

RAID AVEC CONTRÔLE DE L'INTÉGRITÉ DES DONNÉES

REV 2024-08-29

Dans cet article, voici comment j'ai installé un système de fichiers avec une double sécurité:

1. couche de protection en cas de panne physique d'un disque avec le RAID1 (https://en.wikipedia.org/wiki/Standard_RAID_levels#RAID_1)
2. couche de protection contre le "bitrot" avec integritysetup (https://en.wikipedia.org/wiki/Data_degradation)
3. utilisation d'un système de fichier classique (ici ext4)

Je sais que tout cela peut se faire uniquement avec BTRFS, mais j'ai eu quelques surprises de problème d'intégrité en l'utilisant comme stockage de disques de machines virtuelles.

Les disques physiques seront sdb et sdc, le raid md0 et le volume final data.

0.1 Préparation des disques

Pour commencer, petit nettoyage des disques pour effacer les traces des précédentes partitions et systèmes de fichiers. Pour cela, j'utilise wipefs:

```
wipefs --all /dev/sdb /dev/sdc
```

0.2 Création du RAID1

Création du volume md0, en raid de niveau 1, avec 2 disques (sdb et sdc)

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb /dev/sdc
```

On peut ensuite vérifier les informations du raid:

```
mdadm --query --detail /dev/md0
```

Pour que ce volume RAID1 soit activé au démarrage, il faut ajouter sa configuration à /etc/mdadm/mdadm.conf. Cette ligne de configuration est obtenue par

```
mdadm --detail --scan /dev/md0
```

Par exemple: ARRAY /dev/md0 metadata=1.2 name=hostname:0 UUID=12345678:abcdefgh:12345678:

0.3 Création du journal des sommes d'intégrité

Integritysetup fait parti du projet "cryptsetup" (<https://gitlab.com/cryptsetup/cryptsetup>). La création de ce volume se fait avec la commande format puis la mise à disposition se fait avec open.

```
integritysetup format /dev/md0  
integritysetup open /dev/md0 data
```

Tout comme pour le RAID, il faut informer le système de l'existence de ce volume. Cette fois, c'est dans le fichier `/etc/integritytab` (<https://manpages.debian.org/bookworm/systemd/integritytab.5.en.html>).

Le format est `volume-name block-device`, par exemple:

```
data /dev/md0
```

0.4 Création du système de fichiers

Rien d'exceptionnel ici, par exemple pour un système de fichiers ext4:

```
mkfs.ext4 -L data /dev/mapper/data
```

Comme c'est du RAID1, pas d'histoire de *stripping* à configurer à la création du système de fichier.

Il ne reste qu'à ajouter le point de montage à `/etc/fstab`.

TAGS: GNU/Linux, RAID