

LIBGUESTFS: outils de gestion d'images disques

Table of Contents

- [Installation de libguestfs-tools](#)
- [Création d'un système de fichier avec virt-make-fs](#)
 - [Type de fichier](#)
 - [Table de partition](#)
 - [Allocation d'espace additionnel](#)
- [Montage d'une image avec guestmount](#)

libguestfs est un ensemble d'outils permettant de créer, d'accéder et de modifier les images de disque de machines virtuelles. On peut l'utiliser pour afficher et éditer des fichiers dans des invités, écrire des modifications sur des ordinateurs virtuels, surveiller des statistiques sur les disques utilisés/disponibles, créer des invités, P2V, V2V, effectuer des sauvegardes, cloner des ordinateurs virtuels, construire des ordinateurs virtuels, formater des disques, redimensionner des disques, etc.

libguestfs peut accéder à presque toutes les images disque imaginables.

libguestfs fournit (entre autres choses) une suite d'outils :

- **virt-make-fs**: création d'un système de fichiers à partir d'une archive tar ou de fichiers
- **guestmount**: montage d'un système de fichiers invité sur l'hôte à l'aide de FUSE et de libguestfs
- **Guestfish**: shell et de ligne de commande permettant d'examiner et de modifier les systèmes de fichiers de machines virtuelles.

Installation de libguestfs-tools

installer le jeu d'outils libguestfs sous Linux comme suit.

Sur un système basé sur Debian/Ubuntu:

```
$ sudo apt-get install libguestfs-tools
```

Sur un système basé sur Red Hat/Centos:

```
$ sudo yum install libguestfs-tools
```

Création d'un système de fichier avec virt-make-fs

Virt-make-fs est un outil de ligne de commande permettant de créer un système de fichiers à partir d'une archive tar ou de certains fichiers d'un répertoire. Il peut créer des types de systèmes de fichiers courants tels que ext2/3 ou NTFS.

L'utilisation de base est:

```
virt-make-fs input output.img
```

où input est soit un répertoire contenant les fichiers que l'on souhaite ajouter, soit une archive tar (tar compressé ou tar compressé au format gzip); et output.img est une image disque. Le type d'entrée est détecté automatiquement. L'image disque de sortie par défaut est une image fragmentée ext2 brute.

Format long	Format court	Libellé
--help		Afficher une aide
--floppy		Créer une disquette virtuelle. Actuellement, cela présélectionne la taille (1440K), le type de partition (MBR) et le type de système de fichiers (VFAT). À l'avenir, vous pourrez également choisir la géométrie.
--size=N	-s N	Utiliser l'option -size (ou -s) pour choisir la taille de l'image de sortie. Si cette option n'est pas donnée, alors l'image de sortie sera juste assez grande pour contenir tous les fichiers, sans trop d'espace perdu. Pour choisir un disque de sortie de taille fixe, spécifier un nombre absolu suivi de b/K/M/G/T/P/E en octets, kilo-octets, méga-octets, giga-octets, giga-octets, péta-octets ou Exa-octets. Cela doit être assez grand pour contenir tous les fichiers d'entrée.
--size=+N	-s +N	Pour laisser un espace supplémentaire, spécifier + (signe plus) et un nombre suivi de b/K/M/G/T/P/E en octets, kilo-octets, méga-octets, giga-octets, giga-octets, péta-octets ou Exa-octets. Par exemple: --size=+200M signifie un espace suffisant pour les fichiers d'entrée et (environ) 200 Mo d'espace libre supplémentaires. virt-make-fs estime l'espace libre et ne produira donc pas de systèmes de fichiers contenant précisément l'espace disponible demandé. (Il est beaucoup plus coûteux et fastidieux de produire un système de fichiers disposant précisément de l'espace libre souhaité).
--format=FMT	-F FMT	format d'image disque de sortie. La valeur par défaut est raw (image de disque fragmenté brut).
--type=FS	-t FS	type de système de fichiers de sortie. La valeur par défaut est ext2. Tout système de fichiers supporté en lecture-écriture par libguestfs peut être utilisé ici.
--label=LABEL		étiquette du système de fichiers.
--partition		Si spécifié, cet indicateur ajoute une table de partition MBR à l'image de disque de sortie.
--partition=PARTTYPE		On peut changer le type de table de partition, par exemple. -partition = gpt pour les grands disques. Virt-make-fs définit automatiquement l'octet de type de partition pour MBR.
--verbose	-v	Activer les informations de débogage.
--version	-V	Affiche le numéro de version et quitte.
	-X	Activer la trace libguestfs.

Type de fichier

À peu près tout type de système de fichiers pris en charge par libguestfs peut être utilisé (mais pas les formats en lecture seule tels que ISO9660). Voici quelques-uns des choix les plus courants:

- **ext2**: le type de système de fichiers par défaut.
- **ext3**: les systèmes de fichiers ext3 contiennent un journal, généralement d'une taille comprise entre 1 et 32 Mo.
- **ntfs ou vfat**: Utile si on exporte des données vers un invité Windows.
- **minix**: Surcharge plus faible que ext2, mais certaines limitations concernant la longueur du nom de fichier et la taille totale du système de fichiers.

```
virt-make-fs --type=minix input minixfs.img
```

Table de partition

Virt-make-fs peut éventuellement ajouter une table de partitions au disque de sortie.

Ajouter une partition peut rendre l'image disque plus compatible avec certains systèmes d'exploitation virtualisés qui ne s'attendent pas à voir un système de fichiers situé directement sur un périphérique bloc (Linux ne s'en occupe pas et gérera avec plaisir les deux types).

Par contre, si on a une table de partition, l'image de sortie n'est plus un système de fichiers raw. Par exemple, on ne pourra pas exécuter `fsck` directement sur une image disque partitionnée. (Cependant, des outils libguestfs tels que `guestfish` et `virt-resize` peuvent toujours être utilisés).

Par exemple pour ajouter une partition MBR:

```
virt-make-fs --partition -- input disk.img
```

Si l'image disque en sortie peut être d'une taille égale ou supérieure à un téraoctet, il est préférable d'utiliser une table de partition compatible EFI/GPT:

```
virt-make-fs --partition=gpt --size=+4T --format=qcow2 input disk.img
```

Allocation d'espace additionnel

Contrairement aux formats tels que tar et squashfs, un système de fichiers ne "s'adapte pas" aux fichiers qu'il contient, cependant on peut vouloir laisser de l'espace afin que davantage de fichiers puissent être ajoutés ultérieurement. Virt-make-fs utilise par défaut la réduction de l'espace supplémentaire, mais on peut utiliser l'indicateur `-size` pour laisser de l'espace dans le système de fichiers.

Un autre moyen de laisser de l'espace supplémentaire sans augmenter l'image de sortie est d'utiliser un autre format d'image disque (au lieu du format "brut" par défaut). L'utilisation de `--format=qcow2` utilisera le format d'image natif qemu/KVM qcow2. Cela vous permet de choisir une

grande taille, mais l'espace supplémentaire ne sera pas réellement alloué dans l'image.

Note: on peut également utiliser des commandes locales, notamment `resize2fs` et `virt-resize` pour redimensionner les systèmes de fichiers existants, ou réexécuter `virt-make-fs` pour créer une nouvelle image à partir de zéro.

```
virt-make-fs --format=qcow2 --size=+200M input output.img
```

Montage d'une image avec guestmount

libguestfs supporte pratiquement tous les types d'images disque, y compris qcow2.

Une fois le jeu d'outils libguestfs installé, on peut monter une image qcow2 à l'aide d'un outil en ligne de commande appelé `guestmount`, comme suit.

Attention: lorsqu'une machine virtuelle est en cours d'exécution, on ne doit **pas** monter son image disque en mode "lecture-écriture". Sinon, on risque d'endommager l'image du disque. Ainsi, il est prudent de toujours éteindre une machine virtuelle avant de monter son image disque.

```
$ sudo guestmount -a /chemin/vers/qcow2/image -m <périphérique>  
/chemin/vers/mount/point
```

L'option "-m <périphérique>" permet de spécifier une partition à monter dans l'image disque (par exemple, `/dev/sda1`).



Si on ne sait pas quelles partitions existent dans l'image disque, on peut fournir un nom de périphérique factice. L'outil `guestmount` montrera ensuite les noms de périphériques disponibles.

Par exemple, pour monter `/dev/sda1` dans l'image de disque 'xenserver.qcow2' au point de montage `/mnt`, exécuter la commande suivante.

```
$ sudo guestmount -a /var/lib/libvirt/images/xenserver.qcow2 -m /dev/sda1  
/mnt
```

Par défaut, l'image disque est montée en mode "lecture-écriture". Ainsi, on peut modifier n'importe quel fichier dans `/mnt` après l'avoir monté.

Si on souhaite le monter en mode "lecture seule":

```
$ sudo guestmount -a /var/lib/libvirt/images/xenserver.qcow2 -m /dev/sda1 --  
ro /mnt
```

Pour le démonter:

```
$ sudo guestunmount/mnt
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=notes:libguestfs-tools>

Last update: **2025/02/19 10:59**

