

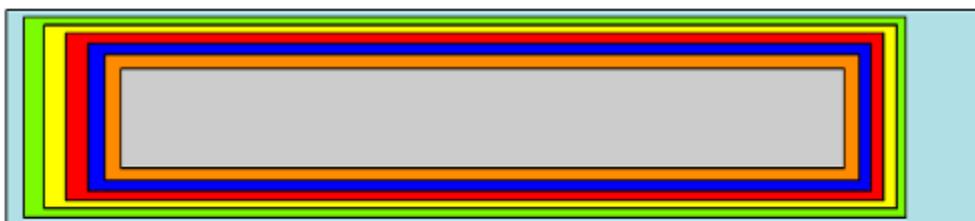
Augmentation de taille LVM avec PV sur partition par ajout de PV

La méthode de PV sur partition est utilisée à chaque fois pour les volumes du système d'exploitation notamment parce qu'il y a souvent au moins une partition primaire simple dédié à /boot et parfois une pour le Swap si elle n'est pas dans le LVM.

Pour augmenter un LVM avec PV sur partition on dispose de deux méthodes qui sont utilisables ou pas à chaud en fonction de partprobe. Cf. description dans [generalites_lvm](#).

Nous détaillons ici la méthode où on crée une partition de type LVM sur l'espace libre ainsi qu'un PV et on étend le groupe de volume à ce nouveau PV.

Situation initiale.



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



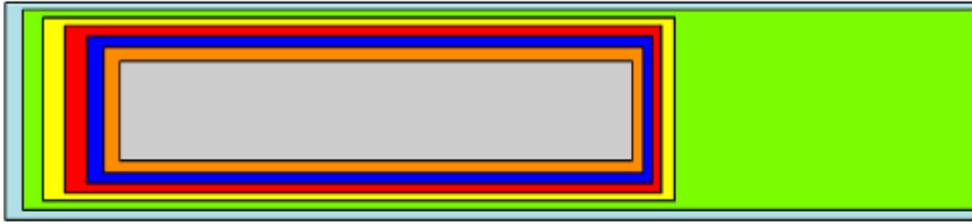
orange foncé : volume logique



gris : système de fichiers



Augmenter la taille avec l'outil de gestion des disques du SAN.



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Se rendre sur le système Linux et soit rebooter soit rescanner à chaud le bus SCSI correspondant au disque pour que le système prenne en compte la modification de taille qui a été réalisée sur l'outil de gestion du SAN (Cf. [operations_volumes_scsi](#)).

Liste des étapes

- Ajout d'une nouvelle partition LVM (8e) à coté de l'existante avec fdisk.
- Création d'un PV sur cette partition LVM.
- Ajout de ce volume dans le VG.
- Augmenter le LV et le FS.

Étapes

Ajout d'une nouvelle partition LVM.

```
fdisk /dev/sdb  
n pour ajouter une partition.  
p pour partition primaire.  
2 numéro de la partition (il y a déjà une partition numéro 1).  
Secteur de début par défaut (taper entrée).  
Secteur de fin par défaut (taper entrée).
```

Ainsi on utilise la totalité du disque virtuel.

p pour afficher la table de partition (ici on voit la partition que l'on a créé avec l'existante).

Par défaut la partition est créée avec un type Linux ext3 (numéro 83) mais ici nous voulons utiliser les LVM.

Nous allons donc changer le tag de la partition.

t pour type de partition.

2 pour sélectionner la seconde partition présente (il y a déjà une partition numéro 1).

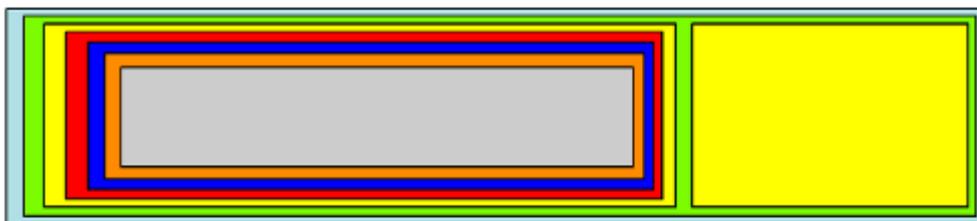
L pour afficher la liste.

Entrer le numéro : 8e pour Linux LVM.

Vérifier avec un p.

w pour exécuter la création de la partition.

Il faut obligatoirement rebooter (sauf si [partprobe](#) fonctionne pour vous) pour réinitialiser la table de partition sinon le noyau utilise toujours l'ancienne table de partition et on ne peut utiliser la nouvelle partition créée.



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Nous devons ensuite l'initialiser pour pouvoir le prendre en compte en LVM.

```
pvcreate /dev/sdb2
```

Pour vérifier.

```
pvs
```

```
pvscan  
pvdisk /dev/sdb2
```



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Nous devons ensuite ajouter l'espace nouvellement défini (partition LVM numéro 2) au groupe de volume initial.

```
vgextend datavg /dev/sdb2
```

La commande inverse.

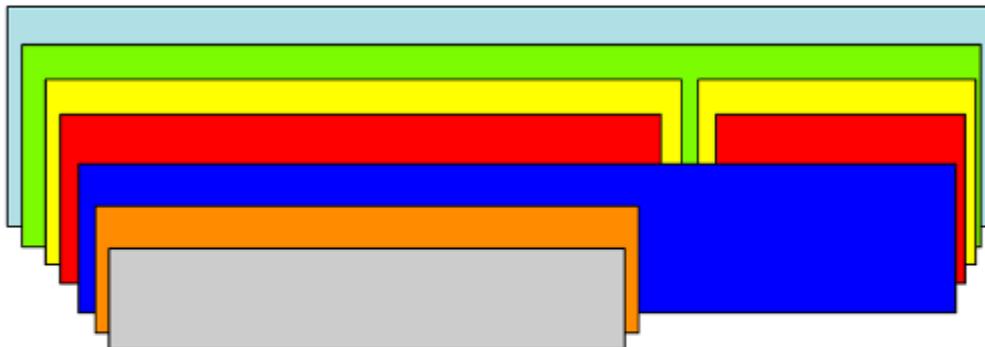
```
vgreduce datavg /dev/sdb2
```

Pour vérifier.

```
pvdisk /dev/sdb2  
vgdisplay -v datavg -D (-D pour voir les volumes non-actifs)  
vgs
```

Vérifier la totalité des VG et voir les deux PV assignés au VG.

```
vgdisplay -v
```



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Nous avons pris l'espace en considération dans le VG, il nous faut maintenant augmenter la taille du volume logique (ici de 3 Go).

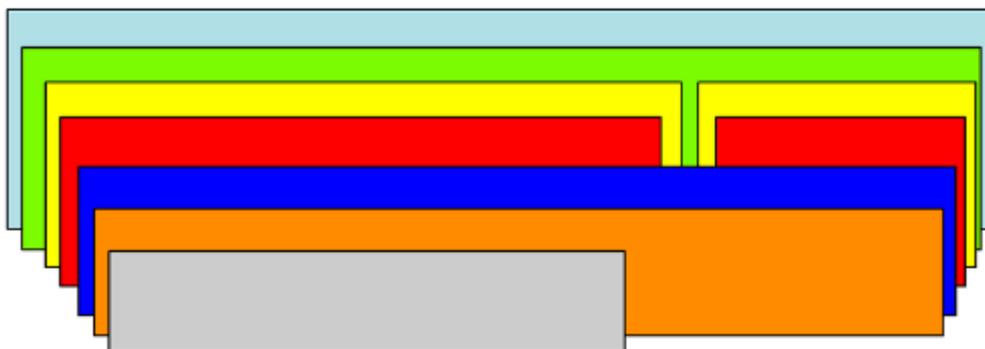
Prendre la valeur Free PE de la commande `vgdisplay`, ce sont les Physical Extent de libre. Avec cette valeur, on remplira tout l'espace du volume sans en perdre.

```
lvextend -l +NOMBRE_PE cheminlv devicepv
```

```
lvextend -L +700 /dev/datavg/data1v /dev/sdb2  
ou  
lvextend -L +3G /dev/datavg/data1v /dev/sdb2
```

La commande inverse pour information.

```
lvreduce -l 700 /dev/datavg/data1v
```



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Enfin, nous devons agrandir le système de fichiers afin de pouvoir accéder à un plus grand espace de stockage.

Faire une vérification du système de fichiers uniquement si on peut démonter le volume car cette opération se fait à froid.

```
e2fsck -f /dev/datavg/data1v
```

Lancer l'opération d'extension du système de fichiers (fonctionne avec ext3 ou ext4 sans problème).

```
resize2fs /dev/datavg/data1v
```

Par défaut, l'augmentation va jusqu'au bout de la partition.



bleu poudre : grappe de disques



vert pelouse : disque virtuel



jaune : partition LVM



rouge : volume physique



bleu : groupe de volume



orange foncé : volume logique



gris : système de fichiers



Les mêmes opérations précédentes en une seule commande. L'augmentation du lv et du file system en même temps pour attribuer 50% de l'espace libre disponible.

```
lvextend -r -l +50%FREE /dev/datavg/data1v
lvresize -r -l +50%FREE /dev/datavg/data1v
lvdisplay
lvs
```

Remonter le volume si il a été démonté.

```
mount /dev/datavg/data1v /data
ou
mount -a #pour monter ce qui n'est pas monté dans /etc/fstab
```

La partition est de nouveau opérationnelle.

L'opération est terminée.

From:
<https://wiki.ouieuhtoutca.eu/> - **kilsufi de noter**

Permanent link:
https://wiki.ouieuhtoutca.eu/augmentation_volume_lvm_sur_partition_ajoutpv

Last update: **2021/01/21 21:42**

