User:Cburnett, CC BY-SA 3.0 < http://creativecommons.org/licenses/by-sa/3.0/ >, via Wikimedia Commons
	en>User:Cburnett, CC BY-SA 3.0 < http://creativecommons.org/licenses/by-sa/3.0/ >, via Wikimedia Commons
	en>User:Cburnett, CC BY-SA 3.0 < http://creativecommons.org/licenses/by-sa/3.0/ >, via Wikimedia Commons
	en>User:Cburnett, CC BY-SA 3.0 < http://creativecommons.org/licenses/by-sa/3.0/ >, via Wikimedia Commons
	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

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