
qemu-img

Permet de créer, convertir et modifier des images offline.

OPTIONS

check [-q] [-f fmt] [--output=ofmt] [-r [leaks | all]] filename

create [-q] [-f fmt] [-o options] filename [size]

commit [-q] [-f fmt] [-t cache] filename

compare [-f fmt] [-F fmt] [-p] [-q] [-s] filename1 filename2

convert [-c] [-p] [-q] [-n] [-f fmt] [-t cache] [-O output_fmt] [-o options] [-s snapshot_id_or_name] [-l snapshot_param] [

info [-f fmt] [--output=ofmt] [--backing-chain] filename

map [-f fmt] [--output=ofmt] filename

snapshot [-q] [-l | -a snapshot | -c snapshot | -d snapshot] filename

rebase [-q] [-f fmt] [-t cache] [-p] [-u] -b backing_file [-F backing_fmt] filename

resize [-q] filename [+ | -] size

amend [-q] [-f fmt] -o options filename

Paramètres

filename Nom du fichier

fmt Format de l'image

--backing-chain énumère les informations sur les fichiers dans une chaîne d'image disque.

size Taille du fichier en octets. Supporte les suffixes "k" or "K" (kilobyte, 1024) "M" (megabyte, 1024k), "G" (gigabyte, 1024M) et T (terabyte, 1024G). "b" est ignoré

output_filename Nom de destination du disque

output_fmt Format de destination

options Liste séparé par des ',' d'options sous la forme nom=valeur. Utiliser "-o ?" pour les options supportées par le format utilisé.

snapshot_param Est le paramètre utilisé pour le snapshot intern, le format est 'snapshot.id=[ID],snapshot.name=[NAME]' ou '[ID_OR_NAME]'

-c L'image doit être compressée(qcow only)

-h Affiche les formats supportés

-p Affiche une barre de progression

-q Mode silencieux. N'affiche que les erreurs.

-S size Indique le nombre consécutif d'octets qui doivent contenir seulement des 0 pour créer une image sparse durant la conversion.

-t cache Spécifie le mode du cache à utiliser avec le fichier de destination.

Paramètres pour la commande snapshot

snapshot est le nom du snapshot à créer, appliquer ou supprimer

- a Applique un snapshot (revient à l'état sauvegardé)
- c Crée un snapshot
- d Supprime un snapshot
- l Liste tous les snapshot dans l'image donné

Paramètres pour pour la commande compare

- f Premier format d'image
- F Second format d'image
- s Mode strict - échoue sur une taille d'image différente ou une allocation de secteur différent

Paramètres pour pour la commande convert

- n Saute la création du volume cible

Description de commande

check [-f fmt] [--output=ofmt] [-r [leaks | all]] filename Effectue une vérification de consistante sur l'image disque filename. La commande peut sortir dans le format ofmt qui est soit "human" soit "json". Si "-r" est spécifié, qemu-img tente de réparer les inconsistances. "-r leaks" répare seulement les clusters perdus, "-r all" fixe toutes les erreurs. Seul les formats qcow2, qed, et vdi supportent la vérification de consistence. Dans le cas où l'image a des inconsistances, la vérification quitte avec 0. D'autres codes se sortie indique le type d'inconsistance :

- 0 Vérification complète, l'image est consistante
- 1 Vérification non complète à cause d'erreurs interne
- 2 Vérification complète, l'image est corrompue
- 3 Vérification complète, l'image a des clusters perdus, mais n'est pas corrompue
- 63 La vérification n'est pas supportée par le format de l'image.

Si "-r" est spécifié, les codes de sortie représentant l'état de l'image réfère à l'état après sa réparation.

create [-f fmt] [-o options] filename [size]

Crée la nouvelle image disque à la taille et au format spécifié. En fonction du format, des options sont disponibles.

Si l'option `backing_file` est spécifié, l'image enregistrera seulement les différence depuis le `backing_file`. Aucune taille n'est nécessaire dans ce cas. `backing_file` ne sera jamais modifié sauf si vous utilisez `qemu-img commit` ou la commande `monitor commit`.

La taille peut également être spécifiée en utilisant l'option `taille` avec `-o`, qui n'a pas besoin d'être spécifiée séparément dans ce cas.

commit [-f fmt] [-t cache] filename

Envoie les changements enregistrés dans le fichier spécifié dans son image de base. Si le fichier de base est plus petite que le snapshot, alors le `backing file` sera redimensionné à la même taille que le snapshot. Si le snapshot est plus petit que le `backing file`, ce dernier ne sera pas tronqué. Si vous voulez que le `backing file` matche la taille du plus petit snapshot, pour pouvez le tronquer vous-même une fois le `commit` effectué.

compare [-f fmt] [-F fmt] [-p] [-s] [-q] filename1 filename2

Vérifie si les 2 images ont le même contenu. Vous pouvez comparer les images avec différents formats ou paramètres. Le format est sondé sauf si -f et/ou -F sont utilisés.

Par défaut, les images avec une taille différente sont considérés identique si la plus grande image ne contient seulement des secteurs non-alloués ou remplis de 0 à la fin de l'image. En plus, si un secteur n'est pas alloué dans une image et contient seulement des 0 dans l'autre, il est évalué identique. Vous pouvez utiliser le mode strict en spécifiant l'option -s, ce qui fait échouer dans le cas où la taille des images diffèrent ou si un secteur est alloué dans une image et pas dans l'autre.

compare quitte avec le code de sortie 0 si les images sont identiques, et 1 sinon. D'autres codes de sortie signifient que des erreurs ont été rencontrés :

- 0 Les images sont identiques
- 1 Les images diffèrent
- 2 Erreur en ouvrant une image
- 3 Erreur en vérifiant une allocation de secteur
- 4 Erreur en lisant les données

convert [-c] [-p] [-n] [-f fmt] [-t cache] [-O output_fmt] [-o options] [-s snapshot_id_or_name] [-l snapshot_param] [-S sparse_size] filename [filename2 [...]] output_filename

convertis l'image disque filename ou un snapshot snapshot_param en image disque output_filename en utilisant le format output_fmt. Peut optionnellement être compressé (-c) ou chiffré, en fonction du format.

Seul les formats qcow et qcow2 supportent la compression. La compression est lecture seul. Cela signifie que si un secteur compressé est réécrit, la réécriture sera une donnée non compressée. La conversion d'image est également utile pour obtenir des images plus petites en utilisant un format scalaire tel que qcow ou cow : les secteurs vides sont détectés et supprimés de l'image de destination. backing_file permet de forcer l'image de sortie à être créée en copie de l'image de base. le backing_file devrait avoir le même contenu que l'image de base de l'entrée, cependant le chemin, le format, etc. peuvent différer. Si -n est spécifié, la création du volume cible sera ignoré. C'est utile pour des formats tels que rbd si le volume cible a déjà été créé avec les options spécifique du site qui ne peuvent pas être fournis par qemu-img.

info [-f fmt] [-output=ofmt] [-backing-chain] filename

Donne des informations sur l'image disque. À utiliser en particulier pour connaître a taille réservée dans le disque qui peut être différent de la taille affichée. Si les snapshot sont stockés dans l'image disque, elle peuvent être affichées également. La commande peut afficher dans le format ofmt qui est soit "human" soit "json".

Si une image diques a une chaîne backing file, les informations sur chaque image disque dans la chaîne peuvent être récursivement énumérés en utilisant l'option -backing-chain.

Si vous avez un chaîne d'image :

base.qcow2 <- snap1.qcow2 <- snap2.qcow2

Pour énumérer les informations sur chaque image disque :

qemu-img info -backing-chain snap2.qcow2

map [-f fmt] [-output=ofmt] filename

Dump les méta-données de l'image et sa chaîne de backing_file. En particulier, cette commande dump l'état d'allocation de tous les secteurs de filename, ensemble avec le fichier le plus haut dans la chaîne. 2 formats d'options sont possibles. Le format par défaut ('human') dumps seulement les aires non-zero du fichier. Ces parties du fichier sont omis, de même pour les parties qui ne sont pas allouées le long de la chaîne. La sortie de qemu-img va identifier un fichier d'où le donnée peuvent être lues, et l'adresse dans le fichier. Chaque ligne va inclure 4 champs, les 3 premiers sont un héra.

Par exemple :

Offset Length Mapped to File

0 0x20000 0x50000 /tmp/overlay.qcow2

0x100000 0x10000 0x95380000 /tmp/backing.qcow2

signifie que les octets 0x20000 (131072) commençant à l'offset 0 dans l'image sont disponibles dans /tmp/overlay.qcow2 (ouvert en format raw) commençant à l'offset 0x50000 (327680). Les données qui sont compressées, chiffrée ou autre ne sont pas disponible dans le format brute et vont générer des erreur pour le format "human". Notez que les noms de fichiers peuvent inclure des sauts de ligne, ce qui n'est pas sûr pour analyser ce format de sortie dans les scripts.

Le format alternatif json retourne un tableau de dictionnaires au format json. Il va inclure des information similaires dans les champs start, length et offset; il inclus également d'autres informations plus spécifiques :

- Si les secteurs contient une donnée ou non (data); si false, les secteurs sont soit non alloués, ou remplis de 0.
- Si La données est 0 (zero)
- Pour raccourcir la sortie, le fichier cible est exprimé en depth, par exemple, une profondeur de 2 réfère au backing file du backing file de filename.

snapshot [-l | -a snapshot | -c snapshot | -d snapshot] filename

Liste, applique, créer, ou supprime un snapshot de l'image filename.

rebase [-f fmt] [-t cache] [-p] [-u] -b backing_file [-F backing_fmt] filename

Change le backing file au backing_file et (si le format de l'image de filename le supporte) le format du backing file est changé en backing_fmt. Si backing_file est spécifié "", l'image est rebasé dans aucun backing file (ex : elle va exister indépendamment de tout backing file).

Il y a 2 modes différents dans lequel "rebase" peut opérer :

safe mode C'est le mode par défaut et effectue une opération de rebase réel. Le nouveau backing file peut différer de l'ancien et qemu-img rebase va s'occuper de conserver inchangé le contenu visible à l'invité de filename. Pour cela, tous les clusters qui diffèrent entre backing_file et l'ancien backing file de filename sont fusionné dans filename avant de changer de backing file. Noter que le safe mode est plus coûteux, comparable à convertir une image. Ne fonctionne que si l'ancien backing file existe.

unsafe mode qemu-img utilise le mode unsafe si -u est spécifié. Dans ce mode, seul en nom du backing file et le format de filename sont changés sans vérification du contenu du fichier. L'utilisateur doit spécifier le nouveau backing_file correcte, ou le contenu visible de l'invité de l'image sera corrompue. Ce mode est utile pour renommer ou déplacer le backing file ailleurs. Il peut être utilisé sans ancien backing file, par ex, on peut l'utiliser pour fixer un image dont le backing file a déjà été déplacé/renommé.

rebase peut être utilisé pour effectuer une opération diff dans 2 images disque. Cela peut être utile quand un invité est copié ou cloné

Disons que base.img a été clonée en modified.img en la copiant, et cet invité modified.img a été lancé donc il y a quelques changements comparés à base.img. Pour construire une petite image appelée diff.qcow2 qui contient seulement la différence :

qemu-img create -f qcow2 -b modified.img diff.qcow2

qemu-img rebase -b base.img diff.qcow2

modified.img peut être supprimé, vu que base.img + diff.qcow2 contient la même information.

resize filename [+ | -]size

Change l'image disque comme si elle avait été créée avec la taille donnée. Avant d'utiliser cette commande pour réduire une image disque, vous devez utiliser les outils de système de fichier et de partitionnement dans la VM pour réduire les tailles alloués aux systèmes de fichiers et partitions, sinon des données seront perdues. Après avoir utilisé cette commande pour agrandir une image disque, vous devez utiliser les outils de partitionnement et de systèmes de fichiers dans la v pour utiliser le nouvel espace disque.

amend [-f fmt] -o options filename

Modifie les options spécifiques au format de l'image. Tous les formats ne supportent pas cette opération.

Notes

Les formats d'image suivant sont supportés :

raw Format d'image brute, il a l'avantage d'être simple et facilement exportable à d'autres émulateurs.

qcow2 Format d'image qemu, le format le plus versatile. Il a les images les plus petites, supporte le chiffrement, la compression et les snapshots multiples.

Options du format qcow2

compat Détermine la version qcow2 à utiliser. `compat=0.10` utilise le format traditionnel qui peut être lu par tous les qemu. `compat=1.1` active les extensions du format.

backing_file Nom d'une image de base

backing_fmt Formate de l'image de base

encryption Active le chiffrement 128-bits AES-CBC

cluster_size Change la taille de cluster (doit être entre 512 et 2M). Les petites taille améliorent la taille de l'image, et les plus grandes taille améliorent les performance.

preallocation (off, full, falloc, metadata). Alloue tout l'espace disque. Améliore les performances.

lazy_refcounts à on, Les updates de comptage de référence sont reportés pour éviter les E/S de méta-données et améliorer les performances. Intéressant avec `cache = writethrough`. Le compromis est qu'après un crash d'hôte, les tables de comptage de référence doivent être reconstruites, `qemu-img check -r all` est requis et peut prendre du temps.

nocow A on, désactive le COW du fichier. Seulement valide sur brtfs.