
lvmlockd

Service de lock LVM

Les commandes LVM utilisent lvmlockd pour coordonner l'accès au stockage partagé. Quand LVM est utilisé sur des périphériques partagés par plusieurs hôtes, les locks vont :

- Coordonner la lecture/écriture des métadonnées LVM
- Valider le cache des métadonnées
- Empêcher l'activation concurrente des volumes logiques

lvmlockd utilise un gestionnaire de lock externe, les options sont :

sanlock Place les locks sur le disque dans le stockage LVM

dlm Utilise la communication réseaux et un gestionnaire de cluster

OPTIONS

- testl-T** Mode test
- foregroundl-f** Ne lance pas en tâche de fond
- daemon-debugl-D** Ne fork pas et affiche le debug sur stdout
- pid-filel-p path** Spécifie le fichier PID
- socket-pathl-s path** Définis le chemin du socket
- syslog-priorityl-S err|warning|debug** Spécifie la priorité syslog
- gl-type-l-g sanlock|dlm** Définis le type de lock global
- host-idl-i num** Définis l'id hôte pour le sanlock local
- host-id-filel-F path** Un fichier contenant le sanlock local host_id
- sanlock-timeoutl-o seconds** Remplace le timeout d'E/S sanlock par défaut
- adoptl-A 0|1** récupère les locks d'une instance précédente de lvmlockd

Utilisation

En utilisant LVM avec lvmlockd la première fois, il est nécessaire d'effectuer les étapes suivantes :

1. Choisir un gestionnaire de lock

dlm Si dlm (ou corosync) est déjà utilisé par un autre logiciel cluster, sélectionner dlm. dlm utilise corosync qui nécessite une configuration additionnelle au delà du périmètre de ce document.

sanlock sanlock ne dépend pas d'un logiciel de clustering

2. Configurer les hôtes pour utiliser lvmlockd

 Sur tous les hôtes où lvmlockd fonctionne, configurer lvm.conf :

```
locking_type = 1
use_lvmlockd = 1
# avec sanlock, assigner un host_id unique (de 1 à 2000) dans /etc/lvm/lvmlocal.conf local/host_id
```

3. **Démarrer lvmlockd** Utiliser un fichier d'init ou simplement lancer lvmlockd
4. **Démarrer le gestionnaire de lock** (sanlock) `systemctl start wdm` sanlock ou (dlm) `systemctl start corosync dlm`
5. **Créer un VG sur les périphériques partagés** `vgcreate --shared <vgname> <devices>`. L'option `shared` définit le type de lock VG à sanlock ou dlm en fonction du gestionnaire utilisé.
6. **Démarrer le VG sur tous les hôtes** `vgchange --lock-start`. lvmlockd nécessite que les VG partagés soient démarrés avant qu'ils soient utilisés. Le lock manager démarre (joint) le lockspace du VG et peut prendre du temps.
7. **Créer et activer les LV** Les commandes `lvcreate` et `lvchange` sont utilisées pour créer et activer les LV dans un VG partagé. Un LV activé exclusivement sur un hôte ne peut pas être activé sur un autre. Quand plusieurs hôtes doivent utiliser le même LV simultanément, le LV peut être activé avec un lock partagé.

Démarrage et arrêt Une fois la configuration initiale, le démarrage et l'arrêt incluent les étapes suivantes. Elles sont effectuées manuellement ou en utilisant le gestionnaire de service système :

- Démarrer lvmlockd
- Démarrer lvmlockd
- Démarrer le lock manager
- `vgchange --lock-start`
- Activer les LV dans les VG partagés

La séquence d'arrêt est l'inverse :

- Désactiver les LV dans les VG partagés
- `vgchange --lock-stop`
- Arrêter le gestionnaire de lock
- Arrêter lvmlockd
- Arrêter lvmlockd

Contrôle d'accès au VG

Les termes suivants sont utilisés pour décrire les différentes formes de contrôle d'accès au VG.

Un **lockd VG** est un VG partagé qui a un 'lock type' `dlm` ou `sanlock`. Il nécessite lvmlockd. Ces VG existent sur des stockages partagés qui sont visibles à plusieurs hôtes. Les commandes LVM utilisent lvmlockd pour effectuer le lock pour ces LV quand ils sont utilisés.

Si le gestionnaire de lock pour le type de lock n'est pas disponible (par ex non démarré est échoué), lvmlockd n'est pas capable d'acquiescer les locks pour les commandes LVM. Les commandes LVM qui lisent uniquement le VG vont généralement être autorisés à continuer sans locks dans ce cas. Les commandes pour modifier ou activer le VG vont échouer.

Un **local VG** est utilisé par un simple hôte. Il n'a pas de type de lock ou un type 'none'. Les commandes LVM et lvmlockd n'effectuent pas de lock pour ces VG. Un VG local existe typiquement sur les périphériques locaux et ne peuvent pas être utilisés par différents hôtes.

Si un VG local existe sur des périphériques partagés, il devrait être possédé par un simple hôte en ayant son `system_id` définis. Seul l'hôte avec un `system_id` correspondant peut utiliser le VG local. Un VG sans type de lock et sans `system_id` devrait être exclus de tous sauf un hôte en utilisant les filtres `lvm.conf`. Sans protection, un VG local sur un périphérique partagé peut facilement être endommagé ou détruit.

Un **VG clvm** est un VG sur un stockage partagé qui nécessite `clvmd` pour le clustering.

VG lockés sur les hôtes n'utilisant pas lvmlockd

Seul les hôtes qui utilisent les VG lockd devraient être configurés pour lancer lvmlockd. Cependant, les périphériques partagés utilisés par les VG lockd peuvent être visible aux hôte n'utilisant pas lvmlockd. D'un hôte n'utilisant pas lvmlockd, les VG lockd sont ignorés de la

même manière que les VG étranger. L'option `-shared` pour le reporting et les commandes d'affichage affichent les VG lockd sur un hôte n'utilisant pas `lvmlockd`, tout comme `-foreign` affiche les VG étranger.

Comparaison `vgcreate`

Le type de contrôle d'accès VG est spécifié dans la commande `vgcreate` pour toutes les options `vgcreate`

`vgcreate <vgname> <devices>` Créé un VG local avec le `system_id` local quand `lvmlockd` et `clvm` ne sont pas configurés

Créé un VG local avec le `system_id` local quand `lvmlockd` est configuré

Créé un VG `clvm` quand `clvm` est configuré

`vgcreate -shared <vgname> <devices>` Nécessite que `lvmlockd` soit configuré et fonctionnel

Créé un VG lockd avec le type de lock `sanlockldlm` en fonction du gestionnaire de lock courant

Les commandes LVM nécessitent des locks de `lvmlockd` pour utiliser le VG

`lvmlockd` obtient les locks depuis le gestionnaire de lock sélectionné

`vgcreate -cl-clustered y <vgname> <devices>` Nécessite que `clvm` soit configuré et fonctionnel

Créé un VG `clvm` avec le flag `clustered`

Les commandes LVM réclament les lockd à `clvmd` pour utiliser le VG

Créer le premier VG sanlock

Créer le premier VG sanlock n'est pas protégé par le locking et nécessite une attention particulière. Les sanlock existent dans le VG, donc ils ne sont pas disponibles tant que le VG n'existe pas. Le premier sanlock VG contient le lock global.

- Le premier `vgcreate` doit avoir le chemin vers un périphérique qui n'a pas été initialisé avec `pvcreate`. L'initialisation `pvcreate` nécessite le lock global, qui ne sera pas disponible tant que le premier sanlock VG n'est pas créé
- En lançant `vgcreate` pour le premier sanlock VG, s'assurer que le périphérique utilisé n'est pas utilisé par une autre commande LVM. L'allocation de périphériques partagés est généralement protégée par le lock global, mais cela ne peut pas être fait pour le premier sanlock VG qui va maintenir le lock global
- En lançant `vgcreate` pour le premier sanlock VG, s'assurer que le nom du VG n'est pas utilisé par une autre commande LVM. L'unicité des noms des VG est généralement assurée par le lock global.
- Parce que le premier sanlock VG contient le lock global, ce VG doit être accessible à tous les hôtes qui utilisent les VG partagés avec sanlock. Tous les hôtes doivent utiliser le lock global depuis le premier VG sanlock.

Utiliser les VG lockd

Il y a quelques considérations spéciales en utilisant les VG lockd. Quand `use_lvmlockd` est activé dans `lvm.conf`, puis le premier VG lockd est créé, il n'existe pas de lock global. Dans cet état initial, les commandes LVM tentent et échouent à acquérir le lock global, produisant une alerte, et certaines commandes sont désactivées. Une fois le premier VG lockd créé, le lock global sera disponible, et LVM sera pleinement opérationnel.

Quand un nouveau VG lockd est créé, son espace de lock est automatiquement démarré sur l'hôte qui l'a créé. Les autres hôtes doivent lancer `vgchange -lock-start` pour démarrer le nouveau VG avant de pouvoir l'utiliser.

Depuis la commande `vgs`, les VG lockd sont indiqués par 's' dans le champ `attr`. Le type de lock et les arguments de lock peuvent être affichés avec **`vgs -o+locktype,lockargs`**

Les VG lockd doivent être démarrés et stoppés, à la différence d'autres types de VG.

vgremove d'un VG lockd échoue si d'autres hôtes ont le VG démarré. Lance `vgchange --lock-stop <vgname>` sur tous les autres hôtes avant un `vgremove`.

Démarrer et stopper les VG

Démarrer un VG lockd (`vgchange --lock-start`) indique au gestionnaire de lock de démarrer (joindre) le lockspace pour le VG sur l'hôte où il est lancé. Les locks créés pour le VG sont disponibles pour les commandes LVM sur l'hôte. Avant qu'un VG soit démarré, seules les commandes LVM qui lisent/affichent le VG sont autorisées à continuer sans locks. En utilisant le type de lock `sanlock`, démarrer un VG peut prendre du temps.

Stopper un VG lockd (`vgchange --lock-stop`) indique au gestionnaire de stopper (quitter) le lockspace pour le VG sur l'hôte où il est lancé. Un VG ne peut pas être stoppé pendant qu'il a des LV actifs.

Un VG lockd peut être démarré une fois que les points suivants soient vrais :

- `lvmlockd` est en cours de fonctionnement
- le gestionnaire de lock est en cours de fonctionnement
- Le VG est visible au système

Un VG lockd peut être stoppé si tous les LV sont désactivés. Tous les VG lockd peuvent être démarrés/stoppés en utilisant :

`vgchange --lock-start`

`vgchange --lock-stop`

Les VG individuels peuvent être démarrés/stoppés en utilisant :

`vgchange --lock-start <vgname> ...`

`vgchange --lock-stop <vgname> ...`

Pour que `vgchange` n'attende pas que le démarrage soit complété :

`vgchange --lock-start --lock-opt nowait ...`

`lvmlockd` peut être appelé directement pour stopper tous les lockspaces

`lvmlockctl --stop-lockspaces`

Pour démarrer seulement les VG lockd sélectionnés, utiliser `lvm.conf` `activation/lock_start_list`. Quand défini, seuls les VG nommés dans cette liste sont démarrés par `vgchange`. Si la liste n'est pas définie, tous les VG lockd visibles sont démarrés.

Démarrage et activation automatique

Les scripts ou programmes dans l'hôte qui démarrent automatiquement les VG utilisent l'option `auto` pour indiquer que la commande est lancée automatiquement par le système :

`vgchange --lock-start --lock-opt auto [<vgname> ...]`

Sans configuration additionnelle, l'option `auto` n'a pas d'effet ; tous les VG sont démarrés sauf restreint par `lock_start_list`

Cependant, quand `activation/auto_lock_start_list` dans `lvm.conf` est défini, la commande de démarrage `auto` effectue une phase de filtrage supplémentaire pour tous les VG démarrés, en testant chaque nom de VG avec la liste `auto_lock_start_list`. Cette directive définit les VG lockd qui seront démarrés par la commande `auto`. Les VG lockd visible non incus dans la liste sont ignorés par la commande de démarrage `auto`. Si la liste est indéfinie, tous les noms de VG passent ce filtre.

`auto_lock_start_list` permet à un utilisateur de sélectionner certains VG lockd qui devraient être démarrés automatiquement par le système. Pour utiliser l'auto activation des LV lockd, (voir `auto_activation_volume_list`), le démarrage automatique des VG lockd correspondants sont nécessaire.

Lock interne

Pour optimiser l'utilisation de LVM avec lvmlockd, il y'a 3 types de locks :

GL lock

Le lock global est associé avec les informations globales, qui ne sont pas isolés à un seul VG. Cela inclus l'espace de nom VG global, le jeu de périphérique non-utilisés et les PV orphelins et les propriétés des PV orphelins. Le lock global est utilisé en mode partagé par les commandes qui lisent cette information, ou en mode exclusif par les commandes que la change.

La commande vgs acquière le lock global en mode partagé parce qu'il reporte la liste de tous les noms des VG.

vgcreate acquière le lock global en mode exclusif parce qu'il créé un nouveau nom VG, et prend un PV de la liste des PV non utilisés

Quand une commande LVM est donnée avec un tag, ou utilise select, il doit lire tous les VG pour correspondre au tag ou à la sélection, qui nécessite d'acquérir le lock global.

VG lock

Un lock VG est associé avec chaque VG. Il est obtenu en mode partagé pour lire le VG et en mode exclusif pour le changer. Ce lock sérialise l'accès au VG avec toutes les autres commandes LVM accédant au VG depuis tous les hôtes.

La commande vgs n'acquière le lock global que pour lire la liste de tous les noms VG, mais acquière le lock VG pour chaque VG avant de le lire.

lock LV

Un lock LV est acquis avant que le LV soit activé, et est relâché après que le LV soit désactivé. Si le lock LV ne peut pas être acquis, le LV n'est pas activé. Les locks LV sont persistant et restent en place une fois l'activation faite. Les locks GL et VG sont transitoires et sont maintenus seulement pendant une commande LVM.

Retentatives de lock

Si une requête pour un lock GL ou VG échoue à cause d'un conflit de lock avec un autre hôte, lvmlockd retente automatiquement pendant un court laps de temps avant de retourner une erreur à la commande LVM. Si ces retentatives sont insuffisantes, la commande LVM va retenter toute la demande de lock un nombre de fois spécifié par global/lvmlockd_retries avant d'échouer. Si une requête pour un lock LV échoue à cause d'un conflit de lock, la commande échoue immédiatement.

Gérer le lock global dans les VG sanlock

Le lock global existe dans un des VG sanlock. Le premier VG sanlock créé va contenir le lock global. Les VG sanlock suivant vont chacun contenir des locks global désactivés qui peuvent être activés ultérieurement si nécessaire.

Le VG contenant le lock global doit être visible à tous les hôtes utilisant les VG sanlock. Cela peut être une raison de créer un petit VG

sanlock, visible à tous les hôtes, et dédié à maintenir ce lock global. Bien que non requis, cette stratégie peut éviter les problèmes si des VG sont déplacés ou supprimés.

La commande `vgcreate` acquiert typiquement le lock global, mais dans le cas du premier VG sanlock, il n'y a pas de lock global à acquérir tant que le premier `vgcreate` n'est pas complété. Donc, créer le premier VG sanlock est un cas spécial qui ignore le lock global.

`vgcreate` pour un VG sanlock détermine qu'il est le premier à exister si aucun autre VG sanlock n'est visible. Il est possible que d'autres VG sanlock existent mais ne sont pas visibles à l'hôte que lance `vgcreate`. Dans ce cas, `vgcreate` crée un nouveau VG sanlock avec le lock global activé. Quand l'autre VG sanlock avec le lock global apparaît, `lvmlockd` va voir plus d'un VG avec un lock global activé, et les commandes LVM vont reporter qu'il y a des locks globaux dupliqués.

Si la situation se produit où plus d'un VG sanlock contient un lock global, le lock global devrait être désactivé manuellement dans tous sauf un avec la commande :

```
lvmlockctl -gl-disable <vgname>
```

Un problème opposé peut se produire si le VG maintenant le lock global est supprimé. Dans ce cas, aucun lock global n'existe, et les commandes LVM échouent à l'acquiescer. Dans ce cas, le lock global doit être manuellement activé dans un des VG sanlock restant avec la commande :

```
lvmlockctl -gl-enable <vgname>
```

Un petit VG sanlock dédié pour maintenir le lock global peut éviter ce cas.

LV lvmlock interne

Un VG sanlock contient un LV caché appelé `lvmlock` qui maintient les locks sanlock. `vgreduce` ne peut pas supprimer le PV maintenant le LV `lvmlock`. Pour supprimer ce PV, changer le type de lock VG à `none`, lancer `vgreduce`, puis changer le type de lock VG à `sanlock`. Similairement, `pvmove` ne peut pas être utilisé dans un PV utilisé par le LV `lvmlock`. Pour placer le LV `lvmlock` dans un périphérique spécifique, créer le VG avec seulement ce périphérique, puis utiliser `vgextend` pour ajouter d'autres périphériques.

LV partagés

Quand un LV est utilisé en concurrence par plusieurs hôtes, le LV peut être activé sur plusieurs hôtes en utilisant un lock partagé. Pour activer le LV avec un lock partagé : **`lvchange -asy vg/lv`**

Avec `lvmlockd`, une mode d'activation non-spécifié est toujours exclusif, par exemple, `-ay` vaut `-aey`.

Si le type LV ne permet pas au LV d'être utilisé en concurrence par plusieurs hôtes, le lock d'activation partagé n'est pas permis et la commande `lvchange` va reporter une erreur. Les types LV qui ne peuvent pas être utilisés en concurrence depuis plusieurs hôtes sont les `thin`, `cache`, `raid`, `mirror`, et `snapshot`.

`lvextend` sur un LV avec des locks partagés n'est pas encore permis. Le LV doit être désactivé, ou activé exclusivement pour lancer `lvextend`.

Récupérer les locks sanlock des PV perdus

L'approche générale est de changer le type de lock VG à `none`, puis remettre à `sanlock`. Cela recrée le LV `lvmlock` interne et les locks nécessaires. D'autres étapes peuvent être nécessaires pour gérer les PV manquants.

Erreurs système de lock

lvmlockd Si lvmlockd échoue ou est terminé pendant qu'il maintient les lock, les locks sont orphelins dans le gestionnaire de lock. lvmlockd peut être redémarré avec une options pour adopter les locks dans le gestionnaire de lock qui les maintenait depuis une instance précédente.

dlm/corosync Si dlm ou corosync échoue, le système de clustering va isoler l'hôte en utilisant une méthode configuré dans le cluster. Les commandes LVM sur les autres hôtes ne pourront pas acquérir de lock tant que le processus de récupération n'est pas terminé.

Erreur de stockage sanlock

Si le PV sous un LV lvmlock dans un VG sanlock est déconnecté, ne répond pas ou est trop lent, sanlock ne peut pas renouveler le bail pour les locks des VG. Après un certain temps, le bail expire, et les locks que l'hôte possède dans le VG peuvent être acquis par d'autres hôtes. Le VG doit être désactivé sur l'hôte avec le bail expiré avant que d'autres hôtes puissent acquérir les locks.

Quand le service sanlock détecte que le bail du stockage est perdu, il lance la commande `lvmlockctl --kill <vgname>`. Cette commande émet un message syslog indiquant que le bail du stockage est perdu pour le VG et les LV doivent être immédiatement désactivés.

Si aucun LV n'est actif dans le VG, le lockspace avec un bail expiré est supprimé, et les erreurs sont reportés en tant qu'il n'est possible d'utiliser le VG. Utiliser `lvmlockctl --drop` pour effacer le lockspace gelé de lvmlockd.

Si le VG a des LV actifs quand le lock est perdu, les LV doivent être rapidement désactivés avant que le bail lockspace n'expire. Une fois tous les LV désactivés, `lvmlockctl --drop <vgname>` efface le lockspace de lvmlockd. Si tous les LV dans le VG ne sont pas désactivés dans les 40 secondes, sanlock réinitialise l'hôte en utilisant le watchdog local. La réinitialisation machine est effectivement une forme sévère de désactivation des LV avant de pouvoir être activés dans d'autres hôtes.

Dans le future, `lvmlockctl kill` pourra tenter automatiquement de forcer la désactivation des LV avant le que le bail n'expire.

Erreur du service sanlock

Si le service sanlock échoue pour quitter alors que le lockspace est démarré, le watchdog local réinitialise l'hôte. C'est nécessaire pour protéger les ressources d'application qui dépendent des bails sanlock.

Changer le nom du cluster dlm

Quand un VG dlm est créé, le nom du cluster est sauvegardé dans les métadonnées du VG. Pour utiliser le VG, un hôte doit être dans le cluster dlm nommé. Si le nom change, ou le VG est déplacé dans un nouveau cluster, le nom du cluster dlm sauvegardé dans le VG doit également être changé.

Pour voir le nom du cluster dlm dans le VG, utiliser

`vgs -o+locktype,lockargs <vgname>`

Pour changer le nom du cluster dlm dans le VG :

- stopper le VG sur tous les hôte

`vgchange --lock-stop <vgname>`

- Changer le type de lock VG à none

`vgchange --lock-type none <vgname>`

- Changer le nom du cluster dlm sur l'hôte ou déplacer le VG dans le nouveau cluster. Le nouveau cluster dlm doit être actif sur l'hôte

`cat /sys/kernel/config/dlm/cluster/cluster_name`

- Changer le type de lock VG à dlm qui définit le nouveau nom du cluster

`vgchange --lock-type dlm <vgname>`

Démarrer le VG sur les hôtes qui l'utilisent

vgchange --lock-start <vgname>

Pour changer le nom du cluster dlm dans le VG quand le nom du cluster dlm a déjà changé, ou le VG a déjà été déplacé dans un autre cluster :

- S'assurer que le VG n'est pas utilisé par un hôte

- Le nouveau cluster dlm doit être actif

cat /sys/kernel/config/dlm/cluster/cluster_name

- Changer le type de lock VG à none

vgchange --lock-type none --force <vgname>

- Changer le type de lock à dlm

vgchange --lock-type dlm <vgname>

Démarrer le VG sur les hôtes qui l'utilisent

vgchange --lock-start <vgname>

Changer un VG local en VG lockd

Tous les LV doivent être inactifs pour changer le type de lock. lvmlockd doit être configuré et en cours de fonctionnement.

Changer un VG local en VG lockd avec la commande

vgchange --lock-type sanlockldlm <vgname>

Démarrer le VG sur les hôtes qui l'utilisent

vgchange --lock-start <vgname>

Changer un VG lockd en VG local

Stopper le VG lockd sur tous les hôtes, puis lancer

Pour changer un VG d'un type lockd à un autre, le changer d'abord en un VG local, puis dans le nouveau type

Changer un VG clvm en VG lockd

Tous les LV doivent être inactif pour changer le type de lock. Changer d'abord le VG clvm en VG local. Dans un cluster clvm en cours de fonctionnement, changer un VG clvm en VG local avec

vgchange -cn <vgname>

Si le cluster clvm n'est plus en cours de fonctionnement sur aucun nœud, des options supplémentaires peuvent être utilisées pour forcer le VG local

vgchange --config 'global/locking_type=0 global/use_lvmlockd=0' -cn <vgname>

Une fois le VG en local, suivre les étapes dans changer un VG local en un VG lockd.

Limites des VG lockd

Listes de ce qui ne fonctionne pas encore dans les VG lockd

- Créer un nouveau pool thin et un nouveau lv thin en une seule commande
- Utiliser lvcreate pour créer des pools cache ou des LV cache (utiliser lvconvert)
- Utiliser des origines externes pour les LV thin
- Splitter les miroirs est snapshots des LV
- vgsplit

-
- vgmerge
 - redimensionner un LV qui est actif dans le mode partagé sur plusieurs hôtes.

Changement lvmlockd depuis clvmd

Bien que lvmlockd et clvmd sont des système entièrement différents, l'utilisation des commande LVM restent similaires. Les différences sont plus notables en utilisant l'option sanlock de lvmlockd. La différence d'utilisation visible entre les VG lockd avec lvmlockd et les VG clvm avec clvmd sont :

- lvm.conf doit être configuré pour utiliser soit lvmvlockd (use_lvmlockd=1) soit clvmd (locking_type=3), mais pas les 2
- vgcreate --shared créé un VG lockd, et vgcreate --clustered y créé un VG clvm
- lvmlockd ajoute l'option pour utiliser sanlock pour le locking, évitant le besoin d'un cluster réseau.
- lvmlockd utilise le mode d'activation exclusif par défaut quand le mode l'activation n'est pas spécifié. -ay signifie -aey
- Les commandes lvmlockd s'appliquent toujours à l'hôte local, et n'a pas d'effet sur les hôte distants
- lvmlockd fonctionne avec les pools thin et cache
- lvmlockd fonctionne avec lvmtools
- lvmlockd sauve le nom du cluster dans le VG lockd utilisant dlm
- lvmlockd nécessite de démarrer/stopper les VG lockd avec vgchange --lock-start et --lock-stop
- vgremove d'un VG sanlock peut échouer en indiquant que tous les hôtes ne sont pas stoppés dans le lockspace du VG
- vgreduce ou pvmove d'un PV dans un VG sanlock échoue s'il maintient le LV lvmlock
- lvmlockd utilise les tentatives lock au lieu de files d'attente de lock
- lvmlockd inclus des options de reporting VG lock_type et lock_args, et une options de reporting LV lock_args pour voir les champs de métadonnées correspondants
- Les commandes vgs, le 6ème champ, "s" pour "shared" est affiché pour les VG lockd
- Si lvmlockd échoue ou est terminé pendant qu'il est en cours d'utilisation, les locks restent, mais sont orphelins dans le gestionnaire de lock. lvmlockd peut être redémarré avec une option pour adopter les locks orphelins depuis une précédente instance de lvmlockd.