NOM

exports - Systèmes de fichiers à partager (pour NFS en mode noyau)

SYNOPSIS

/etc/exports

DESCRIPTION

Le fichier /etc/exports sert de liste de contrôle d'accès pour les systèmes de fichiers à partager avec les clients NFS. Il est utilisé par **exportfs**(8) pour informer **mountd**(8) et le démon serveur NFS en mode noyau **nfsd**(8).

Le format de ce fichier est similaire à celui du fichier exports de SunOS.
Chaque ligne correspond à un point de montage à partager, suivi d'une liste (aux éléments séparés par des espaces) de clients autorisés à monter le système de fichiers situé en ce point. Chaque client de la liste peut être immédiatement suivi par une liste d'options de partage pour ce client (entre parenthèses, les éléments étant séparés par des virgules). Aucun espace n'est toléré entre un nom de client et sa liste d'options.

En outre, chaque ligne peut définir (après le nom du chemin) le réglage par défaut d'une ou plusieurs options, sous forme de tiret (« - ») suivi d'une liste d'options.
La liste d'options est employée pour tous les partages qui suivent, sur cette ligne seulement.

Les lignes blanches sont ignorées. Un « # » indique un commentaire s'étendant jusqu'à la fin de la ligne. Les entrées peuvent s'étendre sur plusieurs lignes en utilisant la barre oblique inverse (antislash).

Si un nom de partage contient des espaces, il doit être protégé par des apostrophes doubles « " ».
Vous pouvez aussi utiliser la barre oblique inverse (antislash) suivi du code octal à trois chiffres pour protéger tout espace ou autre caractère inhabituel dans un nom de partage.

Pour la prise en compte de vos modifications sur ce fichier, vous devez exécuter exportfs -ra ou re-démarrer le serveur NFS.

Formats des noms de machine

Les clients NFS peuvent être indiqués de plusieurs façons :

Une machine seule

C'est le format le plus courant. Vous pouvez indiquer un hôte par son nom abrégé reconnu par votre résolveur de nom, par son nom de domaine complet, ou par son adresse IP.

Groupes de machines

Les groupes de machines NIS peuvent être utilisés (tels que @group).
Seul le nom court de machine de chacun des membres du groupe est utilisé pour la vérification. Les noms de machines vides, ou ceux contenant un simple tiret (-) sont ignorés.

Caractères jokers

Les noms de machine peuvent contenir les caractères jokers * et ?.
Cela sert à rendre le fichier exports plus compact.
Par exemple, *.cs.toto.edu indique toutes les machines du domaine cs.toto.edu.
Puisque les jokers peuvent aussi remplacer les points dans un nom de domaine, ce motif correspondra aussi à toute machine de n'importe quel sous-domaine de cs.toto.edu.

Réseaux IP

Il est aussi possible de partager des répertoires avec toutes les machines d'un (sous) réseau IP. Il suffit d'indiquer une paire adresse IP / masque de réseau (adresse/masque), en utilisant le format décimal pointé, ou la longueur du masque CIDR (on peut donc ajouter soit « /255.255.252.0 » soit « /22 » à l'adresse du réseau pour obtenir un sous réseau avec 10 bits pour la partie machine). En général, les caractères jokers ne fonctionnent pas avec les adresses IP, bien que cela arrive accidentellement quand les recherches inverses de DNS (reverse DNS lookups) échouent.

Sécurité RPCSEC_GSS

Il est possible d'utiliser les chaînes spéciales « gss/krb5 », «gss/krb5i », ou « gss/krb5p » pour n'accepter que les clients qui utilisent la sécurité rpcsec_gss. Toutefois, cette syntaxe est obsolète, et sur les noyaux linux 2.6.23 et supérieurs, il faut plutôt utiliser l'option de partage « sec= ».

sec=

L'option « sec= », suivie d'une liste (délimitée par des virgules) de niveaux de sécurité, limite le partage aux clients qui utilisent cette sécurité.
Les niveaux de sécurité disponibles sont sys (pas de sécurité cryptographique, par défaut), krb5 (authentification seulement), krb5i (protection de l'intégrité) et krb5p (protection de confidentialité).
En ce qui concerne la négociation des niveaux de sécurité, l'ordre est important, les niveaux préférés devant être listés les premiers. La position de l'option sec= par rapport aux autres options n'a pas d'influence, sauf si ces options s'appliquent

différemment selon le niveau de sécurité. Dans ce cas, il faudra utiliser de multiples options sec=, et les options qui suivent ne s'appliqueront alors qu'à ce niveau de sécurité.
Les seules options utilisables dans ce cas de figure sont « ro », « rw », « no_root_squash », « root_squash », et « all_squash ».

Options générales

exportfs accepte les options de partage suivantes :

secure

Cette option impose l'utilisation d'un port réservé (« IPPORT_RESERVED », inférieur à 1024) comme origine de la requête.
Cette option est activée par défaut.
Pour la désactiver, utilisez insecure.

Rw

Permettre les requêtes en lecture et en écriture sur le volume NFS.
Le comportement par défaut est d'interdire toute requête qui modifierait le système de fichiers.
On peut spécifier explicitement ce comportement par défaut en utilisant l'option « ro ».

async

Permettre au serveur NFS de transgresser le protocole NFS en répondant aux requêtes avant que tous les changements impliqués par la requête en cours n'aient été effectués sur le support réel (par exemple, le disque dur).
L'utilisation de cette option améliore généralement les performances, mais au risque de perdre ou de corrompre des données en cas de redémarrage brutal d'un serveur, suite à un plantage par exemple.

Sync

Ne répondre aux requêtes qu'après l'exécution de tous les changements sur le support réel (voir « async » plus haut).
Dans toutes les versions de nfs-utils jusqu'à la « 1.0.0 » (incluse), c'était l'option par défaut.
Dans toutes les versions suivantes, le comportement par défaut est « sync », et « async » doit être explicitement indiquée si vous en avez besoin. Afin d'aider les administrateurs systèmes à prêter attention à cette modification, exportfs affichera un message d'avertissement si ni « sync » ni « async » ne sont précisées.

no_wdelay

Cette option est sans effet si « async » est déjà active.
Le serveur NFS va normalement retarder une requête d'écriture sur disque s'il suspecte qu'une autre requête en écriture liée à celle-ci soit en cours ou puisse survenir rapidement.
Cela permet l'exécution de plusieurs requêtes d'écriture en une seule passe sur le disque, ce qui peut améliorer les performances.
En revanche, si un serveur NFS reçoit principalement des petites requêtes indépendantes, ce comportement peut réellement diminuer les performances. « no_wdelay » permet de désactiver cette option.
On peut explicitement spécifier ce comportement par défaut en utilisant l'option « wdelay ».

nohide

Cette option est basée sur l'option de même nom fournie dans le NFS d'IRIX.
Normalement, si un serveur partage deux systèmes de fichiers dont un est monté sur l'autre, le client devra explicitement monter les deux systèmes de fichiers pour obtenir l'accès complet.
S'il ne monte que le parent, il verra un répertoire vide à l'endroit où l'autre système de fichiers est monté. Ce système de fichiers est « caché ».
Définir l'option « nohide » sur un système de fichiers empêchera de le cacher, et tout client convenablement autorisé pourra alors se déplacer du système de fichiers parent à celui-ci sans s'en apercevoir.
Cependant, quelques clients NFS ne sont pas adaptés à cette situation. Il est alors possible, par exemple, que deux fichiers de ce système de fichiers apparemment unique aient le même numéro d'inode.
L'option « nohide » ne concerne actuellement que les partages vers les hôtes seuls. Elle ne s'applique pas aux partages vers les groupes de machines, sous-réseaux et ceux utilisant les caractères jokers.
Cette option peut être très pratique dans certains cas, mais elle doit être utilisée avec parcimonie, et seulement après vérification du bon comportement du système client à gérer cette situation.
Cette option peut être désactivée explicitement avec « hide ».

crossmnt

Cette option est semblable à « nohide » mais elle permet aux clients de se déplacer du système de fichiers marqué crossmnt aux systèmes de fichiers partagés montés dessus. Ainsi, si un système de fichiers fils « B » est monté sur un système de fichiers père « A », définir l'option crossmnt sur « A » aura le même effet que d'indiquer « nohide » sur « B ».

no_subtree_check

Cette option neutralise la vérification de sous-répertoires, ce qui a des subtiles implications au niveau de la sécurité,

mais peut améliorer la fiabilité dans certains cas.
Si un sous-répertoire dans un système de fichiers est partagé, mais que le système de fichiers ne l'est pas, alors chaque fois qu'une requête NFS arrive, le serveur doit non seulement vérifier que le fichier accédé est dans le système de fichiers approprié (ce qui est facile), mais aussi qu'il est dans l'arborescence partagée (ce qui est plus compliqué).
Cette vérification s'appelle « subtree_check ».
Pour ce faire, le serveur doit ajouter quelques informations sur l'emplacement du fichier dans le « filehandle » (descripteur de fichier) qui est donné au client. Cela peut poser problème lors d'accès à des fichiers renommés alors qu'un client est en train de les utiliser (bien que dans la plupart des cas simples, cela continuera à fonctionner).
La vérification de sous-répertoires est également utilisée pour s'assurer que des fichiers situés dans des répertoires auxquels seul l'administrateur a accès ne sont consultables que si le système de fichiers est exporté avec l'option « no_root_squash » (voir ci-dessous), et ce, même si les fichiers eux-mêmes offrent un accès plus généreux.
D'une façon générale, un système de fichiers des répertoires personnels («home directories »), qui est normalement partagé à sa racine et qui va subir de multiples opérations de renommage de fichiers, devrait être partagé sans contrôle de sous-répertoires. Un système de fichiers principalement en lecture seule, et qui donc ne verra que peu de modifications de noms de fichiers (« /usr » ou « /var » par exemple) et pour lequel des sous-répertoires pourront être partagés, le sera probablement avec la vérification des sous-répertoires.
On peut explicitement spécifier ce comportement par défaut de vérification des sous-répertoires en indiquant l'option « subtree_check ».
À partir de la version 1.1.0 de nfs-utils, le réglage par défaut sera « no_subtree_check » car la vérification des sous-répertoires (« subtree_checking ») pose souvent plus de problèmes qu'elle n'en résout. Si vous voulez activer la vérification des sous-répertoires comme auparavant, vous devrez explicitement indiquer ce choix dans le fichier « **exports** ». Si vous n'indiquez ni l'une ni l'autre option, « exportfs » vous préviendra que cette modification est en suspens.

insecure_locks

no_auth_nlm

Cette option (les deux noms sont synonymes) indique au serveur NFS de ne pas exiger l'authentification des requêtes de verrouillage (c.-à-d. les requêtes qui utilisent le protocole NLM). Normalement le serveur de NFS doit exiger d'une requête de verrouillage qu'elle fournisse une accréditation pour un utilisateur qui a accès en lecture au fichier. Avec cette option, aucun contrôle d'accès ne sera effectué.
Les premières implémentations de clients NFS n'envoyaient pas d'accréditations lors de requêtes de verrouillage, et nombre de clients NFS encore utilisés sont basés sur ces anciennes implémentations. Utilisez cette option si vous constatez que vous ne pouvez verrouiller que les fichiers en lecture pour tous (« world readable »).
On peut explicitement spécifier ce comportement par défaut de demande d'accréditation pour les requêtes NLM en indiquant les options synonymes « auth_nlm » ou « secure_locks ».

no_acl

Sur certains noyaux spécialement modifiés, et lors de partages de systèmes de fichiers implémentant les ACL, cette option indique à nfsd ne pas dévoiler les ACL aux clients, ainsi ils verront seulement un sous-ensemble de permissions réelles sur le système de fichiers donné. Cette option est efficace pour des partages utilisés par les clients NFSv2 et les anciens clients NFSv3 qui effectuent des vérifications d'accès localement. Les clients NFSv3 actuels utilisent l'appel de procédure distante ACCESS («ACCESS RPC ») afin d'effectuer toutes les vérifications d'accès sur le serveur.
Notez que l'option « no_acl » n'a d'effet que sur des noyaux spécifiquement modifiés pour le gérer, et seulement lors du partage de systèmes de fichiers implémentant les ACL.
Par défaut, le partage implémente les ACL (c.-à-d. Par défaut, « no_acl » est désactivée).

mountpoint= chemin

mp

Cette option permet de ne partager un répertoire que si son montage a réussi.
Si aucun chemin n'est précisé (par exemple « mountpoint » ou « mp ») alors le partage doit également être un point de montage.
Si ce n'est pas le cas, alors le partage n'est pas fait. Ceci vous permet d'être sûr que le répertoire d'un point de montage ne sera jamais partagé par accident si, par exemple, le montage du système de fichiers échouait suite à une erreur de disque dur.
Si un chemin est précisé (c.-à-d. « mountpoint= /chemin » ou « mp= /chemin »), le chemin indiqué doit être un point de montage pour le partage qui est fait.

fsid= num|root|uuid

NFS a besoin de reconnaître chaque système de fichiers qu'il offre en partage. Habituellement, il utilisera un « UUID » pour ce système de fichiers (si le système de fichiers dispose d'un tel « UUID ») ou de l'identifiant du périphérique qui héberge ce système de fichiers (si le système de fichiers est stocké sur un périphérique).
Puisque tous les systèmes de fichiers ne sont pas toujours stockés sur des périphériques, et qu'ils n'ont pas toujours un « UUID », il sera parfois nécessaire de spécifier comment NFS identifiera un système de fichiers. C'est le rôle de l'option « fsid=. »

Dans NFSv4, un système de fichiers particulier est la racine de tous les systèmes de fichiers partagés. Il est défini par « fsid=root » ou « fsid=0 », qui veulent tous deux dire exactement la même chose.
Les autres systèmes de fichiers peuvent être identifiés avec un entier court, ou un « UUID » qui doit comporter 32 caractères hexadécimaux et une ponctuation arbitraire.
Les versions du noyau Linux 2.6.20 et précédentes ne comprennent pas les réglages « UUID », l'utilisation d'un entier court est donc nécessaire pour définir l'option « fsid ». La définition conjointe d'un petit nombre et d'un « UUID » est possible pour une même configuration, ce qui rend possible l'utilisation avec d'anciens ou de nouveaux noyaux.

refer=chemin@serveurNFS[+serveurNFS][:chemin@serveurNFS[+serveurNFS]]
Un client qui se connecte à ce partage se verra proposer le choix d'une autre adresse de système de fichiers parmi celles fournies dans cette liste (Notez que le serveur doit absolument avoir un point de montage sur cette destination, bien qu'il ne soit pas nécessaire qu'il s'agisse d'un système de fichiers différent.
Ainsi, « mount --bind /chemin /chemin » suffit).

replicas=chemin@serveurNFS[+serveurNFS][:chemin@serveurNFS[+serveurNFS]]
Si le client demande d'autres adresses pour ce partage, cette liste de possibilités lui sera proposée (Notez que le mécanisme effectif de réplication du système de fichiers doit être géré ailleurs).

Correspondance d'ID utilisateur (« User ID Mapping »)

nfsd base son contrôle d'accès aux fichiers de la machine serveur sur l'UID et le GID fournis dans chaque requête RPC de NFS. Le comportement attendu par un utilisateur est de pouvoir accéder à ses fichiers sur le serveur de la même façon qu'il y accède sur un système de fichiers normal. Ceci exige que les mêmes UID et GID soient utilisés sur le client et la machine serveur. Ce n'est pas toujours vrai, ni toujours souhaitable.
Bien souvent, il n'est pas souhaitable que l'administrateur d'une machine cliente soit également traité comme le superutilisateur lors de l'accès à des fichiers du serveur NFS. À cet effet, l'UID 0 est normalement transformé (« mapped ») en utilisateur différent : le prétendu utilisateur anonyme ou UID « nobody ». C'est le mode de fonctionnement par défaut (appelé
« root squashing »), qui peut être désactivé grâce à « no_root_squash ».
Par défaut, exportfs choisit un UID et un GID de 65534 pour l'accès « squash ». Ces valeurs peuvent également être définies par les options « anonuid » et « anongid ». Pour finir, vous pouvez faire correspondre toutes les requêtes des utilisateurs à l'UID anonyme en indiquant l'option « all_squash ».

Voici la liste complète des options de correspondance (« mapping ») :

root_squash
Transformer les requêtes d'UID/GID 0 en l'UID/GID anonyme. Notez que ceci ne s'applique à aucun autre UID ou GID qui pourrait également être sensible, tel que l'utilisateur « bin » ou le groupe « staff » par exemple.

no_root_squash
Désactiver la transformation du superutilisateur. Cette option est principalement utile pour les clients sans disque dur.

all_squash
Transformer tous les UID/GID en l'utilisateur anonyme. Utile pour les répertoires FTP publics partagés en NFS, les répertoires de spool de news, etc...
L'option inverse est la suivante.

no_all_squash, qui est celle par défaut.

anonuid et anongid
Ces options définissent explicitement l'UID et le GID du compte anonyme. Cette option est principalement utile pour des clients PC/NFS, dans le cas où vous souhaiteriez que toutes les requêtes semblent provenir d'un seul et même utilisateur. Consultez par exemple la ligne définissant le partage pour « **/home/joe »** dans la section « EXEMPLES » ci-dessous, qui attribue toutes les requêtes à l'utilisateur « 150 » (qui est censé être celui de l'utilisateur « Joe »).

EXEMPLES

```
# exemple de fichier /etc/exports
/           master(rw) trusty(rw,no_root_squash)
/projects   proj*.local.domain(rw)
/usr        *.local.domain(ro) @trusted(rw)
/home/joe   pc001(rw,all_squash,anonuid=150,anongid=100)
/pub        (ro,insecure,all_squash)
/srv/www    -sync,rw server @trusted @external(ro)
```

La première ligne exporte l'ensemble du système de fichiers vers les machines « master » et « trusty ».
En plus des droits d'écriture, toute transformation d'UID est désactivée pour l'hôte « trusty ».
Les deuxième et troisième lignes montrent des exemples de noms de machines avec caractères jokers, et de groupes de machines (c'est le sens de « @trusted»).
La quatrième ligne montre une entrée pour le client PC/NFS, présenté plus haut.
La cinquième ligne partage un répertoire public de « FTP », à toutes les machines dans le monde, en effectuant les requêtes sous le compte anonyme. L'option « insecure » permet l'accès aux clients dont l'implémentation NFS n'utilise pas un port réservé.
La dernière ligne partage un répertoire en lecture-écriture à une machine « server » ainsi qu'à un groupe de machines « @trusted », et en lecture seule pour le groupe de machines « @trusted », tous les trois ayant l'option « sync » activée.

FICHIERS

/etc/exports

VOIR AUSSI

exportfs(8), **netgroup**(5), **mountd**(8), **nfsd**(8), **showmount**(8).