
lvmsystemid

ID système LVM

Les VG locaux peuvent exister sur des stockages partagés où ils sont visibles à plusieurs hôtes. Ces VG sont prévus pour être utilisés par une seule machine. Un `system_id` identifiant un hôte unique peut être assigné à un VG pour indiquer le propriétaire du VG. Ce propriétaire peut utiliser le VG normalement, et tous les autres hôtes vont l'ignorer.

Le `system id` n'est pas une propriété dynamique, et peut seulement être changée dans des circonstances très limitées. Ces changements limités ne sont pas parfaitement reflétés via les hôtes. Une vue plus cohérente de stockage partagée nécessite d'utiliser un système de lock inter-hôte pour coordonner l'accès et mettre à jours les caches.

Le `system_id` est une chaîne identifiant un hôte de manière unique. Il peut être défini manuellement ou automatiquement par `lvm` en utilisant un identifiant unique déjà disponible dans l'hôte, par exemple `machine-id` ou `uname`.

Dans `vgcreate`, le `system_id` local est sauvé dans les métadonnées du VG. L'hôte local possède le nouveau VG, et les autres hôtes ne peuvent pas l'utiliser.

Un VG sans `system_id` peut être utilisé par n'importe quel hôte. un `foreign VG` est un VG avec un `system_id` tel que vu par un hôte avec un `system_id` qui ne matche pas le `system_id` du VG.

Les caractères de `system_id` valides sont les mêmes que les caractères valides pour un nom de VG. Si un `system_id` contient des caractères invalides, ils sont omis. Si un `system_id` dépasse la longueur maximale (128), il est tronqué.

Limitations et alertes

pour bénéficier pleinement du `system_id`, tous les hôtes doivent avoir un `system_id`, et les VG également. Un VG sur un stockage partagé peut être endommagé ou détruit dans certains cas.

- Un VG sans `system_id` peut être utilisé sans restriction par tous les hôtes, même les hôtes qui ont un `system_id`. Pour vérifier qu'un VG a un `system_id`, utiliser `vgs -o+systemid`. Pour ajouter un `system_id`, utiliser `vgchange --systemid`
- 2 hôtes ne devraient pas avoir le même `system_id`.
- Les PV orphelins (ou les périphériques inutilisés) sur des stockages partagés sont complètement non-protégés par le `systemid`. Les commandes qui utilisent ces PV, tel que `vgcreate` ou `vgextend`, peut effectuer des opérations conflictuelles et corrompre les PV.

types d'accès VG

Un VG local est prévu pour être utilisé par un simple hôte. Un VG partagé ou clusterisé est prévu pour être utilisé par plusieurs hôtes. Ils peuvent être distingués par :

non-restreint : un VG local qui n'a pas de `system_id`. Ce type de VG n'est pas protégé est accessible à tous les hôtes

possédé : un VG local qui a un `system_id`, tel que vu par un hôte ayant un `system_id` correspondant. Ce VG est accessible à l'hôte

Étranger : un VG local qui a un `system_id`, tel que vu par un hôte ayant un `system_id` qui ne correspond pas.

Exporté : un VG local qui a été exporté avec `vgexport` et n'a pas de `system_id`. Ce type de VG ne peut être accédé que par `vgimport`.

partagé : un VG partagé ou "lockd" a lock_type définis et pas de system_id. il est utilisé sur un stockage partagé pour plusieurs hôtes, uniquement via lvmlockd

clusterisé : Un VG clusterisé ou "clvm" a le flag clustered mis et pas de system_id. Il est utilisé sur un stockage partagé par plusieurs hôtes, en utilisant clvmd.

system_id_source Le system_id d'un hôte peut être définis de plusieurs manières. lvm.conf global/system_id_source définis la méthode utilisée pour trouver le system_id local :

none lvm n'utilise pas de system_id

machineid Le contenu de /etc/machine-id est utilisé comme id système

uname La chaîne utsname.nodename de uname(2) est utilisé comme system_id.

lvmlocal Définis dans lvmlocal.conf local/system_id

file définis dans un fichier spécifié par lvm.conf global/system_id_file

Changer system_id_source change le system_id, ce qui peut empêcher l'utilisation de VG. Si un system_id_source autre que none échoue à la résolution du system_id, l'hôte sera autorisé à accéder aux VG sans system_id, mais ne sera pas autorisé à accéder aux VG avec un system_id.

extra_system_ids

Dans certains cas, l'hôte qui lance la commande assigne son propre system_id au nouveau VG. Pour le remplacer par un autre system_id, utiliser : **vgcreate --systemid <SystemID> <VG> <Devices>**. Si l'argument de --systemid est une chaîne vide, le VG est créé sans system_id.

Rapport/affichage

Le system_id d'un VG est affiché avec l'option de reporting systemid. Les commandes de rapport/affichage ignorent les VG étrangérés par défaut. Pour afficher ces VG, l'option --foreign peut être utilisé. Cela implique que la commande lise les VG sur les disques et non via lvmetad.

vgexport/vgimport

vgexport efface le system_id. Les autres hôtes continuent à voir le VG nouvellement exporté comme étrangéré à cause du caching local (quand lvmetad est utilisé). Mettre à jours manuellement le cache avec pvscan --cache pour reconnaître le nouveau VG exporté.

vgimport définis le system_id. vgimport scanne automatiquement les nouveaux VG exportés.

vgchange

Un hôte peut changer le system_id de ses propres VG, mais la commande nécessite confirmation du fait que l'hôte perd l'accès à ces VG
vgchange --systemid SystemID VG

Le system_id peut être supprimé d'un VG en spécifiant une chaîne vide comme nouveau system_id. Un hôte ne peut pas changer directement le system_id d'un VG étranger. Pour déplacer un VG d'un hôte à un autre, vgexport et vgimport doivent être utilisés. Pour devenir propriétaire d'un VG étranger, un hôte peut ajouter le system_id étranger à sa liste extra_system_ids, changer le system_id de ce VG, puis supprimer le system_id de sa extra_system_ids.

VG partagés

Un VG partagé/lockd n'a pas de `system_id`, permettant à plusieurs hôtes de l'utiliser via `lvmlockd`. Changer un VG en un type `lockd` efface le `system_id`.

VG clusterisés

Un VG clusterisé/clvm n'a pas de `system_id`, permettant à plusieurs hôtes de l'utiliser via `clvmd`. Changer un VG en type `clustered` efface le `system_id` existant. Changer le VG `clustered` en `non-clustered` définit le `system_id` de l'hôte qui a lancé la commande `vgchange`.

creation_host

dans `vgcreate`, le champ de métadonnées VG `creation_host` est défini par défaut à `uname` de l'hôte. Ce champ ne peut pas être changé, et n'est pas utilisé pour contrôler l'accès. Quand `system_id_source` est `"uname"`, `system_id` et `creation_host` sont les mêmes.

Orphelins

Les VG orphelins sont des périphériques non-utilisés. À cause de cela, ils ne sont pas protégés par un `system_id`, et tous les hôtes peuvent les utiliser. La coordination des changements des changements en VG orphelins est hors du périmètre du `system_id`.

Les effets sont spécialement évidents quand `lvm` utilise `lvmtools`. Par exemple, si plusieurs hôtes voient un VG orphelin, et un hôte crée un VG en utilisant l'orphelin, les autres hôtes continuent de le voir orphelin. Rien n'empêche les autres hôtes d'utiliser ce PV et le corrompre. Si les autres hôtes lancent une commande de `rescan`, et mettent à jour `lvmtools`, ils le reconnaîtront.