

/etc/fstab



Le fichier `/etc/fstab` est un fichier de configuration utilisé par le système d'exploitation Linux pour monter automatiquement des systèmes de fichiers. `fstab` signifie "File System Table" (Table des systèmes de fichiers).

Malgré l'amélioration des systèmes d'exploitation, il y a encore des éléments "à l'ancienne" dont nous ne pouvons tout simplement pas nous passer. Les systèmes de fichiers, et par nécessité, les tables de systèmes de fichiers, font partie de ces éléments constants.

Le fichier `/etc/fstab` contient une liste de systèmes de fichiers et de périphériques de stockage qui doivent être montés au démarrage du système. Pour chaque périphérique de stockage, le fichier `/etc/fstab` contient des informations telles que le point de montage (où le système de fichiers doit être monté dans la hiérarchie des répertoires), le type de système de fichiers, les options de montage et les paramètres de sécurité.

La structure de la table elle-même est composée de **6 colonnes**, où chaque colonne désigne un paramètre spécifique et doit être configurée dans l'ordre correct. Les colonnes de la table sont les suivantes, de gauche à droite :

- **Périphérique** : généralement le nom donné ou l'UUID du périphérique monté.
- **Point de montage** : désigne le répertoire où le périphérique est ou sera monté.
- **Type de système de fichiers** : indique le type de système de fichiers en cours d'utilisation.
- **Options** : liste toutes les options de montage actives. Si vous utilisez plusieurs options, elles doivent être séparées par des virgules.
- **Opération de sauvegarde** : (le premier chiffre) il s'agit d'un système binaire où 1 = une sauvegarde à l'aide de l'utilitaire *dump* d'une partition. 0 = pas de sauvegarde. Il s'agit d'une méthode de sauvegarde obsolète et **NE DOIT PAS être utilisée**.
- **Ordre de vérification du système de fichiers** : (deuxième chiffre) ici, nous pouvons voir trois résultats possibles. 0 signifie que `fsck` ne vérifiera pas le système de fichiers. Les nombres supérieurs à celui-ci représentent l'ordre de vérification. Le système de fichiers racine doit être défini sur 1 et les autres partitions sur 2.

Exemple de fichier

`/etc/fstab`

```
UUID=9a1046f4-e5b1-44cc-9e9d-a9b1d8d56be3 /      ext4
defaults,relatime,discard      0 1
UUID=53d697a8-877e-468c-8fa8-fda6dd3f01d1 /boot  ext4
defaults,discard,noatime      0 2
UUID=13d65e2a-21ac-486e-9e48-cc5aac3cc0df /home  ext4      defaults,relatime
0 2
```

```
UUID=005f62eb-a70e-46ab-b457-3c4dcfc22280 /tmp ext4
defaults,discard,noatime 0 2
UUID=09b1e2ca-269e-4971-b91a-498847cb4221 none swap defaults
0 0
#
# Baie 1
rpinas.acegrp.lan:/media/disk_14 /media/disk_14 nfs
defaults,users,auto,nofail,noexec,soft,intr,rsiz=1048576,wsiz=1048576 0 0
```

Les informations contenues dans le fichier `/etc/fstab` sont utilisées par le système d'exploitation pour monter automatiquement les systèmes de fichiers au démarrage, ce qui permet d'accéder facilement aux fichiers stockés sur ces systèmes de fichiers. Sans ce fichier, il serait nécessaire de monter manuellement chaque système de fichiers chaque fois que le système d'exploitation est démarré, ce qui serait fastidieux et source d'erreurs.

Type de système de fichiers

NFS

Le type de système de fichiers **NFS** (Network File System) est un système de fichiers distribué qui permet à un utilisateur d'accéder à des fichiers et des répertoires sur un ordinateur distant comme s'ils étaient locaux. Le système de fichiers **NFS** est largement utilisé dans les environnements informatiques distribués et les réseaux locaux pour partager des fichiers et des données entre plusieurs systèmes. Il est souvent utilisé pour partager des fichiers entre des systèmes **Unix** et **Linux**. Le système de fichiers **NFS** utilise un protocole de communication réseau pour permettre aux clients d'accéder aux fichiers sur le serveur **NFS**.

Bien que le système de fichiers **NFS** soit principalement utilisé sur des systèmes **Unix** et **Linux**, il existe également des implémentations pour Windows. Microsoft propose un client NFS gratuit pour Windows, appelé "**Client for NFS**", qui permet aux utilisateurs Windows d'accéder à des fichiers stockés sur des serveurs NFS distants.

<https://learn.microsoft.com/en-us/windows-server/storage/nfs/nfs-overview>

Les options couramment utilisées pour monter des systèmes de fichiers NFS sont :

- **lookupcache=mode** : spécifie comment le noyau doit gérer le cache des entrées de répertoire pour un point de montage donné.
- **nfsvers=version** : spécifie la version du protocole NFS à utiliser, où version peut être 3 ou 4.
- **noacl** : désactive le traitement des ACL (Access Control Lists).
- **noexec** : désactive le verrouillage de fichiers. Depuis NFS version 3.
- **noexec** : empêche l'exécution de binaires sur les systèmes de fichiers montés. Cela peut être utile si le système de fichiers monté contient des binaires incompatibles avec le système local.
- **nosuid** : désactive les bits set-user-identifier ou set-group-identifier pour empêcher les utilisateurs distants d'obtenir des privilèges supérieurs en exécutant un programme setuid.
- **port=num** : spécifie le numéro de port numérique du serveur NFS. Si num est 0 (la valeur par défaut), mount interroge le service rpcbind de l'hôte distant pour connaître le numéro de port à

utiliser.

- **rsize=num** et **wsize=num** : ces options définissent le nombre maximum d'octets à transférer dans une seule opération de lecture ou d'écriture NFS.
- **sec=flavors** : spécifie les mécanismes de sécurité à utiliser pour accéder aux fichiers sur l'export NFS monté. Exemple : sys, krb5p
- **tcp** : indique au montage NFS d'utiliser le protocole TCP.
- **udp** : indique au montage NFS d'utiliser le protocole UDP.

La configuration optimale dépend de l'environnement spécifique et des besoins du système.

Cependant, voici quelques recommandations générales pour la configuration des options NFS dans /etc/fstab :

- Pour la sécurité, il est généralement recommandé d'utiliser l'option **sec=krb5p** pour chiffrer le trafic NFS et empêcher l'interception ou la modification des données. Cette option nécessite cependant une **infrastructure Kerberos** en place pour l'authentification.
- Pour améliorer les performances, il est souvent conseillé de définir des valeurs personnalisées pour les options **rsize** et **wsize** à une valeur maximale de **1048576** (ou **1 Mo**), qui est la valeur par défaut pour NFS version 3 et 4. Cela permet de maximiser le débit de données et d'optimiser les performances du système : **rsize=1048576,wsize=1048576**.
- Pour éviter les problèmes de verrouillage de fichiers vous pouvez utiliser l'option **noLock**. Lorsqu'un client NFS accède à un fichier sur un serveur, il utilise le verrouillage de fichiers pour empêcher d'autres clients d'accéder ou de modifier le même fichier simultanément. Toutefois, dans certains cas, l'utilisation du verrouillage de fichiers NFS peut entraîner des problèmes de performance ou de fiabilité. En utilisant l'option "noLock", les clients NFS peuvent désactiver la gestion de verrouillage de fichiers NFS pour les fichiers sur un serveur NFS particulier. Cela peut être utile dans des situations où la performance est plus importante que la cohérence des données, ou lorsque le serveur NFS ne supporte pas la gestion de verrouillage de fichiers. **II N'EST PAS conseillé d'utiliser l'option noLock.** L'option "noLock" est spécifique à NFS version 3. Dans les versions antérieures de NFS, le verrouillage de fichiers est toujours utilisé pour empêcher les accès concurrents aux fichiers.
- Pour garantir la disponibilité continue des systèmes de fichiers NFS et éviter les problèmes de blocage, il est recommandé d'utiliser l'option **soft** plutôt que l'option **hard**. Les options **soft** et **hard** sont spécifiques à NFS et ne s'appliquent qu'aux montages NFS. L'option **soft** permet au client NFS de renvoyer une erreur si le serveur ne répond pas ou est indisponible, alors que l'option **hard** fait en sorte que le client continue d'essayer d'atteindre le serveur même si celui-ci est indisponible, ce qui peut entraîner un blocage du système.
- Pour améliorer la tolérance aux erreurs réseau, il est possible d'utiliser l'option **intr** pour interrompre les opérations NFS en cours sur un système de fichiers lorsque le serveur NFS ne répond pas. Si l'option **intr** n'est pas spécifiée, les opérations NFS sont bloquées et le client ne répond plus aux entrées utilisateur tant que le serveur NFS distant ne répond pas ou ne se termine pas. Cela peut être un problème pour les applications interactives ou les scripts qui utilisent NFS, car ils peuvent devenir bloqués si le serveur ne répond pas.

Enfin, il est important de bien comprendre les implications de chaque option et de tester les différentes configurations dans votre environnement avant de les déployer en production.

Les options

noatime

L'option **noatime** dans **fstab** est utilisée pour améliorer les performances du système de fichiers locaux. Normalement, chaque fois qu'un fichier est accédé sur un système de fichiers Linux, l'horodatage de dernière accès (*atime*) est mis à jour. Cela peut entraîner une surcharge inutile du système de fichiers, car même la simple lecture d'un fichier peut entraîner une écriture sur le disque pour mettre à jour l'*atime*.

En utilisant l'option **noatime**, le système de fichiers ne mettra pas à jour la date et l'heure du dernier accès (*atime*) lorsqu'un fichier est accédé. Cela peut considérablement réduire la charge sur le disque et améliorer les performances du système, en particulier pour les systèmes de fichiers très fréquemment utilisés, tels que les systèmes de fichiers racine ou les systèmes de fichiers de base de données.

L'utilisation de l'option **noatime** peut entraîner une perte d'informations d'audit, car le système ne gardera plus une trace du moment où les fichiers ont été accédés pour la dernière fois. Si cela est important pour vos besoins, vous pouvez envisager d'utiliser l'option **relatime** à la place, qui met à jour l'*atime* uniquement si le fichier a été accédé plus récemment que la dernière modification ou l'écriture.

L'option "noatime" est prise en charge par la plupart des systèmes de fichiers Linux modernes, y compris les systèmes de fichiers **ext2**, **ext3**, **ext4**, **XFS**, **Btrfs**, **JFS**, **ReiserFS** et **NTFS** (via le pilote **NTFS-3G**, voir la section [Fichier de configuration du noyau](#)).

Cependant, l'utilisation de l'option **noatime** peut ne pas être appropriée pour tous les types d'applications ou de charges de travail. Par exemple, certains programmes de surveillance de fichiers ou de sauvegarde peuvent avoir besoin de suivre l'horodatage de dernière accès pour fonctionner correctement. Par conséquent, il est recommandé de tester soigneusement l'impact de l'option **noatime** sur votre système avant de l'utiliser en production.

nodev

L'option **nodev** est une option de montage utilisée dans le fichier `/etc/fstab` pour spécifier que les fichiers sur le système de fichiers ne peuvent pas être des fichiers spéciaux de périphérique.

En d'autres termes, si l'option **nodev** est utilisée, les utilisateurs ne pourront pas créer de fichiers spéciaux de périphérique (tels que des fichiers de périphérique de bloc ou de caractère) sur le système de fichiers monté.

L'option **nodev** est souvent utilisée pour des raisons de sécurité, car elle empêche les utilisateurs d'exécuter du code malveillant en créant des fichiers spéciaux de périphérique sur un système de fichiers monté.

Exemple de ligne dans le fichier `/etc/fstab` avec l'option **nodev** :

```
/dev/sda1 /mnt/data ext4 defaults,nodev 0 2
```

Dans cet exemple, le système de fichiers ext4 monté sur /mnt/data ne permet pas la création de fichiers spéciaux de périphérique.

defaults

L'option `defaults` est une option de montage simplifiée qui inclut un ensemble prédéfini d'options de montage pour un système de fichiers spécifique. Les options incluses dans l'option `defaults` varient en fonction du type de système de fichiers.

En général, les options suivantes sont incluses dans l'option `defaults` :

- **rw** : autorise les opérations de lecture et d'écriture sur le système de fichiers
- **suid** : permet l'exécution des fichiers binaires en tant qu'utilisateur propriétaire
- **dev** : autorise la création et la suppression de périphériques spéciaux dans le système de fichiers
- **exec** : permet l'exécution de fichiers binaires
- **auto** : monte le système de fichiers automatiquement au démarrage
- **nouser** : n'autorise pas les utilisateurs ordinaires à monter ou démonter le système de fichiers
- **async** : monte un système de fichiers pour spécifier le mode d'écriture de données asynchrone.

Notez que les options incluses dans l'option `defaults` peuvent varier selon la distribution Linux ou le type de système de fichiers utilisé. Il est important de vérifier la documentation appropriée pour votre système avant de modifier ou de configurer les options de montage.

sync, async

En utilisant l'option `async`, les entrées/sorties peuvent être envoyées à la partition plus rapidement, car le système de fichiers n'attendra pas la confirmation de chaque écriture pour continuer. Cela signifie que les données peuvent être écrites plus rapidement, mais cela peut également entraîner une perte de données en cas de panne du système.

Si l'option `sync` est utilisée, le système de fichiers attendra la confirmation de chaque écriture avant de continuer, ce qui peut ralentir le processus d'écriture, mais garantira l'intégrité des données.

Il est important de noter que l'utilisation de l'option `async` est généralement déconseillée sur des systèmes critiques tels que des serveurs de bases de données ou de fichiers, car cela peut augmenter les risques de perte de données en cas de panne du système ou de coupure de courant.

Pour un SSD, il est généralement recommandé d'utiliser l'option `discard` plutôt que `async` ou `sync` dans le fichier `/etc/fstab`.

L'option `discard` permet au système de fichiers de renvoyer les blocs de données qui ne sont plus nécessaires au SSD, ce qui peut améliorer les performances et la durée de vie du SSD.

user, users

Si un disque n'est pas monté au démarrage du système et que vous souhaitez qu'un utilisateur standard puisse le monter manuellement, vous pouvez utiliser l'option `users` ou `user` dans le fichier `/etc/fstab` pour autoriser les utilisateurs à monter le disque.

L'option `users` permet à n'importe quel utilisateur de monter le système de fichiers, tandis que l'option `user` ne permet que à l'utilisateur propriétaire du point de montage de monter le système de fichiers. Voici comment utiliser ces options:

- Ajoutez l'option `users` à la liste des options pour le système de fichiers dans le fichier `/etc/fstab`, par exemple:

```
/dev/sdb1 /mnt/data ext4 defaults,users 0 0
```

Cela permettra à tous les utilisateurs de monter manuellement le système de fichiers avec la commande `mount /mnt/data`.

- Ajoutez l'option `user` à la liste des options pour le système de fichiers dans le fichier `/etc/fstab`, par exemple:

```
/dev/sdb1 /mnt/data ext4 defaults,user 0 0
```

Cela permettra à l'utilisateur propriétaire du point de montage de monter manuellement le système de fichiers avec la commande `mount /mnt/data`.

auto

L'option `auto` est une option de montage qui spécifie que le système de fichiers doit être automatiquement monté au démarrage du système.

Lorsque l'option `auto` est utilisée pour un système de fichiers spécifique, le système tentera automatiquement de monter le système de fichiers au démarrage, sans intervention de l'utilisateur. Cela peut être utile pour les systèmes de fichiers qui contiennent des données essentielles au fonctionnement du système et doivent être montés automatiquement à chaque démarrage.

nofail

L'option `nofail` est une option de montage qui peut être utilisée dans le fichier `/etc/fstab` pour spécifier qu'un système de fichiers ne doit pas empêcher le démarrage du système si le montage échoue.

Par défaut, lorsqu'un système de fichiers spécifié dans le fichier `/etc/fstab` n'est pas monté avec succès au démarrage, cela peut entraîner des erreurs ou des blocages du système. Cela peut être particulièrement problématique si le système de fichiers contient des données essentielles pour le fonctionnement du système.

En utilisant l'option `nofail` dans le fichier `/etc/fstab`, le système de fichiers sera monté de manière à ne pas bloquer le démarrage du système si le montage échoue. Cela peut être utile dans des situations où un système de fichiers peut ne pas être disponible au démarrage ou où il n'est pas critique pour le fonctionnement du système.

Lorsqu'elle est utilisée en combinaison avec l'option `auto`, l'option `nofail` indique au système de tenter automatiquement de monter le système de fichiers au démarrage, mais de ne pas bloquer le démarrage si le montage échoue.

realtime

Cette option est obsolète et n'est plus utilisée dans les versions récentes des distributions Linux.

L'option `realtime` est utilisée pour les systèmes de fichiers locaux, où elle permet de définir que le système de fichiers doit être monté en mode temps réel pour un accès à haute priorité.

Cependant, pour les systèmes de fichiers NFS, l'option `realtime` n'a pas de sens car les opérations NFS sont déjà exécutées en temps réel. L'utilisation de cette option pourrait donc entraîner des comportements inattendus ou des performances médiocres.

En outre, il est important de noter que le système de fichiers **ext4** ne supporte pas la fonctionnalité en temps réel, contrairement à certains autres systèmes de fichiers tels que **XFS**.

relatime

L'option `relatime` est une option de montage pour les systèmes de fichiers Linux qui permet de mettre à jour le champ d'horodatage d'accès (`atime`) d'un fichier uniquement si celui-ci est plus ancien que le champ d'horodatage de modification (`mtime`) ou de création (`ctime`) du fichier.

L'option `relatime` est une amélioration de l'option `noatime`, qui désactive complètement la mise à jour du champ d'horodatage d'accès lors de la lecture d'un fichier. En comparaison, l'option `relatime` permet de conserver l'horodatage d'accès lorsqu'il est nécessaire pour certaines applications (comme les programmes de surveillance de fichiers), tout en évitant la mise à jour inutile du champ d'horodatage d'accès pour les fichiers qui sont régulièrement consultés.

L'utilisation de l'option `relatime` permet de réduire les opérations d'écriture sur le disque dur par rapport à l'option `atime` par défaut, tout en conservant les horodatages d'accès pour les fichiers qui en ont besoin. Cela peut contribuer à améliorer les performances globales du système de fichiers.

```
/dev/sda1 /mnt/data ext4 defaults,relatime 0 2
```

L'option `relatime` n'est pas prise en charge par tous les systèmes de fichiers et toutes les versions de Linux. Elle est généralement prise en charge par les systèmes de fichiers couramment utilisés, tels que `ext4` et `XFS`, à partir des versions du noyau Linux 2.6.20 et ultérieures.

bind

L'option `bind` dans le fichier `/etc/fstab` permet de monter un répertoire à un emplacement

différent dans le système de fichiers, sans avoir besoin de copier les données.

Lorsque vous utilisez l'option `bind`, vous montez un répertoire existant (source) sur un autre répertoire (destination) qui peut être n'importe où dans le système de fichiers. Le contenu du répertoire source apparaîtra alors dans le répertoire de destination, et toute modification apportée à l'un des deux répertoires sera reflétée dans l'autre.

L'option `bind` est souvent utilisée pour partager des répertoires entre différentes parties du système de fichiers, ou pour créer des liens symboliques persistants entre des répertoires.

Voici un exemple d'utilisation de l'option `bind` dans le fichier `/etc/fstab`:

```
/home/user1/docs /home/user2/shared-docs none bind 0 0
```

Dans cet exemple, le répertoire `/home/user1/docs` est monté sur `/home/user2/shared-docs` en utilisant l'option `bind`. Cela permettra à l'utilisateur `user2` d'accéder aux documents de l'utilisateur `user1` sans avoir à les copier dans son propre répertoire.

Si vous utilisez l'option `bind` pour partager un répertoire entre deux utilisateurs, vous pouvez ajouter l'option `uid` ou `gid` pour spécifier le propriétaire ou le groupe propriétaire du répertoire de destination, afin que les fichiers créés dans ce répertoire appartiennent au bon utilisateur ou groupe.

Voici un exemple d'utilisation de l'option `bind` avec l'option `uid` pour spécifier le propriétaire du répertoire de destination:

```
/home/user1/docs /home/user2/shared-docs none bind,uid=user2 0 0
```

Dans cet exemple, le répertoire `/home/user1/docs` est monté sur `/home/user2/shared-docs` en utilisant l'option `bind` et l'option `uid` pour spécifier que le propriétaire du répertoire de destination est l'utilisateur `user2`.

discard

L'option `discard` dans le fichier `/etc/fstab` est utilisée pour activer la fonction de libération de blocs inutilisés (TRIM ou DISCARD) pour les disques SSD. Lorsqu'un système de fichiers est monté avec l'option `discard`, le système d'exploitation peut informer le disque SSD des blocs de données qui ne sont plus utilisés et qui peuvent être effacés. Cette fonctionnalité de libération des blocs inutilisés peut aider à améliorer les performances et la durée de vie du SSD en évitant la fragmentation des données sur le disque.

Il est important de noter que tous les disques SSD ne prennent pas en charge la fonction TRIM ou DISCARD, et que certains contrôleurs de disque SSD peuvent avoir des performances médiocres lorsqu'ils sont utilisés avec l'option `discard`. Il est donc recommandé de vérifier la compatibilité de votre SSD avant d'utiliser l'option `discard`.

Pour vérifier que votre SSD prend en charge la fonction TRIM ou DISCARD, vous pouvez exécuter la commande suivante en tant que superutilisateur :

```
sudo hdparm -I /dev/sda | grep TRIM
```


Si la commande `hdparm` n'est pas reconnu, installez le programme `hdparm` avec votre gestionnaire de paquets (`dnf`, `yum`, `apt`...).

Assurez-vous de remplacer `/dev/sda` par le nom de périphérique correspondant à votre disque SSD. Si la commande renvoie `Data Set Management TRIM supported (limit 8 blocks)`, cela signifie que votre SSD prend en charge la fonction TRIM.

Vous pouvez également utiliser la commande suivante pour vérifier que la fonction TRIM est activée sur votre système de fichiers :

```
sudo tune2fs -l /dev/sda1 | grep "Discard"
```

Assurez-vous de remplacer `/dev/sda1` par le nom de périphérique correspondant à votre partition contenant le système de fichiers. Si la commande renvoie `Discard options: (0)`, cela signifie que la fonction TRIM est désactivée. Si la commande renvoie `Discard options: (0x0002)`, cela signifie que la fonction TRIM est activée.

```
cedric@rpiapt:~$ sudo hdparm -I /dev/sda | grep TRIM
sudo: hdparm : commande introuvable
cedric@rpiapt:~$ sudo apt install hdparm
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  powermgmt-base
Les NOUVEAUX paquets suivants seront installés :
  hdparm powermgmt-base
0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 117 ko dans les archives.
Après cette opération, 257 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf hdparm armhf 9.60+ds-1 [107 kB]
Réception de :2 http://raspbian.raspberrypi.org/raspbian bullseye/main armhf powermgmt-base all 1.36 [9 604 B]
117 ko réceptionnés en 35s (3 306 o/s)
Sélection du paquet hdparm précédemment désélectionné.
(Lecture de la base de données... 51291 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de .../hdparm_9.60+ds-1_armhf.deb ...
Dépaquetage de hdparm (9.60+ds-1) ...
Sélection du paquet powermgmt-base précédemment désélectionné.
Préparation du dépaquetage de .../powermgmt-base_1.36_all.deb ...
Dépaquetage de powermgmt-base (1.36) ...
Paramétrage de powermgmt-base (1.36) ...
Paramétrage de hdparm (9.60+ds-1) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...
cedric@rpiapt:~$ sudo hdparm -I /dev/sda | grep TRIM
.*          * Data Set Management TRIM supported (limit 8 blocks)
cedric@rpiapt:~$ sudo tune2fs -l /dev/sda1 | grep "Discard"
cedric@rpiapt:~$ sudo tune2fs -l /dev/sda | grep "Discard"
tune2fs: Bad magic number in super-block while trying to open /dev/sda
Found a dos partition table in /dev/sda
cedric@rpiapt:~$ sudo nano /etc/fstab
cedric@rpiapt:~$ sudo reboot
Connection to rpiapt.acegrp.lan closed by remote host.
Connection to rpiapt.acegrp.lan closed.
```

Conseils

Ajouter l'option `discard` pour le système de fichiers `root` et `home`

L'option `discard` permet au système de fichiers de libérer rapidement les blocs inutilisés, améliorant ainsi les performances et la durée de vie des **disques SSD**. Pour ajouter cette option, modifiez les

lignes correspondantes comme suit :

```
LABEL=root_35    /          ext4    defaults,noatime,discard    0 1
LABEL=home_35   /home      ext4    defaults,noatime,discard
0 2
```

Il est important d'aller consulter le chapitre [discard](#).

Utiliser l'option `relatime` plutôt que `noatime`

L'option `relatime` permet au système de fichiers de mettre à jour les horodatages d'accès aux fichiers uniquement si l'accès est plus récent que l'horodatage de modification. Cela permet de réduire le nombre d'écritures sur le disque et d'améliorer les performances. Pour utiliser l'option `relatime`, modifiez les lignes correspondantes comme suit :

```
LABEL=root_35    /          ext4    defaults,relatime,discard    0 1
LABEL=boot_35    /boot      ext4    defaults,relatime    0 1
LABEL=home_35    /home      ext4    defaults,relatime
2                                0
```

Ajouter l'option `noexec` pour les systèmes de fichiers qui ne nécessitent pas l'exécution de fichiers

L'option `noexec` empêche l'exécution de fichiers sur le système de fichiers, ce qui peut aider à renforcer la sécurité. Pour ajouter cette option, modifiez la ligne correspondante pour le système de fichiers `/media/disk_14` comme suit :

```
rpinas.acegrp.lan:/media/disk_14 /media/disk_14 nfs
users,auto,nofail,noexec 0 2
```

Vérifier la configuration du fichier `/etc/fstab`

Il est possible de vérifier la syntaxe et l'intégrité du fichier `/etc/fstab` sans exécuter les instructions en utilisant la commande `findmnt`. Cette commande est disponible sur les systèmes Linux modernes et permet de lister les points de montage actifs et les systèmes de fichiers correspondants.

Pour vérifier la syntaxe du fichier `/etc/fstab`, vous pouvez utiliser la commande suivante :

```
findmnt --verify --evaluate --fstab /etc/fstab
```

Cette commande vérifie la syntaxe du fichier `/etc/fstab` en utilisant l'option `--verify` et affiche les erreurs éventuelles. L'option `--evaluate` permet d'interpréter les expressions du fichier `/etc/fstab` et de les afficher sous forme de chaînes de montage complètes. L'option `--fstab` spécifie le chemin du fichier `/etc/fstab` à vérifier.

Si la commande affiche un message `Succès`, aucune erreur ni avertissement détecté, cela signifie que le fichier `/etc/fstab` est syntaxiquement correct.

La commande `findmnt` ne vérifie pas si les systèmes de fichiers sont réellement accessibles ou s'ils

sont correctement configurés. Elle ne simule que le processus de montage et vérifie la syntaxe du fichier `/etc/fstab`. Il est donc recommandé de vérifier manuellement la configuration de chaque système de fichiers pour s'assurer de son accessibilité et de sa sécurité.

Vérifier les options de montage en cours

Vous pouvez vérifier les options de montage actuelles pour un système de fichiers en utilisant la commande `mount` sur Linux :

- Ouvrez un terminal sur votre système Linux.
- Tapez la commande suivante pour afficher une liste de tous les systèmes de fichiers montés :

```
mount
```

- Recherchez le système de fichiers que vous souhaitez vérifier dans la liste. Par exemple, si vous souhaitez vérifier les options de montage pour le système de fichiers racine (/), recherchez la ligne correspondante dans la liste.
- Les options de montage sont affichées entre parenthèses à la fin de la ligne. Les options sont séparées par des virgules.

```
...  
/dev/sda2 on / type ext4 (rw,noatime,seclabel)  
...  
/dev/sda1 on /boot type ext4 (rw,noatime,seclabel)  
/dev/sdb2 on /home type ext4 (rw,relatime,seclabel)  
/dev/sdc on /home/cedric24c/_tmp type ext4 (rw,relatime,seclabel)
```

-

Si une option est présente, cela signifie qu'elle est activée pour le système de fichiers correspondant. Si une option est absente, cela signifie qu'elle n'est pas activée pour ce système de fichiers.

— [Cédric ABONNEL dit Cédrix](#) & CPT

From:
<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:
<https://www.abonnel.fr/informatique/linux/system/dossiers-remarquables/etc-fstab>

Last update: **2023/06/05 22:01**

