

---

# rsyncd.conf

Fichier de configuration pour le mode service de rsync

## Lancer le service

Le service rsync est lancée avec l'option `-daemon`. Le service doit être lancé avec les privilèges root si l'on souhaite utiliser chroot, pour utiliser un port inférieur à 1024, ou pour définir les propriétaires des fichier. Sinon, il doit simplement avoir la permission de lire et écrire les données appropriées, logs, et fichiers de lock.

Le service peut être lancé via inetd, comme service standalone, ou depuis un client rsync via un shell distant. Via inetd, une ligne dans `/etc/services` :

**rsync 873/tcp**

et une ligne dans `/etc/inetd.conf` :

**rsync stream tcp nowait root /usr/bin/rsync rsyncd -daemon**

## Paramètres globaux

Les premiers paramètres dans le fichier, avant tout `[module]`, sont les paramètres globaux. rsync autorise également l'utilisation d'en nom de module `[global]`. Il est possible d'inclure des paramètres de modules dans la partie globale auquel cas la valeur remplace la valeur par défaut pour ce paramètre

Il est possible de référencer des variables d'environnement dans les valeurs des paramètres, sous la forme `%VAR%`.

**motd\_file** Permet de spécifier un message du jour à afficher au client.

**pid file** Écris le pid dans le fichier

**port** Port d'écoute du service

**address** Address d'écoute du service

**socket options** Options socket pasés à `setsockopt()`

**listen backlog** Valeur backlog quand le service écoute les connexions. Défaut : 5

## Paramètres de module

**comment** Spécifie une description pour le module

**path** Répertoire disponible dans ce module

**use chroot** à true, rsync chroot dans le path avant de commencer le transfert de fichier

**numeric ids** Désactive le mappage des utilisateurs et groupe pou le module

**numge symlinks** modifie les liens comme `-munge-links`

**charset** Jeu de caractère avec lequel les noms des fichiers du module sont stockés. Utiles avec `-iconv`

**max connections** Spécifie le nombre de connexions simultanée maximum

**log file** Log les messages dans le fichier au lieu de syslog

**syslog facility** facilité syslog (auth, authpriv, cron, daemon, ftp, kern, lpr, mail, news, security, syslog, user, uucp, local0-7)

---

**max verbosity** Niveau de verbosité (de 1 à 4)

**lock file** spécifie le fichier à utiliser pour supporter le paramètre max connections. Défaut : /var/run/rsyncd.lock

**read only** Détermine si les clients sont capables d'uploader des fichiers ou non.

**write only** Détermine si les clients sont capables de télécharger des fichiers ou non.

**list** Détermine si ce module est listé quand le client demande un listing des modules disponibles. De plus, à false, le service prétend que le module n'existe pas quand un client est refusé par hosts allow/deny.

**uid** uid pour le service rsync

**gid** gid pour le service rsync

**fake super** à yes, le service fonctionne comme avec l'option -fake-super

**filter** Le service possède sa propre chaîne de filtre qui détermine quels fichiers le client peut accéder

**exclude** Liste de motifs d'exclusion

**include** Liste de motifs d'inclusion

**exclude from** Fichier contenant des motifs d'exclusion

**include from** Fichier contenant des motifs d'inclusion

**incoming chmod** Permet de spécifier un jeu de chaînes chmod qui affectent les permissions de tous les fichiers entrants. Ces changements se produisent après tout les autres calculs de permission

**outgoing chmod** Permet de spécifier un jeu de chaînes chmod qui affectent les permissions de tous les fichiers sortants. Ces changements se produisent en premier, les permissions apparaissent différentes de celles stockées dans le système de fichier

**auth users** Spécifie une liste de règles d'autorisations. Cela peut être une liste d'utilisateurs et peut contenir des caractères wildcard shell. Les usernames et passwords sont stockés dans le fichier spécifié par le paramètre secrets file. Utiliser '@' pour spécifier un groupe. Chaque utilisateur et groupe peut être suffixé par :<deny|rolr|w> pour restreindre les accès. Pour spécifier un utilisateur ou groupe avec un espace dans le nom, commencer la liste avec une ','

**secrets file** Spécifie le fichier qui contient les paires username :password et/ou @group :password utilisés pour l'authentification du module.

**strict modes** Détermine si les permissions dans le fichier secret sont vérifiées. à true, le fichier ne doit pas être lisible par tout le monde

**hosts allow** Spécifie une liste de motifs qui sont matchés avec l'ip ou hostname des clients. Si aucun motif ne match, la connexion est rejetée

**hosts deny** Spécifie une liste de motifs qui sont matchés avec l'ip ou hostname des clients. si un motif match, la connexion est refusée

**reverse lookup** Effectue une recherche inversée sur l'ip du client pour déterminer le nom d'hôte

**forward lookup** Effectue une recherche sur les hôtes spécifiés dans les paramètre host allow/deny.

**ignore errors** Ignore les erreurs d'E/S dans le service en décidant s'il lance la phase delete du transfert.

**ignore nonreadable** Ignore complètement les fichiers qui ne sont pas lisibles par l'utilisateur.

**transfer logging** Active le logging par fichier dans un format similaire à celui utilisé par les services ftp. Le service log toujours le transfert à la fin, donc si le transfert est annulé, rien n'est mentionné dans le fichier de log.

**log format** Spécifie le format utilisé pour logger les transferts de fichier. l'apostrophe permet d'indiquer que la valeur doit être au format human-readable.

**%a** IP distante (mode service uniquement)

**%b** Nombre d'octets transférés

**%B** Bits de permissions du fichier

**%c** Taille totale des checksums de block reçus (seulement pour l'envoi)

**%C** Checksum du fichier complet

**%f** Nom du fichier

**%G** GID du fichier

**%h** Nom d'hôte distant (uniquement pour le mode service)

**%i** Liste itemisée de ce qui est mis à jours

**%l** Longueur en octets du fichier

**%L** La chaîne -> SYMLINK", " => HARDLINK", ou ""

**%m** Nom du module

---

**%M** mtime du fichier  
**%n** nom du fichier, forme courte  
**%o** L'opération (recv, send, ou del)  
**%p** PID du service  
**%P** Chemin du module  
**%T** Date courante  
**%u** L'utilisateur authentifié  
**%U** uid du fichier

**timeout** Remplace le choix du timeout du client.

**refuse options** Spécifie une liste d'options qui sont refusées par le service. Si une de ces options est reçue, le service affiche un message d'erreur et quitte.

**dont compress** motifs spécifiant les fichiers qui ne sont pas compressés

**pre-xfer exec, post-xfer exec** Commande à lancer avant et/ou après le transfert. Si pre-xfer exec échoue, le transfert est annulé. Les variables d'environnement suivantes sont définies :

**RSYNC\_MODULE\_NAME** Nom du module accédé  
**RSYNC\_MODULE\_PATH** Chemin configuré du module  
**RSYNC\_HOST\_ADDR** IP du client  
**RSYNC\_HOST\_NAME** Nom d'hôte du client  
**RSYNC\_USER\_NAME** Nom de l'utilisateur  
**RSYNC\_PID** Identifiant unique pour ce transfert  
**RSYNC\_REQUEST** pre-xfer uniquement : module/chemin spécifié par l'utilisateur  
**RSYNC\_ARG#** Arguments  
**RSYNC\_EXIT\_STATUS** post-xfer uniquement : Valeur de sortie côté serveur.  
**RSYNC\_RAW\_STATUS** post-xfer uniquement : Valeur brut de sortie pour waitpid()

## Directives config

Il y a 2 directives disponibles qui permettent à un fichier de configuration d'incorporer le contenu d'autres fichiers : `&include` et `&merge`. Elles permettent de référencer un fichier ou un répertoire. Elles diffèrent dans la manière dont le contenu des fichiers sont considérés.

`&include` traite chaque fichier plus distinctement, chacun héritant des défauts du fichier parent, en commençant par traiter les défauts/globaux, et laissant les défauts inchangés pour le parsing du reste du fichier parent

`&merge` traite chaque fichier comme s'il était simplement inséré à la place de la directive

L'avantage de `&include` est que l'on peut définir un ou plusieurs modules dans un fichier séparé sans s'inquiéter des effets inattendus entre les fichiers de modules auto-contenus. L'avantage de `&merge` est que l'on peut charger des snippets qui peuvent être inclus dans plusieurs définitions de modules, et que l'on peut définir des valeurs globales qui affectent les connexions

## Authentification forte

Le protocole d'authentification utilisé dans rsync est un système de challenge réponse MD4 128bits. C'est une protection faible. Il est donc recommandé de lancer rsync dans ssh.