nbd-server.config

Fichier de configuration pour ndb-server

Description

Un en-tête de section est un nom unique qui est entre **[** et **]**, et dénote le début d'un section ; une section continue jusqu'à ce qu'une autre section commence ou la fin du fichier. La première section doit s'appeler **generic** et est utilisée pour les option globales qui s'appliques à plus d'un export et doit toujours être présent. Chaque autre section maintient un export ; les noms de ces section ne sont pas important excepté qu'ils doivent être unique.

Options pour la section generic

allowlist (bool,optionnel) permet au client de rechercher une liste d'exports depuis ce serveur. **nbd-client -l** permet d'avoir cette liste.

group (string,optionnel) Le nom du groupe sous lequel le service fonctionne. Non spécifié le serveur ne tente pas de changer son GID.

includedir (string,optionnel) Répertoire contenant les fichiers avec l'extension '.conf' qui contiennent d'autres directives de configuration. la section [generic] ne peut pas être dans un de ces fichiers.

listenaddr (string,optionnel) Contient l'IP local sur laquelle le service écoute.

oldstyle (bool,optionnel) à true, le serveur exporte tous les exports sur un port séparé. Dans ce cas, l'option port pour les exports individuels est mandatoire.

port (string,optionnel) port d'écoute du serveur. Défaut : 10809.

user (string,optionnel) le nom du l'utilisateur sous lequel le service fonctionne. Non spécifié le serveur ne tente pas de changer son UID.

Options pour les sections d'export

authfile (string,optionnel) Le nom du fichier d'autorisation pour cet export. Ce fichier devrait contenir une ligne par adresse IP, ou par réseau en notation CIDR. Si le fichier n'existe pas, tout le monde est autorisé à se connecter. Si le fichier existe mais vide, aucun client n'est autorisé à se connecter.

copyonwrite (bool,optionnel) Spécifie si l'export est copy-on-write, qui n'écrit pas dans le fichier maître, mais dans un fichier séparé, qui est supprimé à la déconnexion.

exportname (string,requis) le nom du fichier qui sera exporté. Doit être pleinement qualifié. Utilisé en conjonction avec **temporary**, spécifie un template pour le fichier temporaire concerné, et donc peut être utilisé pour contrôler le répertoire dans lequel il est créé.

filesize (entier,optionnel) désactive l'autodétection de la taille du fichier ou du périphérique block. Doit être spécifié en octets. Si l'option **multifile** est présent, cette option spécifie la taille de tout l'export, pas les fichiers individuels.

flush (bool,optionnel) à true, le serveur informe le client qu'il supporte et souhaite envoyer des requêtes flush quand la couche élévateur les reçois, ce qui cause un fdatasync() ou fsync() si l'option sync est présente dans le stockage. Cela augmente la fiabilité dans le cas des shutdown non propre au prix d'un dégradation des performances.

fua (bool,optionnel) à true, le serveur informe le client qu'il supporte et souhaite envoyer des commandes fua (force specified commands) quand la couche élévateur les reçois, ce qui cause la commande spécifiée d'être synchronisée sur le stockage avec sync_file_range(), ou fdatasync(). Cela augmente la fiabilité dans le cas des shutdown non propre au prix d'un dégradation des performances.

listenaddr (string,optionnel) Adresse IP d'écoute pour cet export, si ordstyle est spécifié dans la section generic

maxconnections (entier,optionnel) Limite le nombre de connections ouverte pour cet export

multifile (bool,optionnel) à true, le serveur recherche les fichiers sous la forme **exportname.integer**, où exportname est le nom du fichier, suivi d'un nombre unique commençant à 0.

port (integer) requis si oldstyle est spécifié dans la section generic, pour d'écoute pour cet export.

postrun (string,optionnel) si spécifié, assume que c'est une commande qui sera lancée quand un client sera déconnecté. Peut être utile pour nettoyer ce que **prerun** a définis, ou logger quelque chose. le code de sortie de postrun est ignoré.

prerun (string,optionnel) si spécifié, cette commande sera exécutée après que le client se soit connecté au serveur, mais avant que le serveur commence à desservir . Si la commande contient **%s**, cette chaîne sera remplacée par le nom du fichier qui est exporté. Si la commande se termine avec un status autre que 0, le serveur assume que l'export a échoué et refuse de le desservir.

readonly (bool,optionnel) Quand cette option est activée, le serveur informe le client qu'il préférerai envoyer des requêtes dans l'order de l'élévateur. Seulement requis quand le serveur n'utilise par l'algorithme d'élévateur, ou cet algorithme est neutralisé.

sdp (bool,optionnel) à true, le serveur utilise SDP (Socket Direct Protocol) pour desservir l'export, au lieu de simplement l'IP. C'est plus rapide, mais nécessite un hardware spécial (comme InfiniBand).

sparse_cow (bool,optionnel) à true, le serveur utilise les fichiers sparses pour implémenter l'option copy-on-write. De tels fichiers prennent moins d'espace qu'ils apparaissent, ce qui permet au serveur de manipuler des fichiers plus grands que n'est le périphériques block.

sync (bool,optionnel) Quand cette option est activée, le serveur va appeler un fsync() après chaque écriture au backend. Cela augmente la fiabilité dans le cas des shutdown non propre au prix d'un dégradation des performances.

temporary (bool,optionnel) Créé un export temporaire avec un nom basé sur le nom de l'export. L'export ne persistera par entre les invocations du serveur. Incompatible avec l'option **multifile**

timeout (string,optionnel) Nombre de secondes qu'une connexion peut être en attente pour cet export. Quand une connexion attend trop longtemps, le serveur force la déconnexion. Désactivé à 0.

transactionlog (string,optionnel) Si spécifié, ce chemin est utilisé pour générer un log de transaction. Ce log est un fichier binaire qui consiste de requêtes envoyés et des réponses reçus par le serveur, mais en excluant les données.

trim (bool,optionnel) quand cette option est activée, le serveur annonce qu'il support la commande NBD_CMD_TRIM pour l'export, cette commande permet au serveur d'annuler la donnée du disque, mais ne l'oblige pas à le faire.

virtstyle (string,optionnel) Définis le style de virtualisation. La virtualisation permet de créer un export qui va desservir un fichier différent en fonction de l'adresse IP qui se connecte. Quand la virtualisation est active, le paramètre exportname nécessite de contenir la chaine **%s**, qui sera remplacé par l'adresse IP du client. Le résultat de cette transformation est utilisé comme nom de fichier à ouvrir. Il y a 4 type de virtualisation supportés :

ipliteral **%s** est remplacé par l'adresse IP du client. (ex : exporname = /export/%s et le client est 192.168.1.100, le serveur tente de desservir /export/192.168.1.100 ; pour une IPv6, /export/2001 :6f8 :32f :0 :0 :0 :0 :39)

iphash idem, mais remplace les point par des "/" (ex /export/192/168/1/100)

cidrhash Nécessite d'ajouter un espace et un nombre après lui, qui sera utilisé comme masque réseau. (ex pour un cidrhash = 16, tente d'ouvrir /export/192.168.0.0/192.168.1.100).

Exemples

Configuration simple :
[generic]
[export]
exportname = /export/blkdev
Pour plus de sécurité, on peut créer un fichier d'autorisation :
[generic]
user = nbd
group = nbd
[export]
exportname = /export/blkdev
authfile = /etc/nbd-server/allow
Et le fichier /etc/nbd-server/allow :
127.0.0.1

192.168.0.0/8
192.168.1.1
Pour être compatible avec les anciens clients :
[generic]
oldstyle = true
[export]
exportname = /export/blkdev
port = 12345