
lvm

Outils lvm2

lvm fournis les outils en ligne de commande pour LVM2. Si lvm est invoqué sans arguments, affiche un prompt. Les commandes LVM peuvent être entrées directement.

Si lvm est invoqué avec `argv[0]` définis au nom d'une commande spécifique, il agit comme cette commande

À l'invocation, lvm nécessite seulement que les descripteurs de fichier `stdin`, `stdout` et `stderr` soient disponible. Si d'autres sont trouvés, ils sont fermés et des messages l'alerte sont émis.

Là où les commandes prennent des noms VG ou LV en argument, le chemin complet est optionnel. Un LV appelé `lv0` dans un VG appelé `vg0` peuvent être spécifiés sous la forme `vg0/lv0`. Quand une liste de VG est requise mais laissée vide, une liste de tous les VG sera substituée. Quand une liste de LV est requise main un VG est donné, une liste de tous les LV dans ce VG est substitué.

Un avantage d'utiliser le shell intégré est que les informations de configuration sont mis en cache en interne entre les commandes.

Un fichier contenant un simple script avec une commande par ligne peut également être donné sur la ligne de commande. Le script peut également être exécuté directement si la première ligne est `#!` suivie par le chemin de lvm.

Commandes intégrées

Les commandes suivantes sont intégrées dans lvm sans liens créés dans le système de fichier.

config `=lvmconfig`
devtypes Affiche les types de périphériques block intégrés reconnus
formats Affiche les formats de métadonnées reconnus
help Affiche l'aide
lvpoll Opérations `lvmpolld` complet
pvdata Non encore implémenté
segtypes Affiche les types de segments LVM reconnus
systemid Affiche tout system ID actuellement définis dans cet hôte
tags Affiche tous les tags définis dans cet hôte
version Affiche la version

Commandes

Les commandes suivantes implémentent les fonctionnalité de cœur de LVM

pvchange Change les attributs d'un volume physique
pvck Vérifie les métadonnées d'un volume physique
pvcreate Initialise un disque ou une partition à utiliser par LVM
pvdisplay Affiche les attributs d'un volume physique

pvmove Déplace les extents physiques
pvremove Supprime un volume physique
pvresize Redimensionne un disque ou une partition utilisé par lvm
pvs Affiche des informations sur les volumes physique
pvscan Scanne tous les disques pour les volumes physique
vgcfgbackup Sauvegarde la zone de descripteur de volume group
vgcfgrestore Restaure la zone de descripteur de volume group
vgchange Change les attributs d'un volume group
vgck Vérifie les métadonnées d'un volume group
vgconvert convertis le format de métadonnées d'un volume group
vgcreate Créé un volume group
vgdisplay Affiche les attributs d'un volume group
vgextend Ajoute des volumes physiques à un volume group
vgimport rend les volumes group exportés connus du système
vgimportclone Importe et renomme des volume group dupliqués
vgmerge Fusionne 2 volumes group
vgmknodes Recréé un répertoire de Volume Group et les fichier spéciaux de volume logique
vgreduce Réduit un volume group en supprimant un ou plusieurs volumes physique
vgremove Supprime un volume group
vgrename Renomme un volume group
vgs Affiche des informations sur les volume group
vgscan Scanne tous les disques pour les volumes group et reconstruit les caches
vgsplit Split un volume group en 2, en déplaçant les volumes logiques d'un volume group vers un autre en déplaçant tous les volumes physiques
lvchange Change les attributs d'un volume logique
lvconvert Convertis un volume logique d'un linéaire vers un mirror ou un snapshot
lvcreate Créé un volume logique dans un volume group existant
lvdisplay Affiche les attributs d'un volume logique
lvextend Étend la taille d'un volume logique
lvchange Change les attributs de LVM
lvconfig Affiche les informations de configuration après avoir chargé lvm.conf et autres fichiers de configuration
lvmdiskscan Scanne tous les périphériques visible pour LVM2
lvmdump Créé des dump d'informations lvm2 pour diagnostique
lvreduce Réduit la taille d'un volume logique
lvremove Supprime un volume logique
lvrename renomme un volume logique
lvresize Redimensionne un volume logique
lvs Affiche des informations sur les volumes logique
lvscan Scanne tous les disque pour les volumes logique

Les commande suivantes ne sont pas encore implémentées mais le seront dans le future : lvmadc, lvmsar, pvdata.

OPTIONS

Les options suivantes sont disponible pour de nombreuses commandes. Elles sont implémentées de manière générique et documentés ici au lieu des man individuels

- v, **-verbose** mode verbeux
- d, **-debug** définis le niveau de debug, de 1 à 6 fois pour augmenter les détails.
- q, **-quiet** Supprime la sortie et les messages de log. Supprime également tout prompt et répond automatiquement no
- yes Par de prompt et répond toujours yes
- t, **-test** Lance en mode test. Les commandes ne mettent pas à jours les métadonnées.
- driverloaded {yln}** à n, ne tente pas de contacter le pilote device-mapper du kernel
- A, **-autobackup {yln}** Backup automatiquement les métadonnées après un changement.
- P, **-partial** Définis, les outils fond de leur mieux pour fournir un accès aux volumes group qui sont seulement disponible partiellement. Quand un partie d'un volume logique est manquant, /dev/ioerror est substitué, et dmsetup peut être utilisé pour retourner les erreurs I/O.
- S, **-select <selection>** Pour les commandes de rapport, affiche seulement les lignes qui matchent le critère donné.
- M, **-metadatatype Type\$** Spécifie quel type de métadonnée utiliser, tel que lvm1 ou lvm2.
- ignorelockingfailure** Permet de traiter des opérations de métadonnées lecture-seule tel que lvchange -ay et vgchange -ay même si le module de lock échoue. Utile dans un script init si le répertoire de lock est monté en lecture seul.
- ignoreskippedcluster** Permet de quitter avec un status non-zero si la commande est lancé sans lockup clusterisé et certains vg clusterisés doivent être ignorés.
- readonly** Lance la commande en mode spécial lecture seul qui lit les métadonnées sur disque sans nécessiter de créer un lock.
- foreign** Force la commande à accéder aux volumes groupe extérieur. PEut être utilisé pour reporter ou afficher un VG qui est possédé par un autre hôte. Les performances peuvent être très réduite parce que le cache lvmetag n'est pas utilisé
- shared** Accède à un volume group partagé, qui serait ignoré quand lvmlockd n'est pas utilisé.
- addtag tag** Ajoute le tag spécifié à un PV, VG ou LV. Peut être spécifié plusieurs fois. Les tags peuvent être donné sur la ligne de commande à la place des arguments PV, VG, ou LV.
- deltag tag** Supprime le tag spécifié
- alloc {anywhere|contiguous|cling|inherit|normal}** Sélectionne la stratégie d'allocation quand une commande doit allouer des Extents physiques depuis le Volume Group. Chaque Volume Group et Logical Volume a une stratégie d'allocation définie. Défaut pour VG : normal, LV : inherit
 - normal** Applique des règles de sens commun qui ne place pas les stripes parallèles dans le même volume physique.
 - inherit** applique la même stratégie que le VG
 - contiguous** Nécessite que les nouveaux extents soient placé adjacent aux extents physique existant
 - cling** Place le nouvel extent physique sur le même volume physique que les extents physique dans le même stripe du volume logique
- commandprofile ProfileName** Sélectionne le profile de configuration de commande à utiliser en traitant une commande LVM.
- metadataprofile ProfileName** Sélectionne le profile de configuration des métadonnées à utiliser en traitant une commande LVM.
- profile ProfileName** idem à -metadataprofile pour vgcreate, lvcreate, vgchange et lvchange, et -commandprofile pour les autres commandes.
- config ConfigString** Configuration à utiliser

Allocation

Quand une opération doit allouer des Extents physique pour un ou plusieurs volumes physique, les outils traitent comme suit :

Tout d'abord, ils génèrent le jeu complet d'Extents Physique non-alloués dans le Volume Group. Si une plage d'Extents sont fournis à la fin de la ligne de commande, seul les Extents physique non-alloués dans cette plage dans les volumes physique spécifiés sont considérés.

Ensuite, ils tentent chaque stratégie d'allocation en retour, en commençant la stratégie la plus stricte (contiguous) et se termine avec la stratégie spécifiée avec -alloc ou la valeur par défaut. Pour chaque stratégie, en commençant depuis l'Extent logique avec le numéro le plus faible de l'espace du volume logique vide qui doit être remplis, ils allouent autant d'espace que possible en accord avec les restrictions imposées par la stratégie. Si plus d'espace est nécessaire, ils utilise la stratégie suivante.

Les restrictions sont les suivantes :

Contiguous nécessite que l'emplacement physique d'un extent logique qui n'est pas le premier extent logique d'un volume logique soit adjacent à l'emplacement physique de l'extent logique le précédent immédiatement.

Quand un volume logique est stripé ou miroiré, les restrictions ci-dessus sont appliquées indépendamment à chaque stripe ou image miroir qui nécessite de l'espace.

Normal ne choisit pas un Extent Physique qui partage le même volume physique comme Extent Logique déjà alloué à un volume logique parallèle (ex : un striped ou une image miroir différente) au même offset dans ce volume logique parallèle.

Types de volume logique

Certains types de volumes logiques sont simple à créer et peut être fait avec une simple commande `lvcreate`. Les types de volume logique `linear` et `striped` en sont un exemple. D'autres types de volume logique peuvent nécessiter plus d'une commande pour les créer, comme les types `cache` et `thin provisioning`.

Critères de Sélection

Les critères de sélection sont un jeu de déclaration combinés par des opérateurs logique et de groupement. La déclaration consiste de nom de colonne pour lequel un jeu de valeurs valides est définis en utilisant des opérateurs de comparaison. Pour une liste complète de noms de colonne qui peuvent être utilisé dans la sélection, voir la sortie de `<commande lvm> -S help`

Opérateurs de comparaison (cmp_op)

- `==~` Matche l'expression régulière
- `!~` ne match pas l'expression régulière
- `=` égal à
- `!=` Non égal à
- `>=` Supérieur ou égal à
- `>` Supérieur à
- `<=` Inférieur ou égal à
- `<` Inférieur à

Opérateurs logiques binaire (cmp_log)

- `&&` Tous les champs doivent matcher
- `,` Idem
- `||` Au moins un champs doit matcher
- `#` idem

Opérateurs logique unaire

! Négation

Opérateurs de groupement

() parenthèses

[] Liste

{ } Sous-jeu de liste

Spécification de grammaire informelle

STATEMENT = column cmp_op VALUE | STATEMENT log_op STATEMENT | (STATEMENT) | !(STATEMENT)

Variables d'environnement

HOME Répertoire contenant `.lvm_history` si le shell readline interne est invoqué

LVM_COMAND_PROFILE Nom du profile de commande par défaut à utiliser pour les commandes LVM.

LVM_SYSTEM_DIR Répertoire contenant `lvm.conf`. Défaut : `/etc/lvm`

LVM_SUPPRESS_FD_WARNINGS Supprime les alertes sur les descripteurs de fichier inattendus passés à lvm

LVM_VG_NAME Nom de volume group qui est assumé pour toute référence à un volume logique qui ne spécifie pas de chemin.

LVM_LVMETAD_PIDFILE Nom du volume group qui stocke l'ID du process `lvmetad`

LVM_LVMETAD_SOCKET Chemin du socket utilisé pour communiquer avec `lvmetad`

LVM_LVMPOLLD_PIDFILE Chemin du fichier qui stocke l'ID du processus `lvmpolld`

LVM_LVMPOLLD_SOCKET Chemin du socket utilisé pour communiquer avec `lvmpolld`

LVM_LOG_FILE_EPOCH Chaîne jusqu'à 32 lettre à ajouter au nom du fichier de log et suivi par l'ID du processus et un horodatage. Si définis, chaque processus log dans un fichier séparé

LVM_EXPECTED_EXIT_STATUS Le status anticipé quand le processus quitte. Utiliser ">N" pour matcher un status supérieur à N

LVM_SUPPRESS_LOCKING_FAILURE_MESSAGES Utilisé pour supprimer les messages d'alerte quand le lock configuré est indisponible

DM_ABORT_ON_INTERNAL_ERRORS Annule le traitement si le code détecte une erreur interne non-fatal

DM_DISABLE_UDEV Évite l'interaction avec `udev`. LVM va gérer les nœud dans `/dev` directement