

---

# e2image

Sauvegarder les métadonnées critique d'un système de fichier ext2/3/4 dans un fichier

e2image sauvegarde les métadonnées de système de fichier ext2/3/4 critique dans un fichier. Ce fichier est examiné par dumpe2fs et debugfs, en utilisant l'option -i. Celà permet d'assister un expert dans la récupération catastrophique des systèmes de fichier corrompus. Dans le future, e2fsck sera amélioré pour être capable d'utiliser le fichier image pour aider à récupérer un système de fichier endommagé.

C'est une très bonne idée de créer des fichiers image de tous les systèmes de fichier dans un système et de sauver le layout de partition (qui est généré avec fdisk -l) à intervals réguliers. Au boot, et/ou chaque semaine ou autre. Li fichier image devrait être stocké dans un autre système de fichier pour s'assurer que ces données sont accessibles.

Pour sauver de l'espace disque, e2image créé le fichier image comme fichier sparse, ou au format QCOW2. La taille d'un fichier image ext2 dépend principalement de la taille du système de fichier et du nombre d'inode utilisés. Pour un système de fichier de 10Gio avec 200000 inodes utilisé sur 1,2 millions d'inodes, le fichier image fera environ 35Mio ; un système de fichier de 4Gio avec 15000 inodes utilisés sur 550000, le fichier image fera environ 3Mio. Les fichiers images sont compressibles, un fichier image de 35 Mio sera compressé à 3 ou 4 Mio.

## Restaurer les métadonnées de système de fichier en utilisant un fichier image

L'option -I de e2image permet d'installer les métadonnées stockées dans le fichier image sur le périphérique. Il peut être utilisé pour restaurer les métadonnées dans des situations d'urgence. Attention, si le système de fichier a changé depuis la sauvegarde les données seront perdues.

## Fichiers image raw

L'option -r créé un fichier image brute au lieu d'un fichier image normal. Un fichier image raw diffère d'un fichier image normal de 2 manière. D'abord, les métadonnées du système de fichier est placé dans la bonne position pour que e2fsck, dumpe2fs, debugfs, etc. puis lancer directement l'image raw. Pour minimiser la quantité d'espace disque consommée par un fichier image raw le fichier est créé comme fichier sparse. Ensuite, l'image raw inclus également des blocks indirects et les blocks répertoire, que l'image normale n'a pas.

## Fichiers image qcow2

L'option -Q créé un fichier image qcow2. ce format contient toutes les informations de l'image raw, cependant il n'est pas sparse. l'image qcow2 minimise la quantité d'espace disque en stockant les données dans un format spécial en packant les données ensemble.

## Inclure des données

Normalement, e2image inclus seulement des métadonnées de système de fichier, pas des données de fichier réguliers. L'option -a permet d'inclure toutes les données. Cela donne une image qui est utilisable pour cloner tout le système de fichier. Noter que cette option ne fonctionne seulement avec les format raw et qcow2.

---

# Offset

Normalement un système de fichier commence au début d'une partition et `e2image` est lancé dans cette partition. En travaillant avec des fichiers image, il n'y a pas d'option pour utiliser un périphérique partition, donc il est possible de spécifier l'offset du début du système de fichier directement avec l'option `-o`. Similairement `-O` spécifie l'offset qui devrait être recherché dans la destination avant d'écrire le système de fichier.

Par exemple, avec une image `dd` d'un disque qui contient un système de fichier `ext2` dans une partition commençant à 1Mio, on peut cloner ce système de fichier avec :

```
e2image -aro 1048576 img /dev/sda1
```

On peut cloner un système de fichier dans un fichier image en laissant le premier Mio pour une table de partition :

```
e2image -arO 1048576 /dev/sda1 img
```