
lvconvert

Convertis un volume logique lineaire vers un miroir ou snapshot

lvconvert est utilisé pour changer le type de segment ou les caractéristiques d'un volume logique. Par exemple, il peut ajouter ou supprimer les images redondantes d'un volume logique, changer le type de log d'un miroir, ou désigner un volume logique comme dépôt de snapshot.

Si la conversion nécessite l'allocation d'extents physique, l'allocation sera restreinte à ces extents physique. Si la conversion libère des extents physiques et qu'un ou plusieurs PV sont spécifiés, les extents physiques sont d'abord libérés depuis ces PV.

opérations

Une seule de ces option est requise :

- cache, -H, -type cache** Convertis un volume logique en LV cache avec l'utilisation d'un pool cache spécifié avec `-cachepool`.
- corelog** identique à `-mirrlog core`
- merge** Fusionne un snapshot dans son volume d'origine ou fusionne une image raid1 (qui a été splittée de son miroir avec `-trackchanges`) dans son miroir. Si les 2 volumes ne sont pas ouverts, la fusionne démarre immédiatement. Fusionner un snapshot dans un origine qui ne peut pas être fermé et différé à la prochaine activation du volume. Durant la fusion, les lectures et écritures de l'origine apparaissent comme s'ils étaient dirigés dans le snapshot à fusionner. Une fois terminé, le snapshot est supprimé. Plusieurs snapshots peuvent être spécifiés sur la ligne de commande ou un `@tag` peut être utilisé pour spécifier plusieurs snapshots.
- mirrlog {diskcore|mirrored}** Spécifie le type de log à utiliser. Défaut : `disk`, qui est persistant est nécessite peut d'espace disque, généralement sur un disque séparé. `core` peut être utile pour les miroirs à durée de vie courte : il signifie que le miroir est régénéré en copiant les données depuis le premier périphérique à chaque fois que le périphérique est activé. `mirrored` crée un log persistant qui est lui-même miroiré.
- mirrors** Spécifie le degré du miroir à créer. Par exemple `"-m 1"` convertis le volume logique d'origine en un miroir avec 2 parties ; c'est à dire un volume linéaire avec une copie. Il y a 2 implémentations miroir qui correspondent aux type de segment `mirrlog` et `raid1`. le type par défaut est `raid1`.
- repair** Répare un miroir après une erreur disque et tente de fixer les métadonnées de pool thin. Par défaut, le nombre de miroir d'origine sera restauré si possible. `-y` permet de répondre automatiquement aux questions. `-f` permet de ne rien remplacer. `-use-policies` utilise la stratégie de remplacement spécifiée dans `lvm.conf`. La réparation des pools thin ne peut se faire que sur les volume thin inactifs. Il n'y a pas de validation des métadonnées entre le kernel et `lvm2`, et nécessite un travail manuel ultérieur. Une fois réparé, les anciennes métadonnées sont disponibles dans le LV `"<pool>_meta<n>"`.
- replace** Supprime le PV spécifié et le remplace avec un disponible dans le VG ou depuis la liste fournie. Seulement disponible pour les types de segment `raid`.
- snapshot** Recrée un snapshot de volumes logique après avoir été séparés en utilisant `-splitsnapshot`.
- split** Sépare le volume logique spécifié.
- splitcache** Sépare le volume logique cache du pool de cache. Avant qu'il soit sortie du cache, le cache est vidé. Le volume est ainsi laissé inutilisé et peut être utilisé pour cacher un autre volume.
- splitsnapshot** Sépare le volume logique snapshot spécifié de son origine. Le volume splitté contient les chunks qui diffèrent de l'origine avec les métadonnées les décrivant. Ce volume peut être effacé et détruit avec `lvremove`. l'inverse de `-snapshot`
- splitmirrors** Spécifie le nombre d'images redondantes d'un miroir à splitter pour former un nouveau volume logique. Un nom doit être fournis pour le nouveau LV avec `-name`, sauf si `-trackchanges` est spécifié
- T, -thin, -type thin** Converti le volume logique en volume logique thin du pool thin spécifié avec `-thinpool`. Le volume logique original est renommé en un nouveau volume logique lecture-seule. utiliser `-originname` pour spécifier un nom. Le volume ne peut plus être modifié tant qu'il est utilisé comme volume d'origine externe pour les zones non-provisionnées d'un volume logique thin.

- type SegmentType** Utilisé pour convertir un volume logique en un autre type de segment (cache, cache-pool, raid1, snapshot, thin, ou thin-pool). En convertissant un volume logique en LV cache, l'argument –cachepool est requis. En convertissant un volume logique en LV thin, –thinpool est requis.
- uncache** décache le volume logique cache spécifié. Avant que le volume devienne non-caché, le cache est vidé. à la différence avec –splitcache le volume pool cache est supprimé. Cette option est l'inverse de –cahe.

OPTIONS

- b, –background** Lance en tâche de fond
- cachepolicy Policy** Définis la stratégie de cache. mq est le nom de la stratégie de base. smq est plus avancé.
- cachepool CachePoolLogicalVolume{Name|Path}** Cet argument est nécessaire en convertissant un volume logique en LV cache.
- cachesettings Key=Value** Définis les paramètres de cache.
- c, –chunksize ChunkSize [b|B|s|K|k|M|G]** Donne la taille de chunk pour les volumes logique snapshot, pool cache et pool thin. Pour les snapshots, la valeur doit être en puissance de 2 entre 4Kib et 512Kib. Défaut : 4. pour les cache, la valeur doit être entre 32KiB et 1GiB, défaut : 64. pour les pools thin, la valeur doit être entre 64KiB et 1GiB.
- discards {ignore|nopassdown|passdown}** Traitement des annulations par la couche thin dans le kernel et passé au volume physique.
- i, –interval <seconds>** Interval d'affichage de la progression en %
- name** Le nom à appliquer à un volume logique qui a été splitté depuis un volume logique miroir
- noudevsync** Désactive la synchronisation udev. Le processus n'attend pas de notification udev.
- oniginname NewExternalOriginVolumeName** Le nouveau nom pour le volume logique original, qui devient le volume origine externe pour un volume logique thin. Sans cette option, le nom par défaut est "lvol<n>" où <n> est le numéro LVM interne du volume logique. Ce volume sera lecture seule est ne peut plus être modifié tant qu'il est utilisé comme origine externe.
- poolmetadata PoolMetadataLogicalVolume{Name|Path}** Spécifie les métadonnées cache ou thin. La taille devrait être entre 2Mio et 16Gio. Le pool cache est spécifié avec l'option –cachepool. Le pool thin est spécifié avec l'option –thinpool. Si le pool spécifié existe déjà, les métadonnée du pool seront placées dans le volume logique. Les propriétés du pool sont préservées par défaut. Il peut être utile pour réparer les métadonnées de pool ou pour redimensionner offline.
- poolmetadatasize PoolMetadataSize [b|B|s|K|k|M|G]** Définis la taille des métadonnées du pool thin ou cache. La valeur par défaut est estimée avec (Pool_LV_size / Pool_LV_chunk_size addentry articles autoadd autofind autoprod createalpha createbeta createdb createprod findentry fullpowa generator.php genhtml genman genmd gentex html insert man md pdf regen setfor setfor2 sql temp temp-new-pc tex threads ToDo 64bits).
- poolmetadataspare {y|n}** Contrôle la création et la maintenance de métadonnées de volume logique spare utilisé pour la récupération automatique de pool. Seul un tel volume est maintenu dans un volume group avec une taille des plus grosses métadonnées de volume du pool. Défaut : yes.
- r, –readahead {ReadAheadSectors|auto|none}** Définis le compteur de secteurs de readhead des métadonnées de volume logique du pool thin. Défaut : auto.
- R, –regionsize MirrorLogRegionSize** Un miroir est divisé en régions de cette taille (en Mo), et le log miroir utilise cette granularité pour tracer quelles région sont en synchro
- stripes** Indique le nombre de stripes. C'est égale au nombre de volumes physiques pour disperser le volume logique. Ne peut pas s'appliquer à un espace déjà alloué.
- I, –stripesize** Indique le nombre de Ko pour la granularité des stripes. La tailles doit être une puissance de 2 mais ne peut excéder la taille d'extent physique.
- thinpool ThinPoolLogicalVolume{Name|Path}** Spécifie ou convertis un volume logique en un volume de données de pool thin. Le contenu du volume convertis est perdu. Les métadonnées du volume logique du pool thin peuvent être spécifiés avec l'option –poolmetadata ou alloués avec –poolmetadatasize.
- trackchanegs** Avec –splitmirrors dans un raid1, suit les changements pour que l'image lecture seul détachée peut être fusionnée efficacement de nouveau dans le miroir. Seul les régions du périphérique détaché où les données ont changés seront resynchronisés.
- Z, –zero {y|n}** Contrôle les 0 des premiers 4Kio de données dans le snapshot. Si le volume est lecture seul, le snapshot ne sera pas mis à 0. Pour les volume thin il contrôle les 0 des blocks provisionnés. Noter que provisionner un grand nombre de chunks à 0 impacte les performances.

Exemples

Convertir le volume logique linéaire "vg00/lvol1" en volume logique miroir :

```
lvconvert -m1 vg00/lvol1
```

Convertir le volume logique linéaire "vg00/lvol1" en volume logique RAID1 :

```
lvconvert --type raid1 -m1 vg00/lvol1
```

Convertir un miroir avec un log disque en un miroir avec un log en mémoire :

```
lvconvert --mirrorlog core vg00/lvol1
```

Convertir un miroir avec un log en mémoire en un miroir avec un log disque :

```
lvconvert --mirrorlog disk vg00/lvol1
```

Convertir un volume logique miroir en un volume logique linéaire :

```
lvconvert -m0 vg00/lvol1
```

Convertir un volume logique miroir en un volume logique RAID1 avec le même nombre d'images :

```
lvconvert --type raid1 vg00/mirror_lv
```

Convertir le volume logique "vg00/lvol2" en snapshot du volume original "vg00/lvol1" :

```
lvconvert -s vg00/lvol1 vg00/lvol2
```

Convertir un volume logique linéaire "vg00/lvol1" en miroir, en utilisant des extents physiques /dev/sda :0-15 et /dev/sdb :0-15 pour l'allocation des nouveaux extents :

```
lvconvert -m1 vg00/lvol1 /dev/sda :0-15 /dev/sdb :0-15
```

Convertir le volume logique miroir "vg00/lvmirror1" en linéaire, libérant les extents physiques de /dev/sda :

```
lvconvert -m0 vg00/lvmirror1 /dev/sda
```

Fusionner "vg00/lvol1_snap" dans son origine :

```
lvconvert --merge vg00/lvol1_snap
```

Si "vg00/lvol1", "vg00/lvol2" et "vg00/lvol3" sont tous taggés avec "some_tag" chaque volume logique snapshot sera fusionné en série. Si `--background` est utilisé il démarre la fusion des snapshots en parallèle :

```
lvconvert --merge @some_tag
```

Extraire une image du miroir, créant un nouveau volume logique "lv_split". Le miroir d'où l'image est extraite est réduite en accord. S'il y avait un miroir 2-way (créé avec `-m 1`), le volume original résultant sera linéaire :

```
lvconvert --splitmirrors 1 --name lv_split vg00/lvmirror1
```

Un volume logique créé avec `--type raid1` peut utiliser `--trackchanges` en splittant une image. Détacher une image du lv miroir `lv_raid1` comme périphérique lecture seul et suivre les changements :

```
lvconvert --splitmirrors 1 --trackchanges vg00/lv_raid1
```

Fusionner une image qui a été détachée temporairement de son miroir avec `--trackchanges` dans son miroir d'origine :

```
lvconvert --merge vg00/lv_raid1_rimage_1
```

Remplacer le volume physique /dev/sdb1 dans le volume logique RAID1 `my_raid1` avec le volume physique /dev/sdf1.

```
lvconvert --replace /dev/sdb1 vg00/my_raid1 /dev/sdf1
```

Convertir le volume logique `vg00/lvpool` en un pool thin avec une taille de chunk de 128Kio et convertir `vg00/lv1` en volume thin dans ce pool. Le lv d'origine est utilisé comme origine externe d'origine, où toutes les écritures vers de tels volumes sont stockés dans le pool :

```
lvconvert --type thin --thinpool vg00/lvpool -c 128 lv1
```

Convertir le volume logique `vg00/origin` en volume thin dans le pool thin `vg00/lvpool`. Ce volume thin utilisera `vg00/origin` comme volume d'origine externe pour les zones non-provisionnées dans ce volume. L'origine externe utilise le nouveau nom `vg00/external` :

```
lvconvert -T --thinpool vg00/lvpool --originname external vg00/origin
```

Convertir un volume logique existant en LV pool cache utilisant les métadonnées de cache données :

```
lvconvert --type cache-pool --poolmetadata vg00/lvx_meta vg00/lvx_data
```

```
lvrename vg00/lvx_data vg00/lvx_cachepool
```

Convertir un volume logique existant en un LV cache dans le pool donné et une taille de chunk de 128Kio :

```
lvconvert --cache --cachepool vg00/lvx_cachepool -c 128 vg00/lvx
```

Détacher le pool cache d'un volume logique caché `vg00/lvol1` existant et laisser le pool cache non-utilisé :

```
lvconvert --splitcache vg00/lvol1
```

Supprimer le pool cache d'un volume logique existant `vg00/lvol1` :

```
lvconvert --uncache vg00/lvol1
```