

Synchroniser des répertoires à l'aide de Lsyncd sous Linux

Table of Contents

- [Synchroniser des répertoires à l'aide de Lsyncd sous Linux](#)
 - [Installation et configuration de Lsyncd](#)
 - [Installation de Lsyncd](#)
 - [Configuration Lsyncd](#)
 - [Utilisation de Lsyncd](#)
 - [Synchroniser des répertoires locaux](#)
 - [Synchroniser des répertoires distants](#)

Cet article explique comment installer **Lsyncd** (Live Syncing Mirror Daemon) sur les systèmes Linux et comment synchroniser les répertoires distants et locaux sous Linux. **Lsyncd** est une solution de miroir en direct légère qui est relativement facile à installer sans entraver les performances du système de fichiers local existant.

Il est très utile de garder une trace de toute modification de données et de synchroniser ces données entre les répertoires qui sont mis à jour avec de nouveaux contenus. Par défaut, il s'agit uniquement de **rsync**.

Tous les fichiers de configuration personnalisés sont écrits en langage Lua, ce qui permet d'obtenir une configuration puissante, flexible et simple.

Installation et configuration de Lsyncd

Installation de Lsyncd



Sur un système CentOS 7.5, il faut activer le référentiel EPEL: `#yum install epel-release`

```
apt install lsyncd
```

On peut vérifier la version installée en exécutant cette commande :

```
lsyncd -version  
Version : 2.2.2
```

Configuration Lsyncd

Le fichier de configuration **Lsyncd** est créé automatiquement dans `/etc/lsyncd.conf`. Par défaut, son contenu se présente comme ci-dessous :

```
cat /etc/lsyncd.conf
----
-- User configuration file for lsyncd.
--
-- Simple example for default rsync, but executing moves through on the
target.
--
-- For more examples, see /usr/share/doc/lsyncd*/examples/
```

Il faut modifier ce fichier de configuration selon l'objectif. Comme mentionné dans ce fichier de configuration, tous les exemples de scripts sont disponibles à l'emplacement suivant : `/usr/share/doc/lsyncd-2.2.2/examples/`. Parmi ces fichiers, on examinera plus en détail l'utilisation des fichiers:

- **lrsync.lua**: Exemple de configuration pour la synchronisation locale

```
# cat /usr/share/doc/lsyncd-2.2.2/examples/lrsync.lua
----
-- User configuration file for lsyncd.
--
-- Simple example for default rsync.
--
settings {
statusFile = "/tmp/lsyncd.stat",
statusInterval = 1,
}

sync{
default.rsync,
source="src",
target="trg",
}
```

- **lrsyncssh.lua**: Exemple de configuration pour la synchronisation à distance

```
# cat /usr/share/doc/lsyncd-2.2.2/examples/lrsyncssh.lua
----
-- User configuration file for lsyncd.
--
-- Simple example for default rsync, but executing moves through on the
target.
--
sync{default.rsyncssh, source="src", host="localhost", targetdir="dst/"}
```

Utilisation de Lsyncd

Pour conserver ce processus de synchronisation au lieu d'exécuter une tâche cron toutes les X minutes, **Lsyncd** utilise des hooks de noyau Linux pour recevoir des notifications lorsqu'un fichier dans un répertoire a été modifié. Par défaut, il met en file d'attente toutes les commandes de synchronisation en 20 secondes. On peut modifier cet intervalle de temps au besoin avec l'option **-delay** dans la commande **sync**:

```
sync {  
  default.rsynccssh,  
  source = "SRC",  
  target = "DEST",  
  delete = "running", -- empêche la suppression de fichiers au démarrage  
  (c'est-à-dire lorsqu'un serveur revient en ligne, ne supprime pas les  
  fichiers nouveaux sur la sauvegarde)  
  delay = 5, -- exécuté toutes les 5 secondes au lieu de 20 par défaut  
}
```



Lors de l'utilisation de **default.rsynccssh** dans la commande **sync**, il est toujours recommandé d'utiliser l'option **-delete** pour éviter de manquer des fichiers dans les dossiers de destination. Après avoir créé ou modifié le fichier de configuration **Lsyncd**, il faut redémarrer le processus **Lsyncd**.

Étant donné que **Lsyncd** utilise l'outil **rsync** pour copier, déplacer et supprimer des fichiers de la source à la destination. On peut utiliser les commutateurs **rsync** pour éviter de créer des doublons inutiles sur la destination et faciliter ce processus. Certaines des options importantes de **rsync** sont expliquées ci-dessous :

- **-delete**: cette option garantit que tous les fichiers du répertoire distant qui ne se trouvent pas dans le répertoire source sont supprimés.
- **-times**: cette option est assez importante lorsqu'on veut exécuter ce script encore et encore, car elle gardera les temps entre les deux fichiers synchronisés.
- **-force**: Cette option permet de supprimer un répertoire non vide afin d'être remplacé par un répertoire vide.
- **-links**: Cette option est utilisée pour copier des liens symboliques en tant que liens symboliques.
- **-progress2**: il affiche la progression globale de l'ensemble du transfert, pas seulement du fichier unique en cours de copie.
- **-dry-run**: cette option effectue un essai sans effectuer de suppression ou de transfert, mais en indiquant ce qu'elle fera. Il est fortement recommandé d'utiliser cette option la première fois.
- **-owner**: garantit que l'utilisateur propriétaire du fichier est conservé (pas le niveau d'autorisation du propriétaire).
- **-group**: garantit que l'utilisateur de groupe du fichier est conservé (pas le niveau d'autorisation pour le groupe).
- **-perms**: préserve les autorisations.

- **-sparse**: garantit que les fichiers d'images clairsemés sont transférés efficacement.



rsync peut être utilisé pour synchroniser des périphériques de blocs en appliquant le patch **copy-devices.diff** qui introduit l'option **-copy-devices**.

Synchroniser des répertoires locaux



L'utilisation de fichiers journaux est facultative, mais il est préférable de créer des fichiers journaux et les fichiers d'état pour toutes les tâches, pour suivre le processus:

```
mkdir /var/log/lsyncd
touch /var/log/lsyncd/lsyncd.{log,status}
```

Pour synchroniser des dossiers locaux à l'aide de **Lsyncd** (par exemple le dossier source SRC_DIR et le dossier cible à savoir DEST_DIR), il faut modifier le fichier de configuration **Lsyncd**. Comme on l'a vu précédemment, le fichier de configuration **lsyncd** par défaut est `/etc/lsyncd.conf`, il faut mettre à jour ce fichier avec les répertoires source et cible et les fichiers journaux:

```
----
-- User configuration file for lsyncd.
--
-- Simple example for default rsync, but executing moves through on the
target.
--
-- For more examples, see /usr/share/doc/lsyncd*/examples/
--
settings {
logfile = "/var/log/lsyncd/lsyncd.log",
statusFile = "/var/log/lsyncd/lsyncd.status"
}

sync {
default.rsync,
source = "/root/SRC_DIR",
target = "/root/DEST_DIR",
}
```

Une fois cela fait, redémarrer et activer le service **Lsyncd**.

```
# systemctl enable lsyncd
lsyncd.service is not a native service, redirecting to systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable lsyncd
```

```
# systemctl start lsyncd
```

Consulter les fichiers journaux et d'état pour vérifier l'état de la réplication afin de confirmer si elle est terminée ou non.

```
# tail -10 /var/log/lsyncd/lsyncd.log
Thu Aug 2 14:03:16 2018 Normal: --- Startup ---
Thu Aug 2 14:03:16 2018 Normal: recursive startup rsync: /root/SRC_DIR/ ->
/root/DEST_DIR/
Thu Aug 2 14:03:16 2018 Normal: Startup of /root/SRC_DIR/ -> /root/DEST_DIR/
finished.

# more /var/log/lsyncd/lsyncd.status
Lsyncd status report at Thu Aug 2 14:03:27 2018

Sync1 source=/root/SRC_DIR/
There are 0 delays
Excluding:
nothing.

Inotify watching 1 directories
1: /root/SRC_DIR/
```

Afin de synchroniser plusieurs dossiers vers un ou plusieurs répertoires cibles, il faut mettre à jour le fichier de configuration avec plusieurs d'instructions **sync** avec les répertoires source et cible requis.



```
sync{ default.rsync, source='source1', target='target1' }
sync{ default.rsync, source=' 'source2', target='target2' }
```

De même, on peut utiliser ces instructions de commande de synchronisation “default-rsync” plusieurs fois lorsqu'on souhaite synchroniser les mêmes répertoires source vers plusieurs cibles ou plusieurs répertoires source vers le même répertoire cible.

Synchroniser des répertoires distants

Afin de synchroniser des répertoires distants, il faut configurer une connexion SSH sans mot de passe. Cela aidera **Lsyncd** à répliquer automatiquement le contenu du répertoire local dans un répertoire distant sans intervention de l'utilisateur. Puisqu'on effectue la synchronisation sur deux serveurs. Il faut définir un serveur source et un serveur cible pour expliquer plus clairement ce processus:

- **Étape 1:** Création de clés SSH sur le serveur source pour une connexion sans mot de passe: `ssh-keygen -t rsa`
- **Étape 2:** Copier les clés publiques sur le serveur cible: `ssh-copy-id root@45.33.121.82` (à partir de maintenant on peut se connecter au serveur de destination distant à partir de la source sans aucun mot de passe).

- **Étape 3:** Création d'un répertoire cible pour la synchronisation à distance dans le serveur cible : `mkdir Remote_Dir`
- **Étape 4:** Modifier le fichier de configuration **Lsyncd** pour activer la synchronisation à distance

```
# cat /etc/lsyncd.conf
----
-- User configuration file for lsyncd.
--
-- Simple example for default rsync, but executing moves through on the
target.
--
settings {
logfile = "/var/log/lsyncd/lsyncd.log",
statusFile = "/var/log/lsyncd/lsyncd.status"
}
sync{default.rsyncssh, source="/root/SRC_DIR", host="45.33.121.82",
targetdir="/root/Remote_Dir"}
```

- **Étape 5:** Redémarrer le service Lsyncd sur la source

Afin de synchroniser vers plusieurs serveurs distants, il faut mettre à jour le fichier de configuration avec plusieurs d'instructions **sync**, mais il faut aussi activer la connexion SSH sans mot de passe en répétant les étape 1 à 3 sur chacun des serveurs distants:



```
sync{default.rsyncssh, source="source1", host="host1",
targetdir="target1"}
sync{default.rsyncssh, source="source2", host="host2",
targetdir="target2"}
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=outils:lsyncd-rsync>

Last update: **2025/02/19 10:59**

