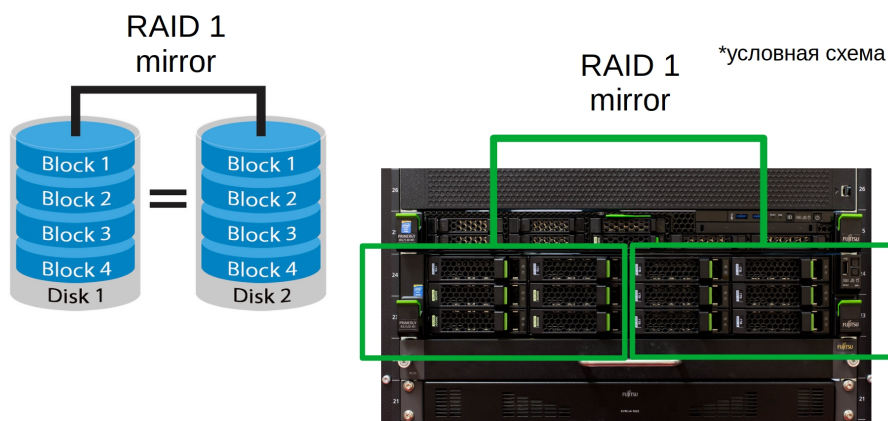


Представьте – у вас есть сервер, на котором работает важный сервис. Вы подняли на сервере LVM, периодически добавляете диски и вроде всё нормально. Пока в один день не выходит из строя один из дисков. Диски периодически портятся, это совершенно нормально и вы можете по гарантии заменить диск. Гораздо страшнее, что случится с вашими данными. Во первых, восстановить данные из повреждённого диска не всегда возможно, а если что-то и можно восстановить, то это обычно стоит бешеных денег. Но это только часть проблем. Если какой-то логический том использовал несколько дисков, включая проблемный, то придётся потратить много часов, а то и дней, чтобы восстановить хоть что-то, а о полном восстановлении можно забыть. Это уже тот случай, когда нужно восстанавливать из бэкапа.

RAID - Redundant Array of Independent Disks



Учитывая, сколько проблем может создать один повреждённый диск, компании готовы заплатить чуть больше за железо, чтобы избежать таких проблем. И задачи примерно такие – выход диска из строя не должен приводить к потере данных и остановке работы сервиса. Для решения этих задач была разработана технология RAID – избыточный массив независимых

NAME

md - Multiple Device driver aka Linux Software RAID

SYNOPSIS

```
/dev/mdn
/dev/md/n
/dev/md/name
```

DESCRIPTION

The **md** driver provides virtual devices that are created from one or more independent underlying devices. This array of devices often contains redundancy and the devices are often disk drives, hence the acronym RAID which stands for a Redundant Array of Independent Disks.

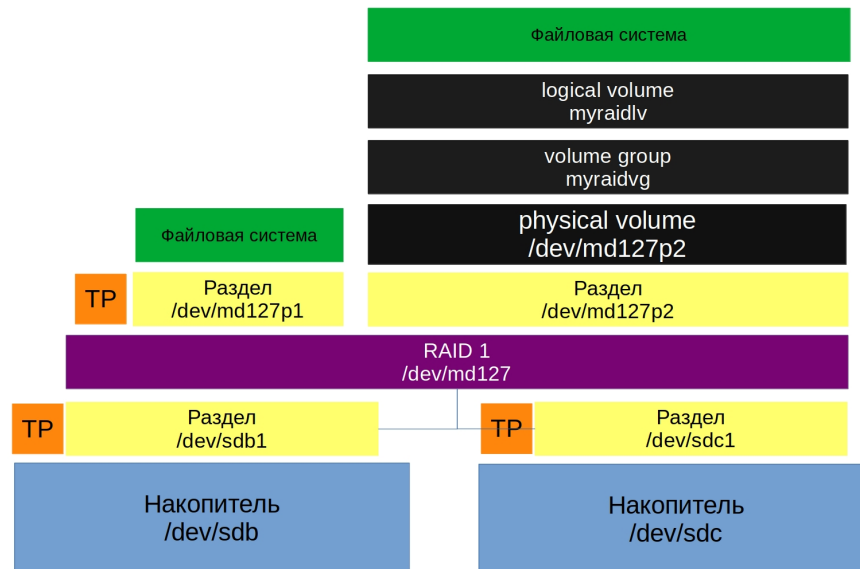
Вообще, RAID можно настроить и с помощью LVM. Но сам по себе LVM при работе с рейдом обращается к другому модулю для работы с рейдом – md – multiple devices - `man md`. И лучше научиться работать с самим md чтобы не зависеть от LVM, так как LVM это делает поверхностно, а с md можно сделать гораздо больше. Мы с вами для теста поднимем RAID 1 на sdb и sdc.

```
[user@centos8 ~]$ df -h /mydata/
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/myvg-mylv  729M  2.5M  691M   1% /mydata
[user@centos8 ~]$ sudo umount /mydata
[user@centos8 ~]$ sudo vgremove myvg
Do you really want to remove volume group "myvg" containing 1 logical volumes? [y/n]: y
Do you really want to remove active logical volume myvg/mylv? [y/n]: y
Logical volume "mylv" successfully removed
Volume group "myvg" successfully removed
[user@centos8 ~]$ sudo wipefs -a /dev/sdb /dev/sdc
/dev/sdb: 8 bytes were erased at offset 0x00000200 (gpt): 45 46 49 20 50 41 52 54
/dev/sdb: 8 bytes were erased at offset 0x3ffffe00 (gpt): 45 46 49 20 50 41 52 54
/dev/sdb: 2 bytes were erased at offset 0x000001fe (PMBR): 55 aa
/dev/sdb: calling ioctl to re-read partition table: Success
/dev/sdc: 8 bytes were erased at offset 0x00000218 (LVM2_member): 4c 56 4d 32 20 30 30 31
[user@centos8 ~]$ sudo nano /etc/fstab
[user@centos8 ~]$ tail -1 /etc/fstab
#/dev/mapper/myvg-mylv  /mydata  ext4 noatime 0 0
[user@centos8 ~]$
```

Но сейчас на них находится LVM - `df -h /mydata`, который мы создали в прошлый раз, он примонтирован в /mydata. У меня здесь ничего важного нет, поэтому я могу отмонтировать файловую систему - `sudo umount /mydata`, удалить vg - `sudo vgremove myvg`, которая при удалении также предложит удалить логический том, а потом затереть все метки с дисков - `sudo wipefs -a /dev/sdb /dev/sdc`. Также не стоит забывать про строчку в fstab - `sudo nano /etc/fstab`; `tail -1 /etc/fstab`, её я просто прокомментирую, так как она нам ещё понадобится.

Прежде чем создадим рейд, прибегнем к одной особенности программного рейда. Если говорить про аппаратный рейд, то очень сильно рекомендуется использовать диски одного вендора, одной модели и чуть ли не одной партии, потому что при одинаковых параметрах дисков – их объёме и скорости чтения и записи, гарантированно всё будет окей. Если диски разные – то при создании рейда будет выбираться скорость и объём самого малого и медленного

диска. У разных вендоров объёмы дисков могут немного отличаться, пусть хоть на 1 сектор, не важно. И если у вас вышел из строя диск, вы купили другой и внезапно оказалось, что там на 100 секторов меньше, чем на других дисках рейда – то ваш рейд просто не примет новый диск. Поэтому для аппаратного рейда практически всегда используются одинаковые диски.



Что касается программного рейда, такой проблемы можно избежать. Для этого под рейд можно давать не весь диск, а чуть меньше, чтобы в случае замены подошёл любой новый диск нужного объёма. Для этого можно использовать разделы.

```
[user@centos8 ~]$ sudo fdisk -l /dev/sdb1 /dev/sdc1
Disk /dev/sdb1: 954 MiB, 1000341504 bytes, 1953792 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 /dev/sdc1: 954 MiB, 1000341504 bytes, 1953792 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
[user@centos8 ~]$
```

Поэтому я создам раздел на каждом из дисков на 1 гигабайт - `sudo fdisk /dev/sdb; g, n,enter,enter,+1GB, p, w;` `sudo fdisk /dev/sdc; g, n,enter,enter,+1GB, p, w;` `sudo fdisk -l /dev/sdb1 /dev/sdc1`, что чуть меньше реального объёма, но в дальнейшем мне будет проще заменить один из дисков.

EXAMPLES

mdadm --query /dev/name-of-device

This will find out if a given device is a RAID array, or is part of one, and will provide brief information about the device.

mdadm --assemble --scan

This will assemble and start all arrays listed in the standard config file. This command will typically go in a system startup file.

mdadm --stop --scan

This will shut down all arrays that can be shut down (i.e. are not currently in use). This will typically go in a system shutdown script.

mdadm --follow --scan --delay=120

If (and only if) there is an Email address or program given in the standard config file, then monitor the status of all arrays listed in that file by polling them ever 2 minutes.

mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hd[ac]1

Create /dev/md0 as a RAID1 array consisting of /dev/hda1 and /dev/hdc1.

Дальше нам понадобится утилита mdadm. У этой утилиты также подробный мануал с примерами - `man mdadm`, `/EXAMPLES`.

```
[user@centos8 ~]$ sudo mdadm --create myraid1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1
[sudo] password for user:
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/myraid1 started.
[user@centos8 ~]$
```

Пишем `sudo mdadm --create` указываем имя рейда, допустим, `myraid1`; указываем уровень рейда - `--level=1` и указываем количество устройств и их имена `--raid-devices=2 /dev/sdb1 /dev/sdc1` - `sudo mdadm --create myraid1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1`. Утилита нас предупреждает, что если мы планируем держать директорию `/boot` на этих дисках, нужно кое-что проверить или изменить, но пока мы этого не планируем. Причём здесь директория `/boot` мы разберём в другой раз, а пока продолжим. Пишем `y` и `enter`. Видим, что рейд у нас создан.

```
[user@centos8 ~]$ ls -l /dev/md
total 0
lrwxrwxrwx. 1 root root 8 Mar 28 18:35 myraid1 -> ../md127
[user@centos8 ~]$ sudo mdadm -D /dev/md127
/dev/md127:
        Version : 1.2
    Creation Time : Sun Mar 28 18:35:44 2021
        Raid Level : raid1
        Array Size : 975872 (953.00 MiB 999.29 MB)
    Used Dev Size : 975872 (953.00 MiB 999.29 MB)
        Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Sun Mar 28 18:35:49 2021
        State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0
```

Если посмотреть - `ls -l /dev/md`, увидим, что это ссылка на устройство `/dev/md127`. Информацию о рейде можем посмотреть с помощью утилиты `mdadm` с опцией `detail` или просто `D` - `sudo mdadm -D /dev/md127`.

```
echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf
mdadm --detail --scan >> mdadm.conf
This will create a prototype config file that describes currently active arrays that are
known to be made from partitions of IDE or SCSI drives. This file should be reviewed
before being used as it may contain unwanted detail.

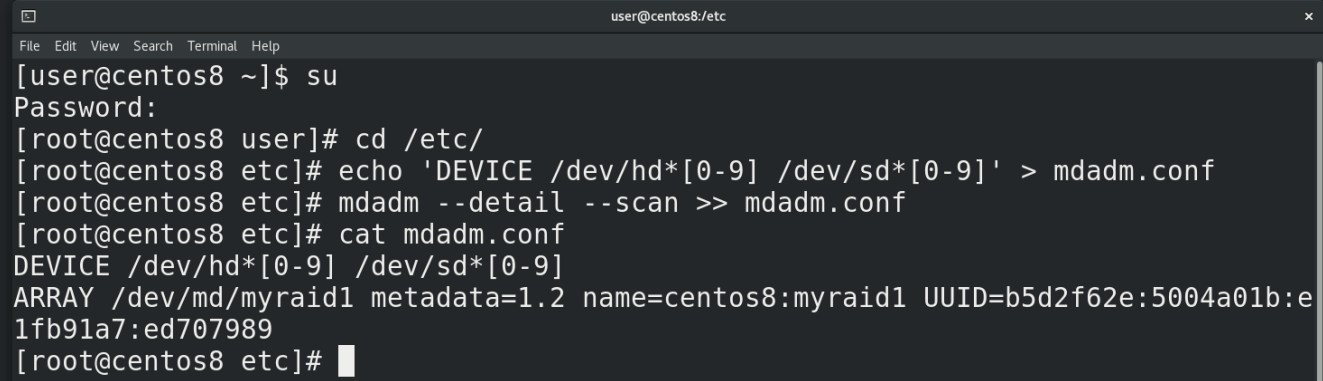
echo 'DEVICE /dev/hd[a-z] /dev/sd*[a-z]' > mdadm.conf
mdadm --examine --scan --config=mdadm.conf >> mdadm.conf
This will find arrays which could be assembled from existing IDE and SCSI whole drives
(not partitions), and store the information in the format of a config file. This file
is very likely to contain unwanted detail, particularly the devices= entries. It should
be reviewed and edited before being used as an actual config file.
```

Дальше нам нужно сохранить текущую конфигурацию рейда. Для этого откроем `man mdadm` и найдём две строчки с `echo` — `man mdadm`, `/echo`. Будет два примера – с разделами или целыми дисками, мы выберем первый, так как мы создали рейд на разделах.

MDADM.CONF(5)	File Formats Manual	MDADM.CONF(5)
NAME	mdadm.conf - configuration for management of Software RAID with mdadm	
SYNOPSIS	/etc/mdadm.conf	
DESCRIPTION	mdadm is a tool for creating, managing, and monitoring RAID devices using the md driver in Linux.	

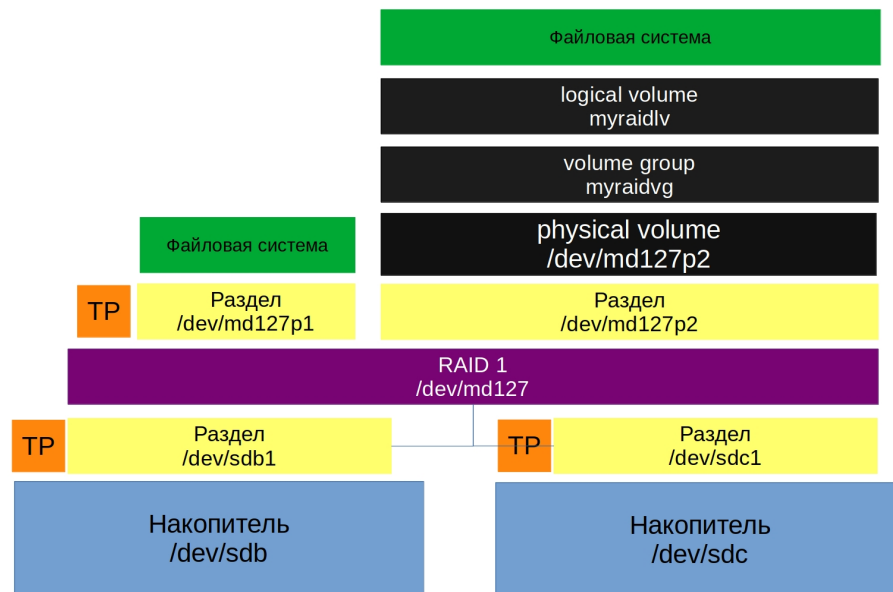
Станем рутом и зайдём в директорию etc. Правда директория может разниться в зависимости от дистрибутива, поэтому лучше всё же посмотреть в ман - `man mdadm.conf`.

```
echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf
mdadm --detail --scan >> mdadm.conf
```



```
[user@centos8 ~]$ su
Password:
[root@centos8 user]# cd /etc/
[root@centos8 etc]# echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf
[root@centos8 etc]# mdadm --detail --scan >> mdadm.conf
[root@centos8 etc]# cat mdadm.conf
DEVICE /dev/hd*[0-9] /dev/sd*[0-9]
ARRAY /dev/md/myraid1 metadata=1.2 name=centos8:myraid1 UUID=b5d2f62e:5004a01b:e1fb91a7:ed707989
[root@centos8 etc]#
```

Первая строчка — `echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf` - создаст файл `mdadm.conf` со списком разделов, которые нужно проверять при запуске системы на наличие рейда. Можно это подправить, оставив только текущие разделы, либо оставить как есть. Вторая строчка - `mdadm --detail --scan >> mdadm.conf` - добавит в файл информацию о текущем рейде и его идентификатор — `cat /etc/mdadm.conf`.



А дальше мы можем записать на файл устройства файловую систему с помощью `mkfs` или указать его как физический том для LVM. Мы сделаем и то и другое, для повторения. У вас может возникнуть вопрос – а зачем LVM, если RAID также объединяет диски, давая общее пространство? Вспомните, что у LVM есть и другие полезные возможности – те же снапшоты, более динамическое управление логическими томами, по сравнению со стандартной таблицей разделов.

```
[user@centos8 ~]$ sudo fdisk -l /dev/md127
Disk /dev/md127: 953 MiB, 999292928 bytes, 1951744 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
Disklabel type: gpt
Disk identifier: E621A24A-4F8B-BD49-BBF3-04CA068EC06A

Device            Start      End Sectors  Size Type
/dev/md127p1      2048    587775  585728   286M Linux filesystem
/dev/md127p2    587776 1951710 1363935   666M Linux filesystem
[user@centos8 ~]$ ls -l /dev/md/
total 0
lrwxrwxrwx. 1 root root 8 Mar 28 19:01 myraid1 -> ../md127
lrwxrwxrwx. 1 root root 10 Mar 28 19:01 myraid1p1 -> ../md127p1
lrwxrwxrwx. 1 root root 10 Mar 28 19:01 myraid1p2 -> ../md127p2
[user@centos8 ~]$
```

Чтобы сделать и стандартный раздел и логический том создадим на этом устройстве таблицу разделов - `sudo fdisk /dev/md127`; `g` и два раздела - `n,enter,enter,+300MB`; `n,enter,enter,enter,p` - и сохраним `w`; `sudo fdisk -l /dev/md127`. Как видите, разделы получили названия `md127p1` и `md127p2`. Тоже самое касается ссылки в директории `/dev/md` - `ls -l /dev/md`.


```
[user@centos8 ~]$ sudo mkfs.ext4 /dev/md127p1
mke2fs 1.45.4 (23-Sep-2019)
Creating filesystem with 292864 1k blocks and 73440 inodes
Filesystem UUID: f77f89da-70f0-4339-bb93-55b46ebd88d6
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

На первый раздел сразу запишем файловую систему - `sudo mkfs.ext4 /dev/md127p1`.

```
[user@centos8 ~]$ sudo pvcreate /dev/md127p2
Physical volume "/dev/md127p2" successfully created.
[user@centos8 ~]$ sudo vgcreate myraidvg /dev/md127p2
Volume group "myraidvg" successfully created
[user@centos8 ~]$ sudo lvcreate myraidvg -n myraidlv -l 80%FREE
Logical volume "myraidlv" created.
[user@centos8 ~]$ █
```

А на второй создадим логический том. Вспоминаем – физический том – группа томов – логический том. `sudo pvcreate /dev/md127p2; sudo vgcreate myraidvg /dev/md127p2; sudo lvcreate myraidvg -n myraidlv -l 80%FREE`.

```
[user@centos8 ~]$ sudo mkfs.ext4 /dev/mapper/myraidvg-myraidlv
mke2fs 1.45.4 (23-Sep-2019)
Creating filesystem with 135168 4k blocks and 33840 inodes
Filesystem UUID: d761e8a3-8821-4dd2-9f53-c8cadb8a9740
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

[user@centos8 ~]$ sudo nano /etc/fstab
[user@centos8 ~]$ tail -1 /etc/fstab
/dev/mapper/myraidvg-myraidlv /mydata ext4 noatime 0 0
[user@centos8 ~]$ █
```

Смотрим в `/dev/mapper` - появилась ссылка на логический том. Записываем на него файловую систему - `sudo mkfs.ext4 /dev/mapper/myraidvg-myraidlv` - и подправляем запись в `fstab` - `sudo nano /etc/fstab`, `tail -1 /etc/fstab`.

```
[user@centos8 ~]$ sudo mount -a
[user@centos8 ~]$ df -h /mydata/
Filesystem                Size      Used Avail Use% Mounted on
/dev/mapper/myraidvg-myraidlv 504M    804K   466M   1% /mydata
[user@centos8 ~]$
```

Можно для теста примонтировать - `sudo mount -a`, `df -h /mydata`. Всё работает. У нас, конечно, прибавился еще один уровень абстракции, но с точки зрения пользовательского пространства ничего не изменилось – та же директория `/mydata`.

```

Raid Devices : 2
Total Devices : 1
Persistence : Superblock is persistent

Update Time : Sun Mar 28 19:18:28 2021
State : clean, degraded
Active Devices : 1
Working Devices : 1
Failed Devices : 0
Spare Devices : 0

Consistency Policy : resync

Name : centos8:myraid1 (local to host centos8)
UUID : b5d2f62e:5004a01b:e1fb91a7:ed707989
Events : 19

Number Major Minor RaidDevice State
-      0      0      0      removed
1      8      17      1      active sync  /dev/sdb1
[user@centos8 ~]$
```

Напоследок, давайте избавимся от одного диска и посмотрим, что же произойдёт. Для этого я выключаю виртуальную машину и удаляю один из дисков. Далее запускаю систему. После запуска системы, я проверяю, примонтирована ли файловая система - `df -h /mydata` - да, с файловой системой всё okay. Но если посмотреть статус рейда - `sudo mdadm -D /dev/md127` - то можно увидеть, что статус рейда – `degraded` – а это значит, что есть какие-то проблемы с рейдом, он не в полноценном состоянии. А внизу видно - один из дисков `removed`.

```
[user@centos8 ~]$ sudo fdisk -l /dev/sdb1
Disk /dev/sdb1: 954 MiB, 1000341504 bytes, 1953792 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
[user@centos8 ~]$ sudo mdadm /dev/md127 --add /dev/sdb1
mdadm: added /dev/sdb1
[user@centos8 ~]$ █
```

Теперь давайте представим, что мы купили новый диск, на замену старому. Опять вырубая виртуальную машину, создаю ещё один диск – тоже на гигабайт. Хотя тут скорее особенность виртуалбокса, не получится при работе виртуалки добавлять диски. На физических серверах необходимости выключать сервер для добавления диска нет. Для начала определим, какое название получил диск. Для этого выполним команду `sudo fdisk -l` и найдём чистый диск. Теперь нужно создать раздел, как мы это уже делали - `sudo fdisk /dev/sdb; g, n,enter,enter,+1GB, p, w` - и добавить его в рейд с помощью ключа `add` - `sudo mdadm /dev/md127 --add /dev/sdb1`.

```
Raid Devices : 2
Total Devices : 2
Persistence : Superblock is persistent

Update Time : Sun Mar 28 19:28:42 2021
State : clean
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0

Consistency Policy : resync

Name : centos8:myraid1 (local to host centos8)
UUID : b5d2f62e:5004a01b:elfb91a7:ed707989
Events : 42
```

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

После добавления статус нового диска поменяется на `spare rebuilding` - `sudo mdadm -D /dev/md127`. При этом мы можем продолжать работать. Так как у нас информации на дисках не было, процесс прошёл довольно быстро.

И так, мы с вами разобрали, как на Linux настроить программный рейд. Выход из строя диска может создать кучу проблем для администратора, не говоря о других. Это не вопрос вероятности, это вопрос времени. При этом предотвратить один из худших кошмаров можно всего за пару минут, настроив, не важно, программный или аппаратный рейд. И хотя мы рассмотрели только первый рейд, в настройке других уровней рейдов больших отличий нет. Как и в большинстве случаев, за парой простых команд скрывается тонна теории, которую важно знать, потому что неправильное обращение с командами может привести к печальным последствиям. По [ссылке](#) вы можете более подробно ознакомиться с программный рейдом на Linux.