

Convertir en configuration Raid1 sur le système Centos existant

Le **RAID** est une technologie qui combine plusieurs disques durs en un seul disque dur logique. Il existe différents types de niveaux **RAID** disponibles, dans la mesure où **RAID1** utilise deux disques durs et conserve des copies identiques des données. Il est également appelé Raid Mirroring. Ainsi, en utilisant **RAID1**, si un disque dur tombe en panne, le système continuera à fonctionner à partir du deuxième disque car les deux disques conservent exactement les mêmes données. L'un des inconvénients du **RAID 1** est que si on dispose de deux disques de 500 Go, l'espace disque total qu'on peut utiliser après la configuration du **RAID1** est toujours de 500 Go. En bref, on ne pourra utiliser que 50 % de l'espace disque total disponible.

On peut réaliser une configuration **RAID** soit au niveau matériel, soit via le logiciel. Le raid matériel est effectué via un périphérique contrôleur **RAID** (carte PCIe). Le **RAID** logiciel est effectué au niveau du système d'exploitation. Le système d'exploitation Linux prend en charge le **RAID** logiciel et peut configurer le **RAID** au moment de l'installation du système d'exploitation.

A titre d'exemple on va supposer qu'on a déjà un système d'exploitation **Centos7** fonctionnel sans **RAID** et on veut connecter un disque supplémentaire et convertir cette installation Centos7 existante en **RAID1**. On va donc voir comment convertir le système **CentOS 7** à disque unique existant en un système **RAID1** à deux disques sans perdre de données ni réinstaller le système. **RAID 1** produit un ensemble en miroir, qui peut tolérer une seule panne de disque. Le chargeur de démarrage **GRUB2** est également configuré de telle manière que le système pourra toujours démarrer en cas de panne de l'un des disques durs.

Le système existant a les caractéristiques suivantes:

```
Operating System : Centos 7.4

# Hard Disk partitions
2GB /boot partition as /dev/sda1
16GB / root partition as /dev/sda2
2GB swap space in /dev/sda3
Partition Type : ext4
```

On ajoute un deuxième disque identique /dev/sdb pour la configuration du **raid 1**. Le **raid** sera un **raid logiciel** Linux géré par **mdadm**. Si la commande **mdadm** n'est pas disponible, il faut l'installer:

```
yum install mdadm
```

Créer un schéma de partition identique à celui du disque actuel /dev/sda

```
sfdisk -d /dev/sda | sfdisk /dev/sdb
```

Vérifier les modifications à l'aide de la commande ci-dessous.

```
fdisk -l
```

Modifier le type de partition en « **Détection automatique du raid Linux** », en utilisant **fdisk** pour convertir les 3 partitions du disque dur en "fd".

```
fdisk /dev/sdb
```

Vérifier le résultat à l'aide de la commande ci-dessous.

```
fdisk -l
```

Maintenant créer un **RAID1** dégradé. Ce qui signifie qu'on va créer des périphériques **RAID1** dans un état dégradé car un disque est manquant. Le /dev/sda qui est actuellement installé avec le système d'exploitation et utilisé pour démarrer.

Dans le cas étudié, on a trois partitions. Il faut donc le faire pour toutes les partitions du nouveau disque /dev/sdb:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 missing /dev/sdb1  
mdadm --create /dev/md1 --level=1 --raid-devices=2 missing /dev/sdb2  
mdadm --create /dev/md2 --level=1 --raid-devices=2 missing /dev/sdb3
```

Vérifier le résultat avec :

```
cat /proc/mdstat
```

Créer les systèmes de fichiers sur les nouvelles partitions des périphériques **RAID**.

```
mkfs.ext4 /dev/md0  
mkfs.ext4 /dev/md1  
mkswap /dev/md2
```

Il faut maintenant répliquer manuellement les données existantes sur les partitions /dev/sda existantes vers de nouvelles partitions **RAID** logicielles. Pour cela, monter / et /boot:

```
mount /dev/md0 /mnt/boot/  
mount /dev/md1 /mnt/
```

Copier les données existantes à l'aide de la commande **rsync**:

```
rsync -auxHAXSv --exclude=/dev/* --exclude=/proc/* --exclude=/sys/* --  
exclude=/tmp/* --exclude=/mnt/* --exclude=/run/* /* /mnt
```

Monter maintenant les dossiers système:

```
mount --bind /proc /mnt/proc
mount --bind /dev /mnt/dev
monter --bind /sys /mnt/sys
mount --bind /run /mnt/run
```

Chrooter dans la partition /mnt pour le disque /dev/sdb:

```
chroot /mnt/
```

Modifier fstab avec les informations UUID du nouveau lecteur:

```
blkid /dev/md*

/dev/md0 : UUID="bdf72caa-716c-431d-ab23-4699feb14bdf" TYPE="ext4"
/dev/md1 : UUID="cef39ac1-e53b-4a25-9608-7baf20fade02" TYPE="ext4"
/dev/md2 : UUID="45b6e2b2-cd79-4fdd-8288-80e1db0da874" TYPE="échange"
```

Maintenant, ouvrir le fichier /etc/fstab et ajouter les entrées comme ci-dessous:

```
vim /etc/fstab
UUID=new-UUID / ext4 defaults 1 1
UUID=new-UUID /boot ext4 defaults 1 2
UUID=new-UUID swap swap defaults 0 0
```



Remplacer **new-UUID** par l'UUID de partition de disque.

Créer maintenant un fichier mdadm.conf à partir de la configuration RAID actuelle :

```
mdadm --detail --scan > /etc/mdadm.conf
```

Sauvegarder, mettre à jour l'**initrd** actuel et reconstruire l'**initramfs** avec le nouveau mdadm.conf:

```
cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bck
dracut --mdadmconf --fstab --add="mdraid" --filesystems "xfs ext4 ext3" --
add-drivers="raid1" --force
/boot/initramfs-$(uname -r).img $(uname -r) -M
```

Modifier maintenant **grub** et ajouter quelques paramètres par défaut à **grub**:

```
vim /etc/default/grub
GRUB_CMDLINE_LINUX="crashkernel=auto rd.auto rd.auto=1 rhgb quiet"
```

```
GRUB_PRELOAD_MODULES="mdraid1x"
```

Créer une nouvelle configuration **Grub**:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

Installer grub sur le nouveau disque /dev/sdb:

```
rub2-install /dev/sdb
```

À ce stade, redémarrer le système en choisissant le nouveau disque /dev/sdb dans le BIOS. Si tout fonctionne, le système démarrera avec le nouveau disque /dev/sdb. Après cela, vérifier les points de montage et l'état du raid à l'aide des commandes ci-dessous.

```
swapon -s  
mount -t ext4  
cat /proc/mdstat
```



Dans le cas étudié, on a changé les ports du disque afin qu'après le redémarrage, le nouveau disque devienne /dev/sda et l'ancien disque devienne /dev/sdb. Il ne faut pas se tromper avec les résultats de la sortie d'écran, si on choisi de démarrer à partir d'un nouveau disque à partir du BIOS au lieu de l'échange de port de disque, dans ce cas, les résultats de la commande **mdstat** afficheront le nouveau disque sous le nom /dev/sdb lui-même.

Il faut maintenant ajouter l'ancien disque à la matrice **RAID**. Il faut changer le type de partition en « **Détection automatique du raid Linux** » pour l'ancien disque en utilisant la commande **fdisk** pour convertir les 3 partitions en "fd" (dans notre cas, c'est /dev/sdb)

```
fdisk /dev/sdb
```

Ajouter maintenant l'ancien disque à la matrice **RAID 1** (Dans notre cas, c'est /dev/sdb):

```
mdadm --manage /dev/md0 --add /dev/sdb1  
mdadm --manage /dev/md1 --add /dev/sdb2  
mdadm --manage /dev/md2 --add /dev/sdb3
```

Vérifier l'état de la reconstruction à l'aide de la commande ci-dessous et vérifier si elle est en cours d'exécution.

```
watch -n1 "cat /proc/mdstat"
```

Réinstaller Grub sur l'ancien disque (dans notre cas, c'est /dev/sdb)

```
grub2-istall /dev/sdb
```

Ceci conclu la configuration de **raid1** sur l'installation **centos 7.4** existante. Une fois la reconstruction terminée, **mdstat** affichera le résultat et à partir de là, on pourra confirmer qu'il est terminé et fonctionne correctement. Redémarrer à nouveau.

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=howto:convert-single-disk-to-raid1>

Last update: **2025/02/19 10:59**

