/etc/exports, /etc/exports.d

Fichier d'exports NFS

Le fichier /etc/exports contient une table de systèmes de fichiers physiques locaux d'un serveur NFS accessibles aux clients NFS.

Formats des noms de machine

nom d'hôte un hôte peut être spécifié soit par son nom abrégé reconnu par le resolver, son fqdn, une adresse IPv4 ou IPv6

Réseaux IP il est possible d'exporter des répertoires à tous les hôte dans un sous-réseau en spécifiant <address>/<netmask>, ou le masque peut être sous la forme /255.255.255.0 ou /24.

wildcards Les noms des machines peuvent contenir des caractère '*' ou ' ?' ou une classe de caractère entre crochets

netgroups Les netgroups NIS peuvent être donnés sous la forme @group. Seule la partie hôte du chaque membre du netgroup est vérifié.

anonymous Spécifié par '*' et matche tous les clients.

Si un client matche plus d'une définition, seul le premier match de la liste est considéré.

Sécurité RPCSEC_GSS

Les chaînes "gss/krb5", "gss/krb5i" ou "gss/krb5p" peunvent être utilisés pour restreindre l'accès aux clients utilisant la sécurité rpcsec_gss. Cependant, cette syntaxe est dépréciée, utiliser l'option "sec=" :

L'option sec=, suivi par une liste de mécanismes de sécurité, restreints l'export aux clients utilisant ces mécanismes : sys (défaut = pas de sécurité cryptographique), krb5 (authentification uniquement), krb5i (protection d'intégrité), et krb5p (chiffrement). La négociation suit l'ordre listé, le mécanisme préférentiel est listé en premier.

Options Générales

secure|insecure Cette option nécessite que les requêtes viennent d'un port inférieur à IPPORT_RESERVED (1024). Activée par défaut.

rw|ro Autoriser les requêtes de lecture/écriture ou de lecture seulement.

[a]sync async permet au serveur NFS de violer le protocole NFS et de répondre aux requêtes avant tout changement sur disque. Cela améliore les performance, mais peut cause une perte de données en cas de crash.

[no_]wdelay avec async, Le serveur NFS retarde normalement une requête d'écriture sur disque s'il suspecte qu'une autre requête d'écriture liée est en cours ou va se produire bientôt. Cela permet à plusieurs requêtes d'écritures d'être committed en une opération ce qui améliore les performances. Si un serveur reçoit principalement des petites requêtes non liées, ce comportement peut cependant réduire les performance. no_wdelay permet de désactiver ce comportement.

[no]hide Normalement, si un serveur exporte 2 systèmes de fichiers donc 1 est monté dans l'autre, le client devra monter les 2 systèmes de fichier explicitement. Si monte seulement le parent, il verra un répertoire vide à l'emplacement du 2eme fs. Il est 'caché'. nohide permet de ne pas cacher ce système de fichier. Cependant, certains client NFS ne gèrent pas cette situation, par exemple, il est possible que 2 fichiers aient le même numéro d'inode.

[no] crossmnt Similaire à nohide, mais rend possible l'accès à tous les systèmes de fichier monté dans le système de fichier marqué avec crossmnt. Donc un fs enfant "B" est monté dans un parent "A" a le même effet que nohide sur B. Avec nohide les fs enfant doivent être explicitement exportés, avec crossmnt ce n'est pas nécessaire. Si un enfant d'un fichier crossmnt n'est pas explicitement exporté, il l'est implicitement avec les mêmes options que le parent, à l'exception de fsid=.

[no_] subtree_check no_subtree_check désactive la vérification de l'arborescence, qui a de légères implications de sécurité, mais peut améliorer les performances dans certaines circonstances. Pour effectuer cette vérification, le serveur doit inclure des informations sur l'emplacement du fichier dans le filehandle donné au client. Cela peut poser problème en accédant à des fichiers qui sont renommé pendant qu'un client les ouvre. La vérification du subtree est également utilisée pour s'assurer que les fichiers dans les répertoire que seul root peut accéder peuvent seulement être accedés si le fs est exporté avec no_root_squash., même si le fichier lui-même autorise un accès plus large. un fs home devrait être exporté avec no_subtree_check. Un fs principalement lecture seule ou n'a pas de renommage important de fichiers (ex : /usr, /var), et pour lesquels les sous-répertoires peuvent être exportés, devraient probablement être exporté avec subtree_check.

[in] secure_locks| [no_] auth_nlm Indique que le serveur NFS n'exige pas d'authentification pour les requêtes le lock (qui utilisent le protocole NLM) Normalement le serveur NFS exige qu'une requête lock maintienne un accréditif pour un utilisateur qui a un accès en lecture sur le fichier.

mountpoint=path|mp Permet d'exporter seulement un répertoire si a été monté avec succès. Si aucun chemin n'est donné le point d'export doit être un point de montage. Cela permet de s'assurer que le répertoire sous un point de montage ne sera jamais exporté par accident si, par exemple, le fs échoue le montage dû à une erreur disque.

fsid=num|root|uuid NFS doit être capable d'identifier chaque système de fichier qu'il exporte. Normalement il utilise un UUID ou un numéro de périphérique. Il peut être nécessaire d'identifier explicitement un système de fichier dans certains cas. root ou 0 indiquent le système de fichier identifié comme le parent de tous les autres fs exportés.

nordirplus Désactive la gestion des requêtes READDIRPLUS. NFSv3 uniquement

refer=path@host[+host][:path@host[+host]] Un client référençant le point d'export sera dirigé pour choisir depuis la liste d'emplacement alternative pour le système de fichier.

replicas=path@host[+host][:path@host[+host]] Si le client demande un emplacement alternatif pour le point d'export, il obtiendra cette liste d'alternatifs.

pnfs Active l'extension pNFS, qui permet au client de bypasser le serveur et d'effectuer les opération d'E/S directement dans les périphériques.

Mappage d'ID utilisateur

nfsd base sont contrôle d'accès aux fichier sur le serveur sur l'uid et le gid fournis dans chaque requête RPC NFS. Le comportement normal qu'un utilisateur attend et qu'il peut accéder à ses fichier sur le serveur comme dans un serveur de fichier normal. Cela nécessite que les même uids et gids soient utilisés dans le client et le serveur. Cela n'est pas toujours vrai, ni toujours désirable.

Très souvent, il n'est pas désirable que root sur une machine soit également traitée en root en accédant à des fichiers sur un serveur NFS. l'uid 0 est normalement mappé à un autre id : nobody. Cette méthode, le root squashing, est le mode par défaut.

Par défaut, exportfs choisis un uid et gid à 65534 pour l'accès squashed. Ces valeurs peuvent être changées par les options anonuid et anongid. Finalement on peut mapper toutes les requêtes utilisateur à l'uid anonyme avec l'option all_squash.

[no_] root_squash Map les requête pour uid/gid 0 en uid/gid anonyme

[no_] all_squash Map tous les uid/gid un uid/gid anonyme

anonuid, anongid Définis explicitement l'uid/gid anonyme

Table d'exports Extra

Une fois la lecture de /etc/exports, exportfs lit les fichiers dans /etc/exports.d. Seul les fichiers se terminant par .exports sont lus.

Exemple

Exporter tout le système de fichier aux machines master et trusty, avec accès rw et désactivation du squashing
/ master(rw) trusty(rw,no_root_squash)
/projects proj*.local.domain(rw)
Exporter /usr en lecture seule à tous les hôtes du domain .local.domain, et en lecture/écriture à tous les membres du netgroup trusted
/usr *.local.domain(ro) @trusted(rw)
Exporter /home/joe pour pc001, avec activation du squashing pour tous les utilisateurs, et redéfinition du compte anonyme
/home/joe pc001(rw,all_squash,anonuid=150,anongid=100)
Exporter /pub en lecture seul à tous le monde
/pub *(ro,insecure,all_squash)
Exporter /srv/www en activant l'option sync à la machine "server" et aux membres de @trusted et @external
/srv/www -sync,rw server @trusted @external(ro)
Exporter /foo en IPv4 et IPv6
/foo 2001 :db8 :9 :e54 : :/64(rw) 192.0.2.0/24(rw)
Utiliser les plages entre [] pour spécifier plusieurs noms d'hôte.
/build buildhost[0-9].local.domain(rw)