

Instalación mdadm en Linux

La herramienta que vamos a utilizar para crear, montar, administrar y supervisar nuestro software RAID-1 se llama mdadm (abreviatura de multiple d iesgos ADM). En distribuciones Linux como Fedora, CentOS, RHEL o Arch Linux, mdadm está disponible por defecto. En distros basadas en Debian, mdadm se puede instalar con aptitud o apt-get.

Fedora, CentOS o RHEL

Como mdadm viene pre-instalado, todo lo que tienes que hacer es iniciar el servicio de monitoreo de RAID, y configurarlo para que se inicie automáticamente en el arranque:

```
# systemctl start mdmonitor  
# systemctl enable mdmonitor
```

Para CentOS / RHEL 6, utilice estos comandos en su lugar:

```
# service mdmonitor start  
# chkconfig mdmonitor on
```

Debian, Ubuntu o Linux Mint

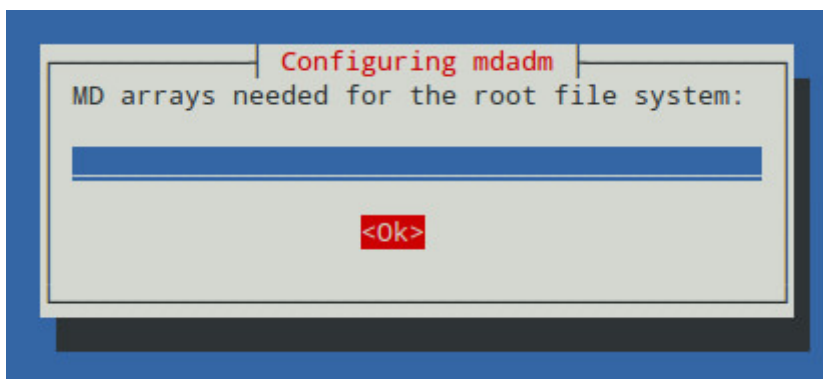
En Debian y sus derivados, mdadm se puede instalar con aptitud o apt-get:

```
# aptitude install mdadm
```

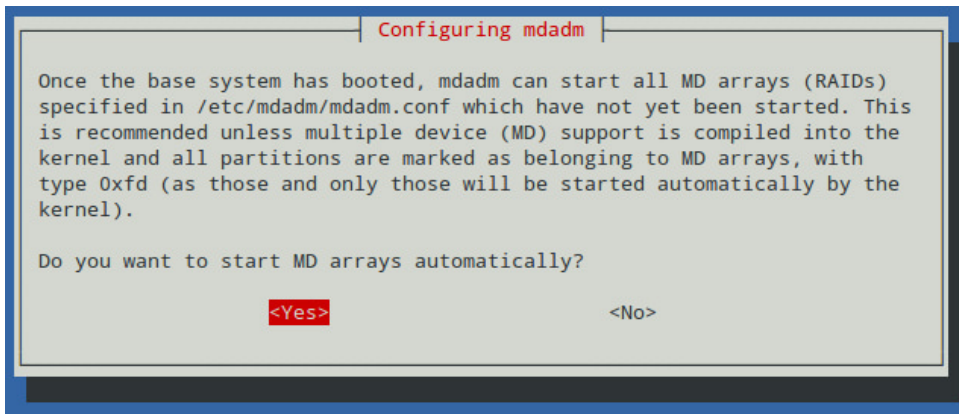
En Ubuntu, se le pedirá configurar postfix MTA para el envío de notificaciones por correo electrónico (como parte del seguimiento de RAID). Puede saltar todo por ahora.

En Debian, la instalación se iniciará con el siguiente mensaje explicativo para ayudarnos a decidir si o no vamos a instalar el sistema de archivos raíz en un array RAID. Lo que necesitamos para entrar en la siguiente pantalla dependerá de la decisión. Léalo detenidamente:

Ya que no utilizaremos nuestra RAID-1 para el sistema de ficheros raíz, vamos a dejar la respuesta en blanco:



Cuando preguntamos si queremos empezar (reensamblar) nuestra array automáticamente en cada inicio, seleccione "Sí". Tenga en cuenta que tendrá que añadir una entrada al archivo /etc /fstab tarde para que el conjunto sea montado correctamente durante el proceso de arranque también.



Particionando el disco duro

Ahora es el momento de preparar los dispositivos físicos que se utilizarán en nuestra matriz. Para esta configuración, He conectado dos unidades USB de 8 GB que se han identificado como / dev / sdb y / dev / sdc de salida de dmesg:

```
# dmesg | less

[ 60.014863] sd 3:0:0:0: [sdb] 15826944 512-byte logical blocks: (8.10 GB/7.5
[ 75.066466] sd 4:0:0:0: [sdc] 15826944 512-byte logical blocks: (8.10 GB/7.5
```

Vamos a utilizar fdisk para crear una partición primaria en cada disco que ocupará la totalidad de su tamaño. Los siguientes pasos muestran cómo realizar esta tarea en / dev / sdb, y se supone que esta unidad no se ha dividido aún (de lo contrario, podemos borrar la partición existente (s) para empezar con un disco limpio):

Pulse 'p' para imprimir la tabla de particiones actual:

```
# fdisk /dev/sdb
```

```
root@debian:~# fdisk /dev/sdb

Command (m for help): p

Disk /dev/sdb: 8103 MB, 8103395328 bytes
250 heads, 62 sectors/track, 1021 cylinders, total 15826944 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xd644cb8a

   Device Boot      Start         End      Blocks    Id  System
Command (m for help):
```

(Si se encuentran uno o más particiones, se pueden eliminar con la opción 'd'. Entonces 'w' opción se utiliza para aplicar los cambios).

Puesto que se encuentran, sin particiones, vamos a crear una nueva partición primaria ['n'] como una partición primaria ['p'], asigne el número de partición = ['1'] a él, y luego indicar su tamaño. Puede pulsar la tecla Intro para aceptar los valores por defecto propuestos, o introduzca un valor de su elección, como se muestra en la imagen de abajo.

```

Command (m for help): n 1
Partition type:
   p  primary (0 primary, 0 extended, 4 free)
   e  extended
Select (default p): p 2
Partition number (1-4, default 1): 1 3
First sector (2048-15826943, default 2048): 4
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-15826943, default 15826943): 5
Using default value 15826943

Command (m for help): w 6
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.

```

Ahora repita el mismo proceso para /dev/sdc.

Si tenemos dos unidades de distintos tamaños, digamos 750 GB y 1 TB por ejemplo, debemos crear una partición primaria de 750 GB en cada uno de ellos, y usar el espacio restante en la unidad más grande para otro propósito, independiente de la matriz RAID.

Crear un array RAID-1

Una vez que haya terminado con la creación de la partición primaria en cada unidad, utilice el comando siguiente para crear una matriz RAID-1:

```
# mdadm -Cv /dev/md0 -l1 -n2 /dev/sdb1 /dev/sdc1
```

Donde:

-Cv: Crea un array y produce verbose output.

/dev/md0: es el nombre del array.

-l1 (l es "level" -nivel-): Indica que el nivel será un array RAID-1.

-n2: indica que vamos a añadir dos particiones a la matriz, es decir, /dev/sdb1 y /dev/sdc1.

El comando anterior es equivalente a:

```
# mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/
```

Si, alternatively, usted desea agregar un dispositivo de repuesto para reemplazar un disco defectuoso en el futuro, puede agregar: '--spare-devices=1 /dev/sdd1' al comando anterior.

Conteste "y" cuando se le solicite si desea continuar con la creación de una matriz, a continuación, presione Entrar:

```

root@debian:~# mdadm -Cv /dev/md0 -l1 -n2 /dev/sdb1 /dev/sdc1
mdadm: Note: this array has metadata at the start and
      may not be suitable as a boot device. If you plan to
      store '/boot' on this device please ensure that
      your boot-loader understands md/v1.x metadata, or use
      --metadata=0.90
mdadm: size set to 7908288K
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
root@debian:~#

```

Usted puede comprobar el progreso con el siguiente comando:

```
# cat /proc/mdstat
```

```

root@debian:~# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
      7908288 blocks super 1.2 [2/2] [UU]
      [=>.....] resync = 10.3% (817280/7908288) finish=31.5min speed=3743K/sec

unused devices: <none>
root@debian:~#

```

Otra forma de obtener más información sobre un array RAID (tanto mientras está siendo ensamblado y después de finalizado el proceso) es:

```

# mdadm --query /dev/md0
# mdadm --detail /dev/md0 (or mdadm -D /dev/md0)

```

```

root@debian:~# mdadm --query /dev/md0
/dev/md0: 7.54GiB raid1 2 devices, 0 spares. Use mdadm --detail for more detail.
root@debian:~# mdadm --detail /dev/md0
/dev/md0:
  Version : 1.2
  Creation Time : Thu Sep 11 13:52:35 2014
  Raid Level : raid1
  Array Size : 7908288 (7.54 GiB 8.10 GB)
  Used Dev Size : 7908288 (7.54 GiB 8.10 GB)
  Raid Devices : 2
  Total Devices : 2
  Persistence : Superblock is persistent

  Update Time : Thu Sep 11 14:25:02 2014
  State : active
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

  Name : debian:0 (local to host debian)
  UUID : 3f2386f9:f99381f1:1716de1e:bf9a197e
  Events : 18

   Number Major Minor RaidDevice State
    0         8      17        0 active sync /dev/sdb1
    1         8      33        1 active sync /dev/sdc1
root@debian:~#

```

De la información proporcionada por '-D mdadm', quizás el más útil es la que muestra el estado de la matriz. El estado activo significa que en la actualidad existe actividad de I / O sucediendo. Otros estados posibles están limpias (toda la actividad de E / S se ha completado), degradada (uno de los dispositivos es defectuoso o que falta), la resincronización (el sistema se está recuperando de un cierre sucio, como un corte de energía), o en recuperación (una nueva unidad ha sido añadido a la matriz, y los datos se están copiadas de la otra unidad en él), por nombrar los estados más comunes.

Formateo y montaje de una matriz RAID

El siguiente paso está formateando (con ext4 en este ejemplo) la matriz:

```
# mkfs.ext4 /dev/md0
```

```
root@debian:~# mkfs.ext4 /dev/md0
mke2fs 1.42.5 (29-Jul-2012)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
494832 inodes, 1977072 blocks
98853 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2025848832
61 block groups
32768 blocks per group, 32768 fragments per group
8112 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

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

root@debian:~#
```

Ahora vamos a montar la matriz, y verifique que se montó correctamente:

```
# mount /dev/md0 /mnt
# mount
```

```
root@debian:~# mount /dev/md0 /mnt
root@debian:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=62949,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmx)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=51480k,mode=755)
/dev/mapper/debian-root on / type ext4 (rw,relatime,errors=remount-ro,user_xattr)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=102940k)
/dev/sda1 on /boot type ext2 (rw,relatime,errors=continue)
/dev/mapper/debian-home on /home type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
/dev/mapper/debian-tmp on /tmp type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
/dev/mapper/debian-usr on /usr type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
/dev/mapper/debian-var on /var type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
/dev/md0 on /mnt type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
root@debian:~#
```

Supervisar una matriz RAID

La herramienta mdadm viene con capacidad de supervisión RAID incorporado. Cuando mdadm está configurado para ejecutarse como un demonio (que es el caso de nuestra configuración RAID), que periódicamente encuestas arrays RAID, y los informes sobre los eventos detectados existentes a través de correo electrónico de notificación o registro de syslog . Opcionalmente, también se puede configurar para invocar comandos de contingencia (por ejemplo, reintentar o quitar un disco) al detectar los errores críticos.

Por defecto, mdadm escanea todas las particiones existentes y los grupos MD, y registra cualquier evento detectado en / var / log / syslog. Alternativamente, puede especificar los dispositivos y matrices RAID para escanear en mdadm.conf situado en /etc/mdadm/mdadm.conf (basado en Debian) o /etc/mdadm.conf (basado en RedHat), en el siguiente formato. Si mdadm.conf no existe, cree uno.

```
DEVICE /dev/sd[bcde]1 /dev/sd[ab]1

ARRAY /dev/md0 devices=/dev/sdb1,/dev/sdc1
ARRAY /dev/md1 devices=/dev/sdd1,/dev/sde1
.....

# optional email address to notify events
MAILADDR your@email.com
```

Después de modificar la configuración de mdadm, reinicie el demonio mdadm:

En Debian, Ubuntu o Linux Mint:

```
# service mdadm restart
```

En Fedora, CentOS / RHEL 7:

```
# systemctl restart mdmonitor
```

En CentOS / RHEL 6:

```
# service mdmonitor restart
```

Auto-montar una matriz RAID

Ahora vamos a añadir una entrada en el archivo / etc / fstab para montar la matriz en / mnt automáticamente durante el arranque (se puede especificar cualquier otro punto de montaje):

```
# echo "/dev/md0 /mnt ext4 defaults 0 2" >> /etc/fstab
```

Para verificar que el monte funciona bien, ahora desmontar la matriz, mdadm reiniciar y volver a montar. Podemos ver que / dev / md0 se ha montado según la entrada que acaba de agregar a / etc / fstab:

```
# umount /mnt
# service mdadm restart (on Debian, Ubuntu or Linux Mint)
or systemctl restart mdmonitor (on Fedora, CentOS/RHEL7)
or service mdmonitor restart (on CentOS/RHEL6)
# mount -a
```

```

root@debian:~# umount /mnt
root@debian:~# service mdadm restart
[ ok ] Stopping MD monitoring service: mdadm --monitor.
[ ok ] Starting MD monitoring service: mdadm --monitor.
root@debian:~# mount -a
root@debian:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=62949,mode=75
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,pt
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=51480k,mode=755)
/dev/mapper/debian-root on / type ext4 (rw,relatime,errors=remount-ro,user_
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=102940k)
/dev/sda1 on /boot type ext2 (rw,relatime,errors=continue)
/dev/mapper/debian-home on /home type ext4 (rw,relatime,user_xattr,barrier=
/dev/mapper/debian-tmp on /tmp type ext4 (rw,relatime,user_xattr,barrier=1,
/dev/mapper/debian-usr on /usr type ext4 (rw,relatime,user_xattr,barrier=1,
/dev/mapper/debian-var on /var type ext4 (rw,relatime,user_xattr,barrier=1,
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
/dev/md0 on /mnt type ext4 (rw,relatime,user_xattr,barrier=1,data=ordered)
root@debian:~#

```

Ahora estamos listos para acceder a la matriz RAID a través de /mnt punto de montaje. Para probar la matriz, copiaremos el fichero /etc/passwd (cualquier otro archivo va a hacer) en /mnt:

```

root@debian:~# cp /etc/passwd /mnt
root@debian:~# ls -l /mnt
total 20
drwx----- 2 root root 16384 Sep 11 15:09 lost+found
-rw-r--r-- 1 root root 1418 Sep 11 16:04 passwd
root@debian:~#

```

En Debian, necesitamos decirle al demonio mdadm para iniciar automáticamente el array RAID durante el arranque mediante la variable AUTOSTART en true en el archivo de mdadm /etc/default/:

```
AUTOSTART=true
```

La simulación de fallos de la unidad

Vamos a simular una unidad defectuosa y retírela con los siguientes comandos. Tenga en cuenta que en un escenario de la vida real, no es necesario marcar un dispositivo como defectuosa en primer lugar, como ya estará en ese estado en el caso de un fracaso.

En primer lugar, desmontar la matriz:

```
# umount /mnt
```

Ahora, observe cómo la salida de 'mdadm -D /dev/md0' indica los cambios después de la realización de cada comando a continuación.

```

# mdadm /dev/md0 --fail /dev/sdb1 #Marks /dev/sdb1 as faulty
# mdadm --remove /dev/md0 /dev/sdb1 #Removes /dev/sdb1 from the array

```

Después, cuando se tiene una nueva unidad para la sustitución, vuelva a agregar de nuevo la unidad:

```
# mdadm /dev/md0 --add /dev/sdb1
```

Los datos a continuación, se inicia de inmediato que ser reconstruido en / dev / sdb1:

Number	Major	Minor	RaidDevice	State
2	8	17	0	spare rebuilding /dev/sdb1
1	8	33	1	active sync /dev/sdc1

Tenga en cuenta que los pasos detallados anteriormente se aplican para los sistemas con discos intercambiables en caliente. Si usted no tiene este tipo de tecnología, también tendrá que dejar una matriz actual, y apagar el sistema en primer lugar con el fin de sustituir la pieza:

```
# mdadm --stop /dev/md0  
# shutdown -h now
```

A continuación, agregue la nueva unidad y volver a montar la matriz:

```
# mdadm /dev/md0 --add /dev/sdb1  
# mdadm --assemble /dev/md0 /dev/sdb1 /dev/sdc1
```

Espero que esto le ayude.