
zpool

Configure des pools de stockage ZFS

Description

Un pool de stockage est une collection de périphériques qui fournissent un stockage physique et une réplication de données pour les datasets ZFS. Tous les datasets dans un pool de stockage partagent le même espace.

Un périphérique virtuel (vdev) décrit un simple périphérique ou une collection de périphériques organisés en accord avec certaines caractéristiques de performances et panne. Les périphériques virtuels suivants sont supportés :

disk Un périphérique block, typiquement localisé sous /dev/dsk. ZFS peut utiliser individuellement des slices ou des partitions, mais le mode d'opération recommandé est d'utiliser des disques entiers. Un disque peut être spécifié par un chemin complet, ou il peut être un nom court (la partie relative à /dev/dsk). Un disque entier peut être spécifié en omettant le slice ou la partition. par exemple, "c0t0d0" est équivalent à "/dev/dsk/c0t0d0s2". En donnant un disque entier, ZFS label automatiquement le disque, si nécessaire.

file Un fichier régulier. L'utilisation de disques régulier est fortement découragé. Il est conçu principalement à des fins expérimentaux.

mirror Un miroir de 2 ou plusieurs disques. Les données sont répliquées de manière identique dans tous les composant d'un miroir. Un miroir avec N disques et taille X peut maintenir X octets et peut résister à N-1 panne de périphériques avant que l'intégrité des données soient compromise.

raidz

raidz1

raidz2

raidz3 Une variation du RAID-5 qui permet une meilleur distribution de la parité et élimine les trous d'écriture des RAID-5 (dans lequel les données et la parité deviennent inconsistants avec une coupure de courant). Les données et la parité sont placés dans tous les disques dans un groupe raidz.

Un groupe raidz peut avoir une parité simple, double ou triple, signifiant que le groupe raidz peut survivre à 1, 2, ou 3 erreurs, respectivement, sans perte de données. raidz est un alias de raidz1.

Un group raidz avec N disques de taille X avec P disques de parité peut maintenir approximativement (N-P)*X octets et peut résister à P pannes de périphériques avec que l'intégrité des données soit compromise. Le nombre minimum de périphériques dans un group raidz est un de plus que le nombre de disque de parité. Le nombre recommandé est entre 3 et 9 pour améliorer les performances.

spare Un pseudo vdev spécial qui conserve la trace de disques de rechange à chaud pour un pool.

log Un périphérique de log séparé. Si plus d'un périphérique de log sont spécifiés, les écritures sont load-balancés entre les disques. Les périphériques de log peuvent être en miroir. Cependant, les vdev raidz ne sont pas supportés.

cache Un périphérique utilisé comme cache de données. Un périphérique cache ne peut pas être configuré comme miroir ou groupe raidz.

Les périphériques virtuels ne peuvent pas être imbriqués, donc un périphérique miroir ou raidz ne peut contenir que les fichiers et des disques. Les miroirs de miroirs ou d'autres combinaisons ne sont pas permises.

Un pool peut avoir n'importe quel nombre de périphériques virtuels en haut de la configuration ("root vdevs"). Les données sont dynamiquement distribués dans tous les périphériques de haut niveau pour load-balancer dans les périphériques. Comme avec les nouveaux périphériques virtuels, ZFS place automatiquement les données dans les nouveaux périphériques disponibles.

Les périphériques virtuels sont spécifié une à la fois dans la ligne de commande, séparés par des espaces blancs. Les mots clé "mirror" et "raidz" sont utilisés pour distinguer où un groupe se termine et où un autre commence.

Par exemple, pour créer 2 vdev root, chaque en miroir de 2 disques :

`zpool create mypool mirror c0t0d0 c0t1d0 mirror c1t0d0 c1t1d0`

Erreur disque et récupération

ZFS supporte plusieurs mécanismes pour gérer les erreurs disques et la corruption de données. Toutes les métadonnées et les données sont hashé, et ZFS répare automatiquement les mauvaises données depuis une bonne copie quand une corruption est détectée.

Pour pouvoir utiliser ces fonctionnalités, un pool doit utiliser une certaine forme de redondance, en utilisant soit des groupes miroir ou raidz. Bien que ZFS supporte les configuration non-redondantes, où chaque vdev root est simplement un disque ou un fichier, c'est fortement découragé.

Le status de santé d'un pool est décrit par un de ces 3 états : online, degraded, faulted. Un pool online a tous ses périphériques fonctionnant normalement. Un pool dégradé est un pool dans lequel un ou plusieurs périphériques sont en échec, mais les données continuent d'être disponibles grâce à la configuration de la redondance. Un pool faulted est un pool qui a une corruption de méta-données, un ou plusieurs périphériques en erreur, et des réplicas insuffisant pour continuer à fonctionner.

La santé du vdev top-level, tel un mirror ou un raidz, est potentiellement impacté par l'état de ses vdev associés, ou ses périphériques. Un vdev top-level ou un composant périphérique est dans un de ces états :

DEGRADED

- Un ou plusieurs vdev top-level sont en état dégradés à cause d'un ou plusieurs composants périphérique offline. Des réplicas suffisants existent pour continuer à fonctionner.
- Un ou plusieurs composants périphérique et est en état dégradé ou faulted, mais des réplicas suffisants existent pour continuer à fonctionner. Les conditions sous-jacentes sont les suivantes :
 - Le nombre d'erreurs de checksums excède le niveau acceptable et le périphérique est dégradé en indication que quelque chose ne va pas. ZFS continue d'utiliser le périphérique si nécessaire.
 - Le nombre d'erreur E/S excède le niveau acceptable. Le périphérique ne pourrait pas être marqué en faulted parce qu'il y'a suffisamment de réplicas pour continuer à fonctionner.

FAULTED

- Un ou plusieurs vdev top-level sont à l'état faulted à cause d'un ou plusieurs composants périphérique offline. Les réplicas existant sont insuffisant pour continuer à fonctionner.
- Un ou plusieurs composant périphérique sont à l'état faulted, et des réplicas existant sont insuffisant pour continuer à fonctionner. Les conditions sous-jacentes sont les suivantes :
 - Le périphérique peut être ouvert, mais son contenu ne matche par les valeurs attendues
 - Le nombre d'erreur d'E/S excède un niveau acceptable et le périphérique est faulted pour empêcher son utilisation.

OFFLINE

- Le périphérique a été explicitement mis offline par la commande `zpool offline`

ONLINE

-
- Le périphérique est online et fonctionnel

REMOVED

- Le périphérique a été physiquement supprimé alors que le système fonctionnait. La détection de la suppression de disque est dépendant du hardware et peut ne pas être supporté par toutes les plateformes.

UNAVAIL

- Le périphérique ne peut pas être ouvert. Si un pool est importé quand un périphérique est indisponible, le périphérique sera identifié par un identifiant unique au lieu de son chemin vu que le chemin n'est jamais correct.

Si un périphérique est supprimé puis ré-attaché plus tard dans le système, ZFS tente de placer le périphérique automatiquement en online. La détection de périphérique attaché est dépendant du hardware et peut ne pas être supporté par toutes les plateformes.

Hot Spares

ZFS permet aux périphériques d'être associés avec des pools en hot spare avec la commande `zpool add` et supprimés avec la commande `zpool remove`. Une fois un spare initialisé, un nouveau vdev spae est créé dans la configuration qui va rester jusqu'à ce que le périphérique original soit remplacé. À ce point, le hot spare deviendra disponible si un autre périphérique échoue.

Si un pool a un spare partagé qui est actuellement utilisé, le pool ne peut pas être exporté vu que d'autres pools peuvent utiliser ce spare partagé.

Tout spare en progression de remplacement peut être annulé en détachant le hot spare. Si le périphérique faulted est détaché, le hot spare assume sa place dans la configuration, et est supprimé de la liste de spare pour tous les pools actifs.

Les spares ne peuvent pas remplacer les périphériques logs

Intent Log

ZFS Intent Log (ZIL) satisfait les pré-requis POSIX pour les transactions synchrones. Actuellement, les bases de données nécessitent que leur transactions soient sur des périphériques de stockage stables en retournant depuis un appel système. NFS et d'autres applications peuvent également utiliser `fsync()` pour s'assurer de la stabilité des données. Par défaut, le intent log est alloué dans les blocks du pool principal. Cependant, il peut être possible d'obtenir de meilleures performances en utilisant des périphériques séparés pour les logs tel que NVRAM ou un disque dédié.

Par exemple :

`zpool create pool c0d0 c1d0 log c2d0`

Plusieurs périphériques de log peuvent également être spécifiés, et ils peuvent être mirrorés. Les périphériques de log peuvent être ajoutés, remplacés, attachés, détachés, et importés/exportés comme partie d'un grand pool. Les logs mirrorés peuvent être supprimés en spécifiant le mirror top-level pour les logs.

Cache Devices

Des périphériques peuvent être ajoutés à un pool de stockage comme périphérique de cache. Ces périphériques fournissent une couche additionnelle de cache entre la mémoire principale et le disque. Pour des opérations de lecture intensive, utiliser des périphériques cache fournis les meilleurs performances.

Pour créer un pool avec des périphériques cache, spécifier un vdev cache avec des périphériques. par exemple :

zpool create pool c0d0 c1d0 cache c2d0 c3d0

Les périphériques cache ne peuvent pas être mirrorés dans une configuration raidz. Si une erreur de lecture est rencontrée dans un périphérique cache, cette lecture est refaite dans le pool originale, qui peut faire partie d'une configuration mirror ou raidz. Le contenu des périphériques cache est considéré volatile.

Processus

Chaque pool importé a un processus associé, nommé `zpool-<poolname>`, les threads dans ce processus sont les threads de traitement E/S du pool, qui manipulent la compression, checksums, et d'autres tâches pour toutes les opérations d'E/S associées avec le pool. Ce processus existe pour fournir une visibilité dans l'utilisation CPU. L'existence de ce processus est une interface instable.

Propriétés

Chaque pool a de nombreuses propriétés. Certaines propriétés sont des statistiques lecture-seule alors que d'autres sont configurables. Les propriétés lecture-seules sont :

alloc Quantité d'espace de stockage dans le pool qui a été physiquement alloué.

capacity % de l'espace du pool utilisé.

deduplication Ration de déduplication spécifiée pour un pool, exprimé comme multiplicateur. ex : 1.76 indique que 1.76 unités de donnée sont stockées mais seulement 1 unité de l'espace disque est consommé.

free Nombre de blocks dans le pool qui ne sont pas alloués.

guid Identifiant unique pour le pool

health Santé actuelle du pool.

size Taille totale du pool de stockage

Ces propriétés d'utilisation de l'espace reportent l'espace physique actuellement disponible dans le pool. l'espace physique peut être différent de la taille totale d'espace que tout dataset peut utiliser actuellement. La quantité d'espace utilisé dans une configuration raidz dépend de ses caractéristiques. En plus, ZFS réserve de l'espace en interne que la commande `zfs` prend en compte, mais pas la commande `zpool`. Pour les pools non-pleins d'une taille raisonnable, ces effets devraient être invisibles. pour les petits pools, ou les pools qui sont presque pleins, ces écarts peuvent devenir plus perceptibles.

Les propriétés suivantes peuvent être définies à la création :

ashift Exposant de taille de secteur du pool, en puissance de 2. Les opérations I/O seront alignées à cette taille. Additionnellement La taille d'écriture disque minimum sera définie à la taille spécifiée, ce qui représente un compromis espace/performance. Le cas typique pour définir cette propriété est quand les performances sont importantes et que les disques utilisent des secteurs 4KiB mais reportent des secteurs de 512 octets à l'OS pour des raisons de compatibilité ; dans ce cas, définissez `ashift=12` ($1 \ll 12 == 4096$). Vu que les grands disques ont des secteurs 4K depuis 2011, ZFS utilise `ashift=12` par défaut pour tous les disques supérieurs à 512Go. Pour des raisons de performances, la taille de secteur devrait être égale ou supérieure à la taille de secteur.

Les propriétés suivantes peuvent être définies à la création et à l'import :

altroot Répertoire root alternatif. Si défini, ce répertoire est ajouté à tout point de montage dans le pool. Cela peut être utilisé en examinant un pool inconnu où les points de montage ne peuvent pas être validés, ou dans un environnement de boot alternatif, où les chemins typiques ne sont pas valides. `altroot` n'est pas une propriété persistante. Elle est valide seulement quand le système est up.

Les propriétés suivantes peuvent être définies à la création, à l'import et avec la command `zpool set` :

- autoexpand=on | off** Contrôle l'expansion automatique du pool quand le LUN sous-jacent grandis. À on, le pool sera redimensionné en accord avec la taille du périphérique étendu. Si le périphérique fait partie d'un mirror ou un raidz, tous les périphériques dans le groupe doivent être étendus avant que le nouvel espace soit disponible dans le pool. (défaut : off).
- autoreplace=on | off** Contrôle le remplacement automatique du périphérique. À off, le remplacement de disque doit être initialisé par l'administrateur en utilisant `zpool replace`. à on, tout nouveau périphérique trouvé dans le même emplacement physique que l'ancien disque appartenant au pool est remplacé et formaté automatiquement. (défaut : off)
- bootfs=pool/dataset** Identifie le dataset bootable par défaut pour le pool root. Cette propriété est prévue pour être définie principalement par les programmes d'installation et de mise à jours.
- cachefile=path | none** Contrôle l'emplacement où la configuration du pool est cachée. Découvrir tous les pools au démarrage du système nécessite une copie cachée des données de configuration stockée dans le système de fichier racine. Tous les pools dans ce cache sont automatiquement importés au boot du système. Certains environnements, tels que l'installation et le clustering, nécessitent de cacher cette information dans un emplacement différent et ces pools ne sont pas automatiquement importés. Définir cette propriété met en cache la configuration du pool dans un emplacement différent et peut être importé ultérieurement avec `zpool import -c`. Définir la valeur spécial `none` crée un pool temporaire qui n'est jamais cacé, et la valeur spéciale `""` utilise l'emplacement par défaut. Plusieurs pools peuvent partager le même fichier de cache, mais le kernel détruit et recrée ce fichier quand les pools sont ajoutés ou recréés. quand le dernier pool utilisant `cachefile` est exporté ou supprimé, ce fichier est supprimé.
- delegation=on | off** Contrôle si un utilisateur non-privilégié à accès basé sur les permissions de dataset définis dans le dataset.
- failmode=wait | continue | panic** Contrôle le comportement du système dans l'éventualité d'une erreur catastrophique du pool. Cette condition est typiquement un résultat d'un perte de connectivité aux disques sous-jacents, ou une erreur de tous les disques dans le pool. Le comportement est déterminé comme suit :
- wait** Bloque les accès E/S au pool jusqu'à ce que la connectivité au disque soit récupéré et que les erreurs aient été validés. Un pool reste en wait jusqu'à ce que le problème du périphérique soit résolu. C'est le mode par défaut.
 - continue** Retourne EIO à toute nouvelle demande d'écriture mais permet de lire tous périphérique en cour de fonctionnement.
 - panic** Affiche un message à la console et génère un crash dump système.
- listsnapshots=on | off** Contrôle si les informations sur les snapshots associés avec ce pool sont affichés quand `zfs list` est lancé sans l'option `-t`. (défaut : off).
- version=version** Version courante du pool. Peut être augmenté, mais pas diminué. La méthode préférée pour mettre à jours les pools est avec la commande `zpool upgrade`. Cette propriété peut être tout nombre entre 1 et la version courante reportée par `zpool upgrade -v`.

Sous-commandes

Toutes les sous-commandes qui modifient l'état sont loggés de manière persistante dans le pool dans leur forme original.

zpool add [-fn] [-o property=value] pool vdev ... Ajoute les périphériques virtuel spécifiés dans le pool donné.

- f** Force l'utilisation de `vdev`, même s'il apparaît en utilisation ou spécifie un conflit au niveau de la réplication.
- n** Affiche la configuration qui sera utilisée sans ajouter le `vdev`.
- o property=value** Définis les propriétés du pool donné (`ashift`). N'ajoute pas de disque qui est actuellement configuré comme périphérique quorum à un `zpool`. Une fois un disque dans le pool, ce disque peut être configuré comme périphérique quorum

zpool attach [-f] [-o property=value] pool device new_device Attache un nouveau périphérique à un `zpool` existant. Le périphérique existant ne peut pas faire partie d'une configuration `raidz`. Si `device` ne fait pas partie actuellement d'une configuration `mirror`, `device` se transforme automatiquement en `mirror 2-way` de `device` et `new_device`. Si `device` fait partie d'un `mirror 2-way`, attache `new_device` et crée un `mirror 3-way`, et ainsi de suite.

- f** Force l'utilisation de `new_device`, même s'il apparaît en cours d'utilisation.
- o property=value** Définit les propriétés du pool donné (`ashift`).

zpool clear [-F [-n]] pool [device] ... Efface les erreurs disques dans un pool. Si aucun argument n'est spécifié, toutes les erreurs périphériques seront effacés. Si un ou plusieurs périphériques sont spécifiés, n'efface les erreurs que de ces périphériques.

-F Initie le mode récupération pour un pool inouvable. Tente d'annuler les dernières transactions dans le pool pour le retourner à un état normal. Tous les pools endommagés ne peuvent pas être récupérés avec cette option.

-n Utilisé en combinaison avec -F. Vérifie si les transactions annulées rendraient le pool ouvrable, mais n'annule rien.

zpool create [-fn] [-o property=value] ... [-O file-system-property=value] ... [-m mountpoint] [-R root] pool vdev ... Créé un pool contenant les périphériques virtuels spécifiés dans la ligne de commande. Le nom du pool doit commencer avec une lettre, et peut contenir des caractères alphanumériques, "_", "-" et ".". Les noms de pool mirror, raidz, spare et log sont réservés, comme les noms commençant par c[0-9]. La commande vérifie que chaque périphérique est accessible et non utilisés.

La commande vérifie également que la stratégie de réplication pour le pool est consistante. Une tentative de combiner des stockage redondant et non-redondants dans un seul pool, ou de mixer des disques et des fichiers, cause une erreur sauf si -f est spécifié.

L'utilisation de disques de taille différentes dans un simple raidz ou mirror est aussi marqué en erreur sauf si -f est spécifié.

Le point de montage par défaut est /<pool>. Le point de montage ne doit pas exister ou doit être vide.

-f Force l'utilisation des vdev, même s'ils apparaissent utilisés ou en conflit de niveau de réplication.

-n Affiche la configuration qui serait utilisée sans créer le pool.

-o property=value [-o property=value] ... Définis une propriété du pool

-O file-system-property=value

[-O file-system-property=value] ... Définis des propriétés de système de fichier dans le système de fichier root du pool.

-R root Équivalent à "-o cachefile=none,altroot=root"

-m mountpoint Définis le point de montage pour le dataset root. Le point de montage doit être un chemin absolu, legacy, ou none.

zpool destroy [-f] pool Détruit le pool donné, libérant les périphériques pour d'autres utilisations. Cette commande tente de démonter tout dataset actif avant de détruire le pool.

-f Force un dataset actif contenu dans le pool à être démonté.

zpool detach pool device Détache device d'un miroir. Cette opération est refusée s'il n'y a pas d'autre replica valide.

zpool export [-f] pool ... Exporte les pools donnés du système. Tous les périphériques sont marqués comme exportés, mais sont considérés en utilisation par d'autres sous-systèmes. Les périphériques peuvent être déplacés entre les systèmes et importés tant que le nombre de périphériques suffisants sont disponibles.

Avant d'exporter le pool, tous les datasets dans le pool sont démontés. Un pool ne peut pas être exporté s'il a un spare partagé qui est actuellement utilisé. Pour que les pools soient portables, vous devez donner à zpool des disques entiers, pas seulement les slices, pour que zfs puisse labeliser les disques avec des labels EFI portables.

-f Force à démonter les datasets.

zpool get "all" | property [...] pool ... Récupère la liste donnée de propriété pour les pools spécifiés. Ces propriétés sont affichées avec les champs suivants :

name Nom du pool

property nom de la propriété

value valeur de la propriété

source Source de la propriété.

zpool history [-il] [pool] ... Affiche l'historique des commandes des pools spécifiés.

-i Affiche les événements ZFS loggés en interne en plus des événements initiés par l'utilisateur.

-l Affiche les logs au format long

zpool import [-d dir | -c cachefile] [-D] Liste les pools disponibles à l'import. Si l'option -d n'est pas spécifiée, cette commande recherche les périphériques dans /dev/dsk. Si le périphérique apparaît comme faisant partie d'un pool exporté, cette commande affiche des informations sur le pool. Les pools détruits, les pools qui ont été précédemment détruits avec zpool destroy, ne sont pas listés sauf si -D est spécifié.

-c cachefile Lit la configuration depuis le cachefile donné qui a été créé avec la propriété cachefile.

-d dir Recherche les périphériques ou fichiers dans dir. Peut être spécifié plusieurs fois.

-D Liste les pools détruits uniquement.

zpool import [-o mntopts] [-o property=value] ... [-d dir | -c cachefile] [-D] [-f] [-R root] [-F [-n]] -a Importe tous les pools trouvés dans les répertoires de recherche. Identiquement à la commande précédente, excepté que tous les pools avec un nombre suffisant de périphériques sont importés. Les pools détruits, les pools qui ont été précédemment détruits par zpool destroy, ne sont pas importés sauf si -D est spécifié.

- o **mntopts** Liste séparée par des ',' d'options de montages à utiliser en montant les datasets dans le pool.
- o **property=value** Définis les propriétés spécifiées dans le pool importé.
- c **cachefile** Lit la configuration depuis le cachefile donné qui a été créé avec la propriété cachefile.
- d **dir** Recherche les périphériques ou les fichiers dans dir. Peut être spécifié plusieurs fois. Incompatible avec -c
- D Importe les pools détruits uniquement. -f est requis
- f Force l'import, même si le pool apparaît potentiellement actif.
- F Mode récupération pour un pool non-importable. Tente de retourner le pool à un état importable en annulant les dernière transactions
- a Recherche et importe tous les pools trouvés
- R **root** Définis la propriété cachefile à none, et altroot à root.
- n Utilisé avec -F. Détermine si un pool non-importable peut être importable, mais ne tente pas la récupération.

zpool import [-o mntopts] [-o property=value] ... [-d dir | -c cachefile] [-D] [-f] [-R root] [-F [-n]] pool | id [newpool]
 Import un pool spécifique. Un pool peut être identifié par son nom ou l'identifiant numérique. Si newpool est spécifié, le pool est importé en utilisant le nom newpool. Sinon, il est importé avec le même nom que son nom exporté.

Si un périphérique est supprimé de son système sans utiliser zpool export avant, le périphérique apparaît comme potentiellement actif. Il ne peut pas être déterminé si c'était un export échoué, ou si le périphérique est vraiment en utilisation dans un autre hôte. Pour importer un pool dans cet état, l'option -f est requise.

- o **mntopts** Liste séparée par des ',' d'options de montages à utiliser en montant les datasets dans le pool
- o **property=value** Définis les propriétés spécifiées dans le pool importé.
- c **cachefile** Lit la configuration depuis le cachefile donné qui a été créé avec la propriété cachefile.
- d **dir** Recherche les périphériques ou les fichiers dans dir. Peut être spécifié plusieurs fois. Incompatible avec -c
- D Importe les pools détruits uniquement. -f est requis
- f Force l'import, même si le pool apparaît potentiellement actif.
- F Mode récupération pour un pool non-importable. Tente de retourner le pool à un état importable en annulant les dernière transactions
- a Recherche et importe tous les pools trouvés
- R **root** Définis la propriété cachefile à none, et altroot à root.
- n Utilisé avec -F. Détermine si un pool non-importable peut être importable, mais ne tente pas la récupération.

zpool iostat [-T u | d] [-v] [pool] ... [interval [count]] Affiche des statistiques d'E/S pour les pool donnés. En donnant un interval, rafraîchis toutes les interval secondes. Si aucun pool n'est spécifié, affiche pour tous les pools dans le système. Si count est spécifié, la commande se termine après count reports.

- T **u | d** Affiche un horodatage.
- v Statistiques plus complètes.

zpool list [-H] [-o props [...]] [pool] ... Liste les pools donnés avec un status de santé et d'utilisation de l'espace. Sans argument, tous les pool dans le système sont listés.

- H Mode scripté. N'affiche pas les headers, et sépare les champs pas une tabulation.
- o **props** Liste séparée par des ',' de propriétés à afficher. (name, size, allocated, free, capacity, health, altroot).

zpool offline [-t] pool device ... Place les disques spécifié offline. Cette commande n'est pas applicable pour les caches et spares.

- t Temporaire. Au reboot, le disque sera replacé à son état précédent.

zpool online [-e] pool device... Place un disque online. Cette commande n'est pas applicable pour les caches et spares.

- e Étend le périphérique pour utiliser tous l'espace disque disponible. Si le disque fait partie d'un mirror ou d'un raidz, tous les périphériques doivent être étendus avec que l'espace soit disponible dans le pool.

zpool remove pool device ... Supprime un périphérique du pool. Cette commande supporte seulement la suppression des hot spare, cache, et log.

zpool replace [-f] pool old_device [new_device] Remplace old_device avec new_device. C'est équivalent à attacher new_device, et détacher old_device. La taille de new_device doit être supérieur ou égal à la taille minimum de tous les périphériques dans un mirror ou raidz.

`new_device` est requis si le `pool` n'est pas redondant. Si `new_device` n'est pas spécifié, son défaut est `old_device`. Cette forme de remplacement est utile après qu'un disque existant ait crashé et a été physiquement remplacé. Dans ce cas, le nouveau disque peut avoir le même `/dev/dsk` que l'ancien périphérique, même s'il a une taille différente.

-f Force l'utilisation de `new_device`, même s'il apparaît en utilisation.

zpool scrub [-s] pool ... Commence un scrub. Le scrub examine toutes les données dans les pools spécifiés pour vérifier qu'ils checksums correctement. Pour les périphériques répliqués, ZFS répare automatiquement tout dommage découvert durant le scrub. `zpool status` reporte la progression du scrub et affiche des informations sur son avancement.

Le scrubbing et le resilvering sont des opérations très similaires. La différence est que le resilvering examine seulement les données que ZFS sait être hors date (par exemple, en attachant un nouveau disque à un miroir ou en remplaçant un périphérique existant), alors que le scrubbing examine toutes les données pour découvrir des erreurs dûs à des erreurs matériels ou disque.

Vu que le scrubbing et le resilvering sont des opération d'E/S intensifs, ZFS permet seulement un à la fois. Si un scrub est déjà en progression, `zpool scrub` se termine et recommence un nouveau scrub. Si un resilver est un progression, ZFS ne permet pas de démarrer un scrub.

-s Stop le scrubbing

zpool set property=value pool Définis une propriété au pool spécifié.

zpool split [-R altroot] [-n] [-o mntopts] [-o property=value] pool newpool [device ...] Coupe un disque de chaque `vdev` top-level mirroré en un pool et crée un nouveau pool avec les disques splittés. Le pool original doit être fait d'un ou plusieurs miroirs et ne doit pas être dans un processus de resilvering. Cette commande choisi le dernier périphérique dans chaque `vdev` mirror sauf spécification dans la ligne de commande.

En utilisant un argument `device`, `split` inclus les périphériques spécifiés dans un nouveau pool, si des périphériques restent non précisés, assigne le dernier périphérique de chaque miroir `vdev` à ce pool, comme il le fait normalement. Si vous n'êtes pas certain de la sortie de spécité, utiliser `-n` pour s'assure que votre command aura l'effet attendus.

-R altroot Import automatiquement le pool nouvellement créé avec le `split`, en utilisant le paramètre `altroot` spécifié.

-n Affiche ce qu'il va faire, mais ne fait rien

-o property=value Définis les propriétés spécifiées dans le pool importé.

zpool status [-xv] [pool] ... Affiche des status détaillés sur l'état de santé des pools donnés. Si aucun pool n'est spécifié, alors le status de chaque pool dans le système est affiché. Si un scrub ou un resilver est en cours, cette commande reporte le pourcentage fait et le temps estimé.

-x Affiche seulement le status des pools qui ont des erreurs ou sont indisponible.

-v Affiche plus d'information sur les erreurs.

zpool upgrade Affichue tous les pools formatés en utilisant une version ZFS différente. Les anciennes version peuvent continuer à être utilisée mais certaines fonctionnalité ne seront pas disponible. Ce pools peuvent être upgradé en utilisant `zpool upgrade -a`. Les pools qui sont formatés avec une version plus récente sont également affichés, bien que ces pool sont inaccessible au système.

zpool upgrade -v Affiche les versions ZFS supportés par le logiciel courant. Les versions ZFS courantes et tous les versions précédentes supportées sont affichées.

zpool upgrade [-V version] -a | pool ... Upgrade à a dernière version. Une fois fait le pool ne sera plus accessible aux système fonctionnant avec des versions plus anciennes.

-a Upgrade tous les pools

-V version Upgrade à la version spéifiée.

Exemples

Créer un Pool RAID-Z de 6 disques :

```
zpool create tank raidz c0t0d0 c0t1d0 c0t2d0 c0t3d0 c0t4d0 c0t5d0
```

Créer un pool avec 2 miroirs de 2 disques chacuns.

```
zpool create tank mirror c0t0d0 c0t1d0 mirror c0t2d0 c0t3d0
```

Créer un pool en utilisant des slices :

```
zpool create tank /dev/dsk/c0t0d0s1 c0t1d0s4
```

Créer un pool en utilisant des fichiers :

```
zpool create tank /path/to/file/a /path/to/file/b
```

ajouter un miroir à un pool :

```
zpool add tank mirror c1t0d0 c1t1d0
```

Lister les pools disponibles :

```
zpool list
```

Affiche toutes les propriété d'un pool :

```
zpool get all pool
```

Détruit un pool

```
zpool destroy -f tank
```

Exporte un pool :

```
zpool export tank
```

Importe un pool :

```
zpool import
```

```
zpool import tank
```

Upgrader les pools à la dernière version :

```
zpool upgrade -a
```

Gérer un spare :

```
zpool create tank mirror c0t0d0 c0t1d0 spare c0t2d0
```

Si un des disques échoue, le pool sera réduit à un état dégradé. le disque défectueux peut être remplacé avec :

```
zpool replace tank c0t0d0 c0t3d0
```

Une fois les donnée resilvered, le spare est automatiquement supprimé et de nouveau disponible. Le spare peut être supprimé de manière permanente avec :

```
zpool remove tank c0t2d0
```

Créer un pool avec les logs séparés :

```
zpool create pool mirror c0d0 c1d0 mirror c2d0 c3d0 log mirror c4d0 c5d0
```

Ajouter des disques cache à un pool :

```
zpool add pool cache c2d0 c3d0
```

Une fois ajoutés, les disques cache se remplissent avec le contenu de la mémoire principale. En fonction de la taille des disques, cela peut prendre des heures pour les remplir. La capacité et les lectures peuvent être supervisés avec iostat :

```
zpool iostat -v pool 5
```

Supprimer un périphérique log :

```
zpool remove tank mirror-2
```

récupérer un pool en faulted, récupérer un pool cache :

```
zpool clear -F data
```

sinon utiliser :

```
zpool import -F data
```

Codes de sortie

- 0 succès
- 1 Une erreur s'est produite
- 2 Options de ligne de commande invalide.

attributs

```
attribute type _____ -_attribute value  
Availability _____ -____SUNWzfsu  
Interface Stability __-Committed
```