
mdadm.conf

Configuration pour mdadm

Le fichier est une collection de mots séparés par un espace. Les mots utilisant un espace sont entre guillemet simple. Une ligne commençant par un espace sont considérés comme la suite de la ligne précédente. Les mots clé sont les suivants :

DEVICE une ligne device liste les périphériques (ou les partitions) qui peuvent contenir un composant d'un array MD. En recherchant les composants d'un array, mdadm scanne ces périphériques. Une ligne device peut contenir plusieurs périphériques séparés par des espaces et peut contenir des wildcard, ou les mots clés containers et partitions. Ce dernier force mdadm à lire /proc/partitions et à inclure tous les périphériques et partitions trouvés. mdadm n'utilise pas les noms dans /proc/partitions mais seulement les numéros majeur et mineur. Il scanne /dev pour trouver le nom qui matche les numéros. Si aucune ligne DEVICE n'est trouvée, assume "DEVICE partitions containers".

par exemple :

```
DEVICE /dev/hda* /dev/hdc*  
DEV /dev/sd*  
DEVICE /dev/disk/by-path/pci*  
DEVICE partitions
```

ARRAY Les lignes array identifient les arrays. le second mot peut être le nom du périphérique où l'array est normalement assemblé, tel que /dev/md1. Si le nom ne commence pas par un '/', il est traité comme relatif à /dev/md. le mot <ignore> peut être donné auquel cas un array qui matche le reste de la ligne n'est jamais automatiquement assemblé. Si aucun nom de périphérique n'est donné, mdadm détermine lui-même un nom approprié. les autres mots identifient l'array, ou l'array comme membre d'un groupe. Si plusieurs identités sont données, un périphérique composant doit matcher toutes les identité pour matcher. Chaque identité a un tag et une valeur. Les tags sont :

uuid= doit correspondre à l'uuid stocké dans le superblock

name= doit correspondre au nom stocké dans le superblock

super-minor= numéro mineur stocké dans le superblock

devices= Liste séparé par ',' de nom de périphérique ou motif. Seul ces périphériques seront utilisés pour assembler l'array. Ces périphériques doivent également être listés dans une ligne DEVICE

level= Niveau de RAID. Pour compatibilité avec la sortie de mdadm -examine -scan

num-devices= nombre de périphériques dans un array actif complet. Pour compatibilité avec la sortie de mdadm -examine -scan

spares= Nombre de spares attendus dans l'array.

spare-group= Nom textuel pour un groupe d'array. Tous les arrays avec le même nom de spare-group sont considérés comme étant du même groupe

auto= indique si mdadm utiliser des array partitionnable ou non, en l'absence de udev. cette option n'est plus nécessaire

bitmap= Fichier dans lequel un bitmap write-intent devrait être trouvé.

metadata= Format des métadonnées de l'array

container= spécifie que cette array est un array membre d'un conteneur. contient un chemin dans /dev ou l'uuid du conteneur

member= Spécifie que cet array est un array membre d'un conteneur. identifie quel membre d'un conteneur appartient l'array.

MAILADDR Donne une adresse email pour les alertes à envoyer quand mdadm fonctionne en mode monitor

MAILFROM Adresse de l'émetteur du mail.

PROGRAM Donne le nom d'un programme à lancer quand mdadm -monitor détecte des événements potentiellement intéressant.

CREATE Donne les valeurs par défaut à utiliser en créant les arrays et les nouveaux membres d'un array. inclus :

owner=

group= id user/group ou noms à utiliser. Défaut : root/wheel ou root/disk.

mode= mode de fichier. Défaut : 0600

auto= correspond à l'option `--auto` de `mdadm`. Donne `yes`, `md`, `mdp`, `part`, pour indiquer combien de périphériques d'entrée manquant devraient être créés

metadata= Forma de métadonnées à utiliser

symlinks=no Normalement, un lien dans `/dev` est créé par `mdadm` avec un nom commençant par `md` ou `md_`. Supprime la création de ce lien

names=yes Permet d'utiliser des noms arbitraires pour les périphériques `md`

bbf=no à `no`, désactive la réservation de l'espace pour la liste des blocks défectueux

HOMEHOST devrait être un nom d'hôte ou les mots clés `<system>` (utilise `gethostname(2)`), `<none>` (n'assume aucun nom) ou `<ignore>` (Permet de définir un nom explicite à la création)

AUTO Une liste de noms de format de métadonnées peuvent être donnés, précédé par un `'+'` ou un `'-'`. `homehost` est également permis équivalent à `all`. (`0.90`, `1.x`, `ddf`, `imsm`).

POLICY Spécifie le comportement automatique permis sur les nouveaux périphériques dans le système et fournit une manière de marquer les spares qui peuvent être déplacés entre les array comme domaine de migration. Les domaines peuvent être définis via les `policy` en spécifiant un nom de domaine pour des chemins dans `/dev/disk/by-path`. Un périphérique peut appartenir à plusieurs domaines. Le domaine d'un array est l'union des domaines de tous les périphériques dans cet array. Un spare peut être automatiquement déplacé d'un array à un autre si le domaine de l'array de destination contient tous les domaines du nouveau disque ou si les array ont le même `spare-group`. Les mots clés dans les `policy` sont :

domain= Chaîne arbitraire

metadata= `0.9` `1.x` `ddf` ou `imsm`

path= glob de fichier correspondant à tout dans `/dev/disk/by-path`

type= `disk` ou `part`

auto= `yes`, `no`, ou `homehost`

action= `include`, `re-add`, `spare`, `spare-same-slot`, ou `force-spare` :

include Permet d'ajouter un disque à un array si les métadonnées dans ce disque matchent cet array

re-add Inclus le périphérique dans l'array s'il apparaît en être un membre ou qu'un membre a été récemment supprimé et l'array a un `bitmap write-intent` pour autoriser la fonctionnalité `re-add`

spare Comme ci-dessus et si le périphérique est vide, il devient un spare s'il y a un array qui est candidat.

spare-same-slot comme ci-dessus et si le slot a été utilisé par un array qui est dégradé et que le périphérique n'a pas de métadonnées, il devient automatiquement membre de l'array (ou son conteneur)

force-spare comme ci-dessus et le disque devient un spare dans tous les autres cas.

Exemple

```
DEVICE /dev/sdc1
DEVICE /dev/hda1 /dev/hdb1
ARRAY /dev/md0 UUID=3aaa0122:29827cfa:5331ad66:ca767371
ARRAY /dev/md1 superminor=1
ARRAY /dev/md2 devices=/dev/hda1,/dev/hdb1
ARRAY /dev/md4 uuid=b23f3c6d:aec43a9f:fd65db85:369432df spare-group=group1
ARRAY /dev/md5 uuid=19464854:03f71b1b:e0df2edd:246cc977 spare-group=group1
ARRAY /dev/md/home UUID=9187a482:5dde19d9:eea3cc4a:d646ab8b auto=part
ARRAY /dev/md9 name='Data Storage'
POLICY domain=domain1 metadata=imsm path=pci-0000:00:1f.2-scsi-* action=spare
POLICY domain=domain1 metadata=imsm path=pci-0000:04:00.0-scsi-[01]* action=include
MAILADDR root@mydomain.tld
PROGRAM /usr/sbin/handle-mdadm-events
CREATE group=system mode=0640 auto=part-8
HOMEHOST <system>
AUTO +1.x homehost -all
```