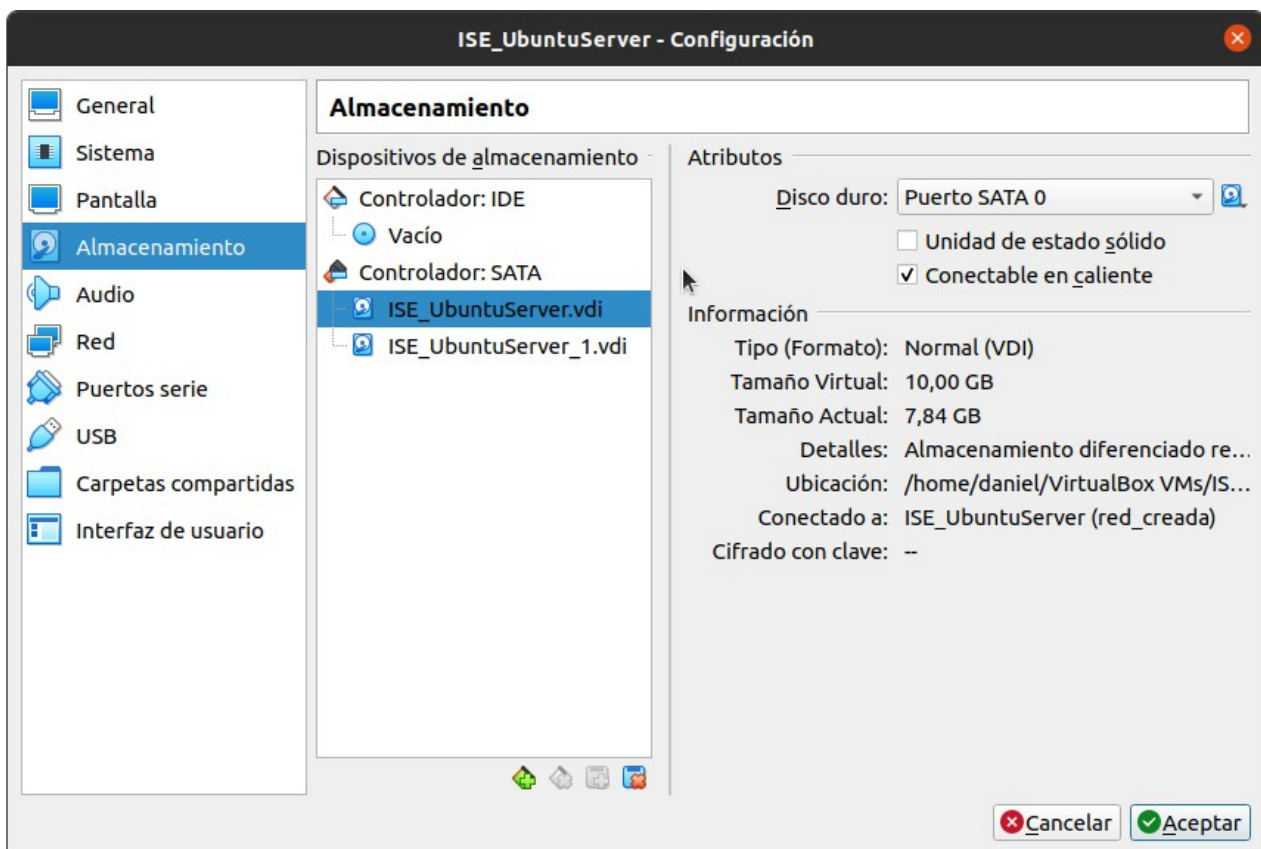


PRACTICA 3: Monitorización y “Profiling”

Lección 1:

Antes de hacer nada debemos hacer una instantánea del estado actual de la máquina, pues vamos a jugar con quitar discos duros y podríamos romperla.

En primer lugar en nuestra máquina de UbuntuServer nos vamos a configuración, almacenamiento y marcamos uno de los discos duros como conectable en caliente.



de forma que una vez este arrancada la máquina ese disco duro podremos conectarlo y desconectarlo. Una vez hecho esto iniciamos la máquina virtual, y podemos comprobar que esté todo correcto con “lsblk”.

Tras esto veremos el estado actual del RAID haciendo “sudo cat /proc/mdstat”.

```

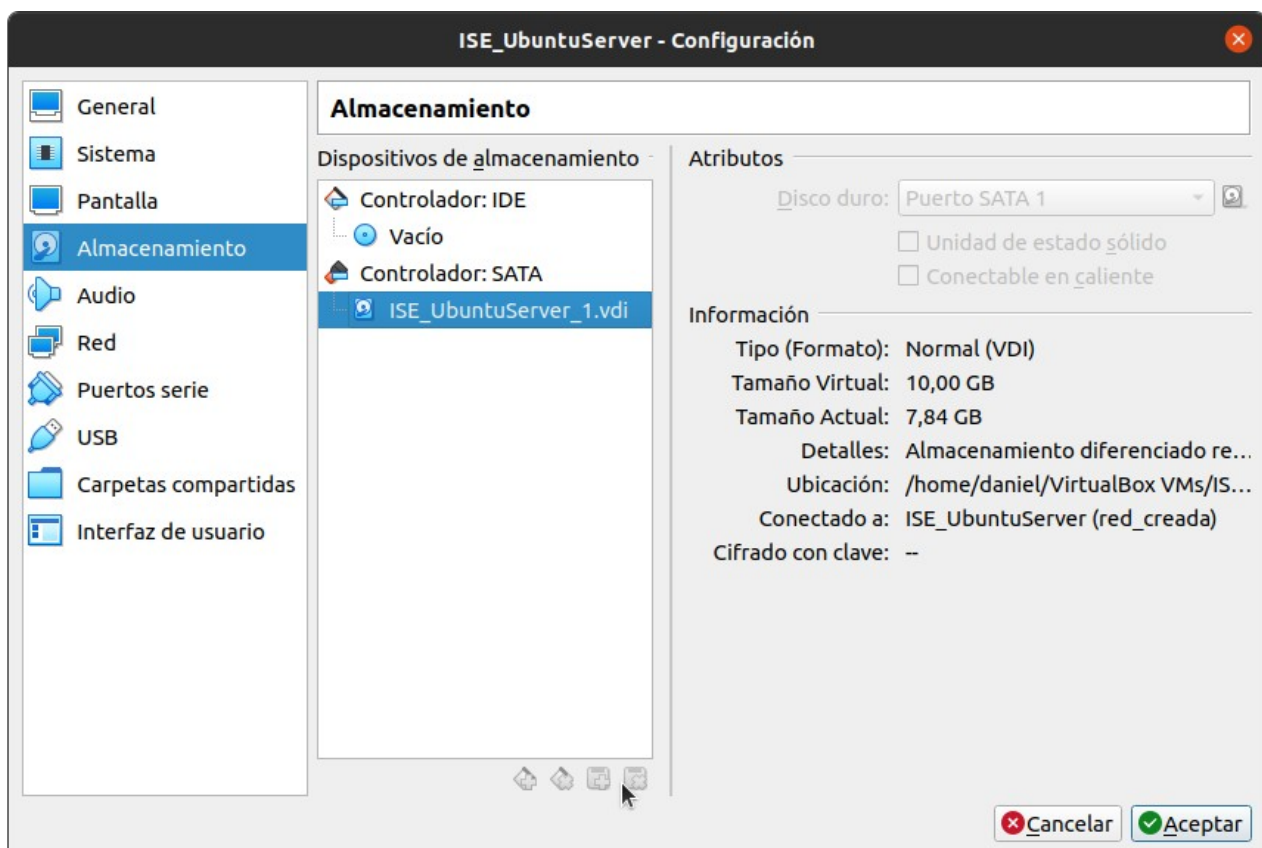
momid@momid:~$ sudo cat /proc/mdstat
[sudo] password for momid:
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1] sda2[0]
      306176 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sdb3[1] sda3[0]
      10166272 blocks super 1.2 [2/2] [UU]

unused devices: <none>
momid@momid:~$ _

```

Vemos que en ambas líneas nos indica que tiene [2/2], ambos discos duros activos [UU] (Up, Up). Ahora vamos a la configuración de la máquina, a almacenamiento y quitamos el disco duro que hemos marcado como conectable en caliente, dándole al icono que hay abajo.



La flecha esta sobre el icono que hay que pulsar, que en nuestro disco duro conectable en caliente estará coloreado. Y seguido le damos a aceptar. Esperamos y veremos que la máquina virtual se quejará, con los siguientes mensajes, y podemos consultar que efectivamente uno de los discos duros ya no está activo volviendo a consultar el fichero “mdstat”.

```

momid@momid:~$ [ 332.215105] sd 2:0:0:0: rejecting I/O to offline device
[ 332.215156] blk_update_request: I/O error, dev sda, sector 618504 op 0x1:(WRITE) flags 0x800 phys
_seg 1 prio class 0
[ 332.215217] md: super_written gets error=10
[ 332.215246] md/raid1:md1: Disk failure on sda3, disabling device.
[ 332.215246] md/raid1:md1: Operation continuing on 1 devices.
[ 336.870607] md/raid1:md0: Disk failure on sda2, disabling device.
[ 336.870607] md/raid1:md0: Operation continuing on 1 devices.

momid@momid:~$ sudo cat /proc/mdstat
Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1]
      306176 blocks super 1.2 [2/1] [_U]

md1 : active raid1 sdb3[1]
      10166272 blocks super 1.2 [2/1] [_U]

unused devices: <none>
momid@momid:~$

```

Vemos que ahora donde antes habían dos U ahora sólo hay una, lo que nos indica que en ambos RAID sólo uno de ellos está activo.

Tras esto hacemos un reboot y vemos que aun así el sistema arranca, pues el RAID se ha autoconfigurado para continuar con un único disco duro, de hecho podemos hacer un “*lblk*” tras el reinicio y ver que falta un disco duro.

```

momid@momid:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                               7:0      0  55.5M  1 loop /snap/core18/1997
loop1                               7:1      0   32.3M  1 loop /snap/snapd/11588
loop2                               7:2      0   71.3M  1 loop /snap/lxd/16099
loop3                               7:3      0   32.3M  1 loop /snap/snapd/11402
loop4                               7:4      0  55.5M  1 loop /snap/core18/1988
loop5                               7:5      0   70.4M  1 loop /snap/lxd/19647
sda                                 8:0      0    10G  0 disk
├─sda1                             8:1      0     1M  0 part
├─sda2                             8:2      0   300M  0 part
│   └─md0                          9:0      0   299M  0 raid1
│       └─md0p1                    259:0    0   294M  0 part /boot
├─sda3                             8:3      0    9.7G  0 part
│   └─md1                          9:1      0    9.7G  0 raid1
│       └─dm_crypt-0               253:0    0    9.7G  0 crypt
│           ├─vg0-lv--home          253:1    0     1G  0 lvm  /home
│           ├─vg0-lv--root          253:2    0     8G  0 lvm  /
│           └─vg0-lv--swap          253:3    0   692M  0 lvm  [SWAP]
sr0                                 11:0     1  1024M  0 rom

```

Una vez probado como funciona el disco conectado en caliente, restauramos la instantánea que hemos hecho al empezar la lección. El caso anterior no es el peor de los casos, el peor caso sería que el disco duro se rompa cuando el ordenador está apagado, para ello directamente desconectamos el primer disco duro con la máquina apagada. Ahora no nos va a dejar iniciar el sistema, nos mandará a initramfs (tardará un rato).

```

[ 1.741631] e1000 0000:00:08:0 enp0s8: renamed from eth1
[ 1.743370] e1000 0000:00:03:0 enp0s3: renamed from eth0
Begin: Loading essential drivers ... [ 1.862091] raid6: avx2x4 gen() 39991 MB/s
[ 1.929937] raid6: avx2x4 xor() 26577 MB/s
[ 1.995512] raid6: avx2x2 gen() 37048 MB/s
[ 2.059779] raid6: avx2x2 xor() 23254 MB/s
[ 2.124518] raid6: avx2x1 gen() 30029 MB/s
[ 2.188914] raid6: avx2x1 xor() 21027 MB/s
[ 2.253363] raid6: sse2x4 gen() 17407 MB/s
[ 2.317642] raid6: sse2x4 xor() 10789 MB/s
[ 2.386282] raid6: sse2x2 gen() 16041 MB/s
[ 2.450047] raid6: sse2x2 xor() 10049 MB/s
[ 2.502191] raid6: sse2x1 gen() 13408 MB/s
[ 2.566129] raid6: sse2x1 xor() 7749 MB/s
[ 2.566419] raid6: using algorithm avx2x4 gen() 39991 MB/s
[ 2.566690] raid6: .... xor() 26577 MB/s, rmw enabled
[ 2.566960] raid6: using avx2x2 recovery algorithm
[ 2.568568] xor: automatically using best checksumming function avx
[ 2.569696] async_tx: api initialized (async)
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... Volume group "vg0" not
found
Cannot process volume group vg0
cryptsetup: Waiting for encrypted source device
UUID=ca06c325-3bd3-4626-96e3-581e1f24e89f...
ALERT! encrypted source device UUID=ca06c325-3bd3-4626-96e3-581e1f24e89f does not exist, can
't unlock dm_crypt-0.
Check cryptopts=source= bootarg: cat /proc/cmdline
or missing modules, devices: cat /proc/modules; ls /dev
Dropping to a shell.

BusyBox v1.30.1 (Ubuntu 1:1.30.1-4ubuntu6.1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

(initramfs)

```

Initramfs es un pequeño sistema de ficheros que se crea usando la memoria RAM, para que se pueda intentar arreglar el sistema de alguna manera y así arrancar de forma normal. Es decir, es un pequeño sistema linux con comandos muy limitados que nos permite intentar arrancar, cosa que no siempre será posible. Nosotros ya sabemos que el error es que falta un disco duro al RAID, luego actuaremos en consecuencia. Dandos dos veces al tabulador se muestran las órdenes disponibles. Usando el comando “*dmseg*” nos muestra por pantalla información del hardware, que podría ser útil, pero nosotros ya sabemos el error, luego hacemos “*cat /proc/mdstat*”.

```

(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : inactive sda2[1](S)
      306176 blocks super 1.2
md1 : inactive sda3[1](S)
      10166272 blocks super 1.2
unused devices: <none>
(initramfs)

```

Vemos que nos enseña que nuestros RAID están inactivos (cuidado, las teclas en initframs están cambiadas). Para arrancar el RAID a mano usamos “*mdadm -R /dev/md0*” y repetimos la orden para el otro RAID.

```
(initramfs) mdadm -R /dev/md0
[ 571.479484] md/raid1:md0: active with 1 out of 2 mirrors
[ 571.479571] md0: detected capacity change from 0 to 313524224
mdadm: started array /dev/md0
(initramfs) [ 571.518693] md0: p1
(initramfs) mdadm -R /dev/md1
[ 575.556689] md/raid1:md1: active with 1 out of 2 mirrors
[ 575.556776] md1: detected capacity change from 0 to 10410262528
mdadm: started array /dev/md1
(initramfs) _
```

Con esto tendremos nuestros dos RAID arrancados manualmente, y comprobamos en mdstat que estén activos.

```
(initramfs) cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active (auto-read-only) raid1 sda2[1]
      306176 blocks super 1.2 [2/1] [_U]

md1 : active (auto-read-only) raid1 sda3[1]
      10166272 blocks super 1.2 [2/1] [_U]

unused devices: <none>
(initramfs) _
```

Vemos que ambos están activos con un disco duro solo. Tras esto hacemos exit, para salir de initramfs y continuar el arranque con normalidad. Entonces desencriptamos el disco y iniciará con normalidad el sistema.

```
momid@momid:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                               7:0      0  32.3M  1 loop  /snap/snapd/11402
loop1                               7:1      0  55.5M  1 loop  /snap/core18/1988
loop2                               7:2      0  55.5M  1 loop  /snap/core18/1997
loop3                               7:3      0  32.3M  1 loop  /snap/snapd/11588
loop4                               7:4      0  70.4M  1 loop  /snap/lxd/19647
loop5                               7:5      0  71.3M  1 loop  /snap/lxd/16099
sda                                  8:0      0   10G   0 disk
├─sda1                              8:1      0    1M   0 part
├─sda2                              8:2      0  300M   0 part
│   └─md0                          9:0      0  299M   0 raid1
│       └─md0p1                    259:0    0  294M   0 part  /boot
├─sda3                              8:3      0   9.7G   0 part
│   └─md1                          9:1      0   9.7G   0 raid1
│       └─dm_crypt-0               253:0    0   9.7G   0 crypt
│           ├─vg0-lv--home          253:1    0    1G   0 lvm    /home
│           ├─vg0-lv--root          253:2    0    8G   0 lvm    /
│           └─vg0-lv--swap          253:3    0  692M   0 lvm    [SWAP]
sr0                                 11:0     1  1024M   0 rom
```

Ahora apagamos la máquina y nos vamos a la configuración de la máquina y creamos un disco duro nuevo. Importante una vez creado el disco duro modificar el puerto SATA al que se conecta, pues sino el primero que comprobará será el del nuevo disco duro, que no tiene nada. Una vez hecho esto volvemos a arrancar la máquina. Comprobamos que el nuevo disco duro (sdb) se haya añadido correctamente con *lsblk* y tiene que estar completamente vacío.

Ya solo nos queda volver a conectar el disco duro a los RAID, pero hay que tener en cuenta que nuestro disco duro tiene particiones, luego las haremos a mano. Para ello usamos la herramienta “*sudo fdisk /dev/sdb*”, *fdisk* y el disco que queremos particionar. Nos meterá en una interfaz y entonces usamos el comando *n*, para indicar que queremos añadir una nueva partición, luego indicamos el tipo de partición (primaria o secundaria), luego el número de partición y finalmente el sector de inicio y el sector final de la partición.

```

    p   primary (0 primary, 0 extended, 4 free)
    e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-20971519, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519, default 20971519): 4096

Created a new partition 1 of type 'Linux' and of size 1 MiB.

Command (m for help): n
Partition type
    p   primary (1 primary, 0 extended, 3 free)
    e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2): 2
First sector (4097-20971519, default 6144): 4097
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4097-20971519, default 20971519): 618496

Created a new partition 2 of type 'Linux' and of size 300 MiB.

Command (m for help): n
Partition type
    p   primary (2 primary, 0 extended, 2 free)
    e   extended (container for logical partitions)
Select (default p): p
Partition number (3,4, default 3): 3
First sector (618497-20971519, default 620544):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (620544-20971519, default 20971519):

Created a new partition 3 of type 'Linux' and of size 9.7 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

momid@momid:~$ _

```

Con esto tendremos nuestras particiones creadas, y lo podemos comprobar con `lsblk`. Ahora solo falta unir cada partición a su correspondiente RAID y así hemos recuperado del todo nuestro sistema.

Para lo último usamos “`sudo mdadm --add /dev/md0 /dev/sdb2`” y de forma análoga para las otras particiones.

```

sr0                11:0      1 1024M  0 rom
momid@momid:~$ sudo mdadm --add /dev/md0 /dev/sdb2
mdadm: added /dev/sdb2
momid@momid:~$ sudo mdadm --add /dev/md1 /dev/sdb3
mdadm: added /dev/sdb3
momid@momid:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                              7:0      0  70.4M  1 loop  /snap/lxd/19647
loop1                              7:1      0  55.5M  1 loop  /snap/core18/1997
loop2                              7:2      0  55.5M  1 loop  /snap/core18/1988
loop3                              7:3      0  71.3M  1 loop  /snap/lxd/16099
loop4                              7:4      0  32.3M  1 loop  /snap/snapd/11402
loop5                              7:5      0  32.3M  1 loop  /snap/snapd/11588
sda                                8:0      0   10G  0 disk
├─sda1                             8:1      0    1M  0 part
├─sda2                             8:2      0   300M  0 part
│   └─md0                          9:0      0   299M  0 raid1
├─sda3                             8:3      0   9.7G  0 part
│   └─md1                          9:1      0   9.7G  0 raid1
│       └─dm_crypt-0               253:0    0   9.7G  0 crypt
│           ├─vg0-lv--home          253:1    0    1G  0 lvm    /home
│           ├─vg0-lv--root          253:2    0    8G  0 lvm    /
│           └─vg0-lv--swap          253:3    0   692M  0 lvm    [SWAP]
sdb                                8:16     0   10G  0 disk
├─sdb1                             8:17     0    1M  0 part
├─sdb2                             8:18     0   300M  0 part
│   └─md0                          9:0      0   299M  0 raid1
├─sdb3                             8:19     0   9.7G  0 part
│   └─md1                          9:1      0   9.7G  0 raid1
│       └─dm_crypt-0               253:0    0   9.7G  0 crypt
│           ├─vg0-lv--home          253:1    0    1G  0 lvm    /home
│           ├─vg0-lv--root          253:2    0    8G  0 lvm    /
│           └─vg0-lv--swap          253:3    0   692M  0 lvm    [SWAP]
sr0                                11:0      1 1024M  0 rom
momid@momid:~$

```

Otra forma sería mirar el fichero “/proc/mdstat”. Podemos usar “*watch -n 1 cat /proc/mdstat*” para comprobar como se recupera nuestro disco duro, pues el último es muy grande y por tanto no es instantáneo.

```

Every 1.0s: cat /proc/mdstat                                momid: Fri Apr 30 21:06:09 2021

Personalities : [raid1] [linear] [multipath] [raid0] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[2] sda2[1]
      306176 blocks super 1.2 [2/2] [UU]

md1 : active raid1 sdb3[2] sda3[1]
      10166272 blocks super 1.2 [2/1] [_U]
      [=====>....]  recovery = 80.4% (8176000/10166272) finish=0.4min speed=74692K/sec

unused devices: <none>

```

Finalmente cuando termine el recovery podremos observar con un *lsblk* que ya todo funciona adecuadamente.

```

momid@momid:~$ lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
loop0                              7:0      0  70.4M  1 loop  /snap/lxd/19647
loop1                              7:1      0  55.5M  1 loop  /snap/core18/1997
loop2                              7:2      0  55.5M  1 loop  /snap/core18/1988
loop3                              7:3      0  71.3M  1 loop  /snap/lxd/16099
loop4                              7:4      0  32.3M  1 loop  /snap/snapd/11402
loop5                              7:5      0  32.3M  1 loop  /snap/snapd/11588
sda                                 8:0      0   10G   0 disk
├─sda1                             8:1      0    1M   0 part
├─sda2                             8:2      0  300M   0 part
├─┬md0                             9:0      0  299M   0 raid1
│  └─md0p1                         259:0    0  294M   0 part  /boot
├─sda3                             8:3      0   9.7G   0 part
├─┬md1                             9:1      0   9.7G   0 raid1
│  └─dm_crypt-0                   253:0    0   9.7G   0 crypt
│     └─vg0-lv--home               253:1    0    1G   0 lvm   /home
│        └─vg0-lv--root             253:2    0    8G   0 lvm   /
│           └─vg0-lv--swap           253:3    0  692M   0 lvm   [SWAP]
sdb                                 8:16     0   10G   0 disk
├─sdb1                             8:17     0    1M   0 part
├─sdb2                             8:18     0  300M   0 part
├─┬md0                             9:0      0  299M   0 raid1
│  └─md0p1                         259:0    0  294M   0 part  /boot
├─sdb3                             8:19     0   9.7G   0 part
├─┬md1                             9:1      0   9.7G   0 raid1
│  └─dm_crypt-0                   253:0    0   9.7G   0 crypt
│     └─vg0-lv--home               253:1    0    1G   0 lvm   /home
│        └─vg0-lv--root             253:2    0    8G   0 lvm   /
│           └─vg0-lv--swap           253:3    0  692M   0 lvm   [SWAP]
sr0                                 11:0     1 1024M   0 rom
momid@momid:~$ _

```


Lección 2:

Primero marcamos uno de los discos de la máquina de UbuntuServer como conectable en caliente, para que cuando se tenga el script automatizado se tenga comprueba que dicho script funciona correctamente.

Tras esto creamos un archivo `mon RAID.py` que contenga

```
f=open('/proc/mdstat')
for line in f:
    b=re.findall('\[[U]*[_]+[U]*\]',line)
    if(b!=[]):
        print("ERROR EN RAID")
print("FIN SCRIPT")
```

```
"mon RAID.py" [New] 8L, 169C written
momid@momid:~$
```

Este script trabaja sobre el archivo “*/proc/mdstat*” y lo que comprueba es que ambos discos duros del RAID estén activos. Podemos comprobar que funciona correctamente con el comando “*python3 mon_raid.py*”, y nos debería salir fin del script.

Tras esto creamos dos archivos, uno `"/etc/systemd/system/mon_raid.timer"` que contendrá,

```
mon1d@mon1d:/etc/systemd/system$ _
```

y otro archivo `/etc/systemd/system/mon_raid.service` que contiene,

~~~~~

```
momid@momid:/etc/systemd/system$
```

Tras esto hacemos “`systemctl start mon_raid.service mon_raid.timer`” para activar el servicio y el temporizador, un “`systemctl enable mon_raid.service mon_raid.timer`” para activar el servicio y el temporizador cuando el sistema se inicie, y finalmente “`systemctl status mon_raid.service mon_raid.timer`” para comprobar el status y tendría que salir algo como

```
momid@momid:/etc/systemd/system$ sudo systemctl start mon_raid.service mon_raid.timer
momid@momid:/etc/systemd/system$ sudo systemctl enable mon_raid.service mon_raid.timer
Created symlink /etc/systemd/system/timers.target.wants/mon_raid.timer → /etc/systemd/system/mon_raid.timer.
momid@momid:/etc/systemd/system$ sudo systemctl status mon_raid.service mon_raid.timer
• mon_raid.service - Monitor RAID Service
   Loaded: loaded (/etc/systemd/system/mon_raid.service; static; vendor preset: enabled)
   Active: inactive (dead) since Tue 2021-05-04 22:37:15 UTC; 9s ago
   TriggeredBy: • mon_raid.timer
   Main PID: 1136 (code=exited, status=0/SUCCESS)

May 04 22:37:15 momid systemd[1]: Started Monitor RAID Service.
May 04 22:37:15 momid python3[1136]: FIN SCRIPT
May 04 22:37:15 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:37:19 momid systemd[1]: /etc/systemd/system/mon_raid.service:4: Failed to parse service t>

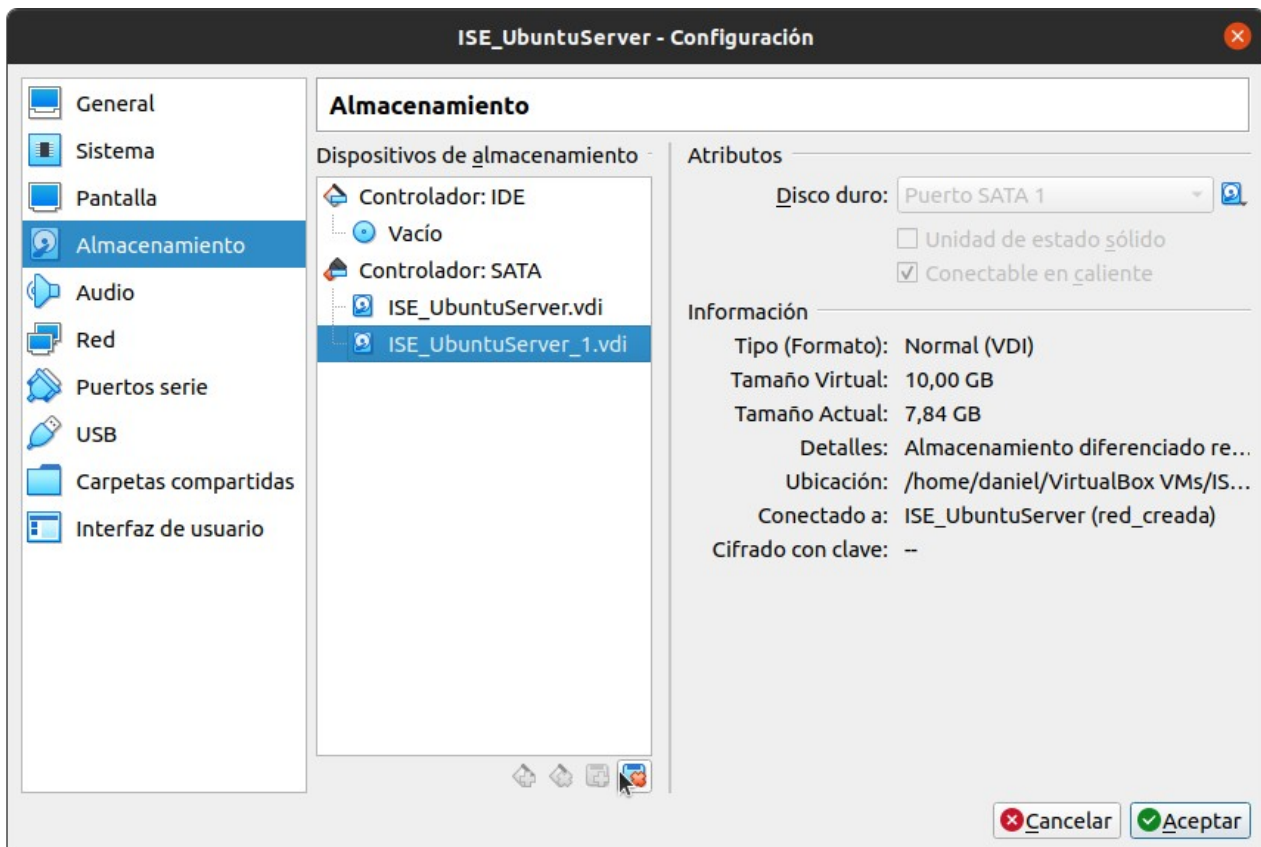
• mon_raid.timer - Monitor RAID Service
   Loaded: loaded (/etc/systemd/system/mon_raid.timer; enabled; vendor preset: enabled)
   Active: active (waiting) since Tue 2021-05-04 22:37:15 UTC; 9s ago
   Trigger: Tue 2021-05-04 22:38:00 UTC; 35s left
   Triggers: • mon_raid.service

May 04 22:37:15 momid systemd[1]: Started Monitor RAID Service.
lines 1-18/18 (END)
```

Podemos usar “`sudo journalctl -u mon_raid --since="yesterday"`”, donde el -u nos indica que solo nos muestra el servicio que le pidamos, mientras que el since nos indica desde cuando nos va a mandar los mensajes. Esta parte se podría mejorar para que nos mandase al correo periódicamente los correspondientes mensajes.

```
momid@momid:/etc/systemd/system$ sudo journalctl -u mon_raid --since="yesterday"
-- Logs begin at Thu 2021-03-18 21:24:53 UTC, end at Tue 2021-05-04 22:25:28 UTC. --
May 04 22:24:49 momid systemd[1]: Started Monitor RAID Service.
May 04 22:24:49 momid python3[3268]: FIN SCRIPT
May 04 22:24:49 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:25:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:25:00 momid python3[3344]: FIN SCRIPT
May 04 22:25:00 momid systemd[1]: mon_raid.service: Succeeded.
momid@momid:/etc/systemd/system$
```

Tras esto comprobamos que monitoriza correctamente, para ello el disco que hemos marcado como conectable en caliente lo desconectamos.



Una vez desconectado vemos que el sistema se ha quejado, de hecho si volvemos a hacer un journalctl nos pondra ERROR RAID.

```
momid@momid:/etc/systemd/system$ [ 1339.310717] sd 3:0:0:0: rejecting I/O to offline device
[ 1339.310788] blk_update_request: I/O error, dev sdb, sector 618504 op 0x1:(WRITE) flags 0x800 phys
_seg 1 prio class 0
[ 1339.310876] md: super_written gets error=10
[ 1339.310916] md/raid1:md1: Disk failure on sdb3, disabling device.
[ 1339.310916] md/raid1:md1: Operation continuing on 1 devices.
[ 1343.974632] md/raid1:md0: Disk failure on sdb2, disabling device.
[ 1343.974632] md/raid1:md0: Operation continuing on 1 devices.
sudo journalctl -u mon_raid --since="yesterday"
-- Logs begin at Thu 2021-03-18 21:24:53 UTC, end at Tue 2021-05-04 22:28:09 UTC. --
May 04 22:24:49 momid systemd[1]: Started Monitor RAID Service.
May 04 22:24:49 momid python3[3268]: FIN SCRIPT
May 04 22:24:49 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:25:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:25:00 momid python3[3344]: FIN SCRIPT
May 04 22:25:00 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:26:03 momid systemd[1]: Started Monitor RAID Service.
May 04 22:26:03 momid python3[3376]: FIN SCRIPT
May 04 22:26:03 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:27:00 momid systemd[1]: Started Monitor RAID Service.
May 04 22:27:00 momid python3[3393]: FIN SCRIPT
May 04 22:27:00 momid systemd[1]: mon_raid.service: Succeeded.
May 04 22:28:03 momid systemd[1]: Started Monitor RAID Service.
May 04 22:28:03 momid python3[3432]: ERROR EN RAID
May 04 22:28:03 momid python3[3432]: ERROR EN RAID
May 04 22:28:03 momid python3[3432]: FIN SCRIPT
May 04 22:28:03 momid systemd[1]: mon_raid.service: Succeeded.
momid@momid:/etc/systemd/system$ _
```

Apagamos la máquina y dejamos el RAID como estaba con la instantánea que hemos hecho previamente.

En primer lugar comprobamos que el servicio ssh no esté enmascarado, pues taskel nos lo puede haber roto.

```
momid@momid:~$ sudo systemctl status sshd.service
[sudo] password for momid:
• ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-05-04 22:41:23 UTC; 45s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 625 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 673 (sshd)
    Tasks: 1 (limit: 4619)
   Memory: 4.6M
   CGroup: /system.slice/ssh.service
           └─673 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

May 04 22:41:22 momid systemd[1]: Starting OpenBSD Secure Shell server...
May 04 22:41:23 momid sshd[673]: Server listening on 0.0.0.0 port 22022.
May 04 22:41:23 momid sshd[673]: Server listening on :: port 22022.
May 04 22:41:23 momid systemd[1]: Started OpenBSD Secure Shell server.
momid@momid:~$
```

Si saliese el servicio enmascarado habría que hacer un “*sudo apt install openssh-server*” con la opción mantener la configuración local que te la dará a elegir.

Tras esto hacemos “*ansible --version*” y nos tiene que mostrar la versión instalada, en otro caso lo instalamos con apt (“*sudo apt install ansible*”).

```
momid@momid:~$ ansible --version
ansible 2.9.6
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/momid/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.8.5 (default, Jan 27 2021, 15:41:15) [GCC 9.3.0]
momid@momid:~$ _
```

Para indicar que servidores tiene que automatizar ansible vamos al archivo “*/etc/ansible/hosts*” y en este archivo incluiremos las IP de las máquinas que queremos monitorizar. Además el propio fichero en si nos indica como rellenarlo.

```
#
# It should live in /etc/ansible/hosts
#
# - Comments begin with the '#' character
# - Blank lines are ignored
# - Groups of hosts are delimited by [header] elements
# - You can enter hostnames or ip addresses
# - A hostname/ip can be a member of multiple groups
# Ex 1: Ungrouped hosts, specify before any group headers.
#green.example.com
#blue.example.com
#192.168.100.1
#192.168.100.10
192.168.56.105
192.168.56.110
# Ex 2: A collection of hosts belonging to the 'webservers' group
#[webservers]
#alpha.example.org
#beta.example.org
#192.168.1.100
#192.168.1.110
# If you have multiple hosts following a pattern you can specify
# them like this:
#www[001:006].example.com
# Ex 3: A collection of database servers in the 'dbservers' group
#[dbservers]
"/etc/ansible/hosts" 47L, 1013C written
momid@momid:~$ _
```

Tras esto lanzamos el archivo de configuración utilizando el comando “*ansible all -m ping momid*”, en este caso nos debe de dar unreachable.

```
momid@momid:~$ ansible all -m ping -u momid
192.168.56.105 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.105 port 22: Connection refused",
  "unreachable": true
}
192.168.56.110 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: ssh: connect to host 192.168.56.110 port 22: No route to host",
  "unreachable": true
}
momid@momid:~$
```

Tras esto arrancamos la máquina de CentOS para hacer la prueba. Tras encender la máquina si volvemos a probar el comando anterior nos volverá a dar unreachable. En primer lugar para que funcione el ping en ubuntu le mandaremos a nuestra propia máquina ubuntu la clave publica de ella misma. Antes que nada autorizamos el envío de contraseñas en el archivo de configuración de sshd. Tras esto hacemos reiniciamos el servicio y hacemos “*ssh-copy-id momid@192.168.56.105 -p 22022*”

```

momid@momid:~$ ssh-copy-id momid@192.168.56.105 -p 22022
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/momid/.ssh/id_rsa.pub"
The authenticity of host '[192.168.56.105]:22022 ([192.168.56.105]:22022)' can't be established.
ECDSA key fingerprint is SHA256:SYwYgmtKxnTqKUpfALtZVaySrLQhE6Vy4z803ER02ZA.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install all the new keys
momid@192.168.56.105's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh -p '22022' 'momid@192.168.56.105'"
and check to make sure that only the key(s) you wanted were added.

momid@momid:~$

```

Tras esto volvemos a probar el ping, y seguirá sin funcionar, pues el puerto al que manda el ping no es el adecuado. Para indicar a ansible el puerto al que mandar el ping vamos al fichero de configuración de ansible en `/etc/ansible/ansible.cfg` y ahí encontraremos el campo `remote_port`, y ese campo lo descomentamos y ponemos 22022. Tras esto probamos una vez más el ping.

```

momid@momid:~$ ansible all -m ping -u momid
192.168.56.105 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3"
  },
  "changed": false,
  "ping": "pong"
}
192.168.56.110 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/libexec/platform-python"
  },
  "changed": false,
  "ping": "pong"
}
momid@momid:~$

```

Y vemos que el ping es correcto tanto a UbuntuServer como a CentOS, tras esto probaremos nuestro script, pero nos dará error en CentOS, puesto que nuestro script de python no está en CentOS, para ello hacemos `ansible all -a "python3 /home/momid/mon_raid.py" -u momid` de esta forma Ubuntu intentará por ssh mandar dicha orden a todas las máquinas incluidas en hosts.

```

momid@momid:~$ ansible all -a "python3 /home/momid/mon_raid.py" -u momid
192.168.56.105 | CHANGED | rc=0 >>
FIN SCRIPT
192.168.56.110 | FAILED | rc=2 >>
python3: can't open file '/home/momid/mon_raid.py': [Errno 2] No such file or directory
non-zero return code
momid@momid:~$ _

```

Vemos que en la máquina de CentOS nos da error, por lo que ya hemos dicho, no está el archivo en CentOS. Para mandar el script por ssh a CentOS usaremos `scp -P 22022 mon_raid.py momid@192.168.56.110:/home/momid/mon_raid.py` `-P` nos indica el puerto, seguido del fichero, el `usuario@dir_ip`:ruta donde mandar el archivo.

```

momid@momid:~$ scp -P 22022 mon_raid.py momid@192.168.56.110:/home/momid/mon_raid.py
mon_raid.py                                100% 154    93.0KB/s   00:00
momid@momid:~$

```

Tras esto podemos comprobar que el fichero esté en CentOS con ls (la máquina debía estar abierta) y finalmente podemos comprobar con ansible que funciona.

```
momid@momid:~$ ansible all -a "python3 /home/momid/mon RAID.py" -u momid
192.168.56.105 | CHANGED | rc=0 >>
FIN SCRIPT
192.168.56.110 | CHANGED | rc=0 >>
FIN SCRIPT
momid@momid:~$
```

Todo esto es mejorable, pues vemos que se tiene que compartir usuario, ruta, etc. Todo esto es ampliable con los ejecutables de ansible. Estos ejecutables son los playbooks, y están escritos en YAML. En ellos se puede ampliar todo lo anterior, pues todo lo anterior es para un servidor sencillo.