

Unix & Linux Stack Exchange is a question and answer site for users of Linux, FreeBSD and other Un*x-like operating systems. Join them; it only takes a minute:

Here's how it works:

Sign up

Anybody can ask a question

Anybody can answer

The best answers are voted up and rise to the top

Difference between UUID from blkid and mdadm?

Can someone explain the difference between the UUID's reported by `blkid` and `mdadm` ? On one of our CentOS systems, for example:

```
[root@server ~]# blkid | grep /dev/md1
/dev/md1: UUID="32cb0a6e-8148-44e9-909d-5b23df045bd1" TYPE="ext4"
```

```
[root@server ~]# mdadm --detail /dev/md1 | grep UUID
UUID : f204c558:babf732d:85bd7296:bbfebeea
```

Why are they different and how would we change the UUID used by `mdadm` ?

I understand we would use `tune2fs` to change the UUID for the partition (which would change what is returned by `blkid`) but not sure how to change what `mdadm` uses.

/ linux / mdadm / uuid

asked May 15 '14 at 11:29



Chris

321 2 4 16

3 Answers

The first one reports the UUID of the ext4 filesystem on the `md` block device. It helps the system identify the file system uniquely among the filesystems available on the system. That is stored in the structure of the filesystem, that is in the data stored on the md device.

The second one is the UUID of the RAID device. It helps the md subsystem identify that particular RAID device uniquely. In particular, it helps identify all the block devices that belong to the RAID array. It is stored in the metadata of the array (on each member). Array members also have their own UUID (in the md system, they may also have partition UUIDs if they are GPT partitions (which itself would be stored in the GPT partition table), or LVM volumes...).

`blkid` is a bit misleading, as what it returns is the ID of the structure stored on the device (for those kind of structures it knows about like most filesystems, LVM members and swap devices). Also note that it's not uncommon to have block devices with structures with identical UUIDs (for instance LVM snapshots). And a block device can contain anything, including things whose structure doesn't include a UUID.

So, as an example, you could have a system with 3 drives, with GPT partitioning. Those drives could have a [World Wide Name](#) which identifies it uniquely. Let's say the 3 drives are partitioned with one partition each (`/dev/sd[abc]1`). Each partition will have a GPT UUID stored in the GPT partition table.

If those partitions make up a md RAID5 array. Each will get a md UUID as a RAID member, and the array will get a UUID as md RAID device.

That `/dev/md0` can be further partitioned with MSDOS or GPT-type partitioning. For instance, we could have a `/dev/md0p1` partition with a GPT UUID (stored in the GPT partition table that is stored in the data of `/dev/md0`).

That could in turn be a physical volume for LVM. As such it will get a PV UUID. The volume group will also have a VG UUID.

In that volume group, you would create logical volumes, each getting a LV UUID.

On one of those LVs (like `/dev/VG/LV`), you could make an ext4 filesystem. That filesystem would get an ext4 UUID.

`blkid /dev/VG/LV` would get you the (ext4) UUID of that filesystem. But as a partition inside the VG volume, it would also get a partition UUID (some partitioning scheme like MSDOS/MBR don't have UUIDs). That volume group is made of members PVs which are themselves other block devices. `blkid /dev/md0p1` would give you the PV UUID. It also has a partition UUID in the GPT table on `/dev/md0` . `/dev/md0` itself is made off other block devices. `blkid /dev/sda1`

will return the raid-member UUID. It also has a partition UUID in the GPT table on `/dev/sda`.

edited Sep 9 '15 at 7:06

answered May 15 '14 at 11:37



Stéphane Chazelas

194k 36 332 563

How would we change the UUID of a RAID device as used by `mdadm`? We just re-imaged a server and the UUID's are different, so we want to restore the previous UUID's so that we don't need to change all the config files. Essentially, `/dev/md0` has a new UUID and we want to return it to the old one (identified from a backup) so the system boots without needing further changes. – Chris May 15 '14 at 12:37

@Chris, The MD UUID (see `man mdadm`) or the ext4 UUID (see `man tune2fs`)? Is the boot problem about finding the root filesystem or about assembling the RAID array based on the `mdadm.conf` stored in the `initramfs`. Either way, it sounds simpler to me to update the `mdadm.conf` or `fstab/grub.cfg` than to mangle with UUIDs. – Stéphane Chazelas May 15 '14 at 12:51

The MD UUID - we restored the server from backup. Do you know how to update the UUID of a RAID device? I can't see that information in the manpage. We're just trying to work out the simplest way to get it working (testing the restore procedure). – Chris May 15 '14 at 13:06

@Chris, you should raise another question for that. – Stéphane Chazelas May 15 '14 at 13:27

@Chris: You want the UUIDs to be different. If `mdadm` sees multiple disks believing to belong to the same raid even though it's different raids (because of UUID conflicts), you will have real problems. – frostschutz May 15 '14 at 15:30



The different UUID was explained already. Not only filesystems have them. There just are UUIDs for different things: raid array, device, partition, LUKS containers, LVM PV's... and finally filesystems.

What annoys me personally is that even the way those UUIDs are formatted is different.

`blkid`:

```
# blkid /dev/sda1
/dev/sda1:
UUID="d8b8b4e5-e47b-2e45-2093-cd36f654020d"
UUID_SUB="3c3e6eac-2139-3f7a-16b7-57280934d88e"
PARTUUID="6a89cedf-69e1-40db-b08c-1c8e45af59f5"
```

`mdadm`:

```
# mdadm --examine /dev/sda1 | grep UUID
Array UUID : d8b8b4e5:e47b2e45:2093cd36:f654020d
Device UUID : 3c3e6eac:21393f7a:16b75728:0934d88e
```

As you can see, they are the same UUIDs, but `blkid` prints them with dashes - whereas

`mdadm` uses colons : . So you get `d8b8b4e5-e47b-2e45-2093-cd36f654020d` VS.

`d8b8b4e5:e47b2e45:2093cd36:f654020d` .

Very annoying, especially if you want to work with UUIDs in scripts. It is not obvious how to convert from one formatting to the other.

answered May 15 '14 at 15:22



frostschutz

17.8k 1 30 48

What about `sl:/-g` or vice versa? ;) – ThiefMaster Dec 3 '14 at 22:08

really the colon : or dash - are there simply so humans can read the long UUIDs in chunks and not get lost somewhere in the middle. The UUID is really a binary number, and has no dashes or colons. It is implemented in Linux as an unsigned character array: `unsigned char uuid_out[16]`. Apparently the UUID spec also has an idea of variants and versions, which we can see implemented by setting the most significant byte `uuid_out[6]` to 4. – TrinitronX Sep 9 '15 at 8:29

The `blkid` UUID above "32cb0a6e-8148-44e9-909d-5b23df045bd1" is the correct one, that is what the OS will use to find the RAID array.

`mdadm` has it's own "internal" UUID which is not used directly by the OS and is what you use in the `mdadm.conf` file eg:

"ARRAY /dev/md1 level=raid1 num-devices=2 uuid=f204c558:babf732d:85bd7296:bbfebeea"

`mdadm` shouldn't call anything UUID when it is separate from the one that `blkid` and the OS recognize. Maybe it should be called `mduuid` or something else to avoid confusion.

answered May 15 '14 at 11:38



cioby23

1,390 4 9
