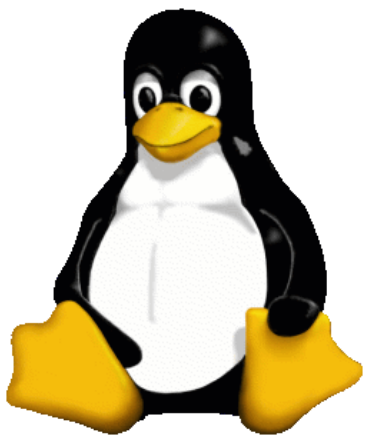


<https://vader-fr.fr/spip.php?article54>



Script sh sauvegarde incrémentale déportée tar.gz via ssh

- Linux & Logiciels Libres -



Publication date: vendredi 16 mars 2012

Copyright © Vader[FR] : ce n'est pas un blog, c'est un Sith - Tous droits
réservés

Admettons que vous ayez une machine A dont vous voulez sauvegarder des données, via une archive tar compressée gz(ip).

lesquelles sauvegardes seraient incrémentales, afin de limiter l'utilisation des disques.
cela se fait avec un fichier d'index : tar -g fichier_index -czf fichier.tar.gz repertoire_a_sauver

il suffit alors, lorsque l'on indique le nom du fichier index a utiliser, de préciser une variable de mois ou de semaine, afin de créer un nouveau fichier index tous les mois/toutes les semaines.... et donc, lorsque le fichier index n'existe pas encore, la sauvegarde est totale.

cependant, vous risquez sur cette machine A de ne pas avoir la place de stocker les sauvegardes.
Il faudrait donc envoyer ce fichier sur un serveur de sauvegarde B.
Et directement. pas faire le tar.gz sur la machine locale avant de le copier.

donc, envoyer du flux (compressé si possible) sur un serveur B.

quelles options ?

- partage SAMBA : pas très sécurisé, en plus monter une ressource SAMBA via un script est assez casse-pieds.
- ftp: pareil, pas très simple à mettre en oeuvre
- SSH :D wiwi, c'est possible, et avec une authentification par clé c'est simple à mettre en oeuvre. Et en plus c'est sécurisé.

il suffit pour cela de copier dans /.ssh/authorized_keys du serveur le contenu de /.ssh/id_rsa.pub du client.

les répertoires utilisateurs "~" sont à adapter en fonction de l'utilisateur choisi pour faire la sauvegarde.

on envoie donc le tar.gz sur une connexion ssh, on récupère le flux avec split afin de créer un fichier (faire tar et récupérer avec gzip ne marche pas)

j'ai testé les diverses commandes susceptibles de récupérer du flux pour en faire un fichier, split semble être le meilleur moyen.

split va "scinder" le flux en.....un seul, en fait, fichier. on renommera ensuite le fichier .tar.gz.00 (limite 2 To sur ext >=3) en .tar.gz pour pouvoir l'exploiter plus tard.

et au cas où, on envoie aussi le fichier index (créé sur A) sur le répertoire de B, afin de pouvoir toujours exploiter les sauvegardes incrémentales situées sur B en cas de défaillance de la machine A

tant qu'on y est, la sauvegarde (tar.gz) prend la date du jour, plus facile pour s'y retrouver ensuite

```
#!/bin/sh
# =====
#
```

Script sh sauvegarde incrémentale déportée tar.gz via ssh

```
# Script sauvegardes locales + déportées sur serveur externe (flux tar.gz incrémental envoyé en ssh)
#
# Lancé par une tâche CRON
#
# =====> A automatiser par script générique avec BDD hote/profil et serveur/repertoire de sauvegarde
# ?
# =====

# =====
# Script sauvegarde de répertoire 1
#
# sauvegarde totale chaque début de mois, incrémentale tous les jours de la semaine
#
# et on garde 2 mois de sauvegarde
# =====
# envoi du flux sur ssh, récupération par split sur 1 fichier tar.gz.0, et renommage de ce fichier de split
tar.gz.0 en tar.gz

# /\ tester la taille du ou des répertoires à sauvegarder par rapport à la taille max de fichier sur le
serveur backup !

# /\ tester la présence du serveur et sa connectivité ssh (ping puis création/récupération fichier témoin
?)

# Traitements de début de mois : effacer les anciennes sauvegardes de plus de 2 mois.
```

Nouveau code, plus simple puisque `date` permet de calculer une date passée ou future, et de la mettre en forme. La sauvegarde est de forme "nomjour-année-mois-jour" et l'index de forme "mois-année". Le nom du jour n'est pas techniquement utile (ni pratique) dans le nom de la sauvegarde, mais c'est parfois plus facile pour les utilisateurs de retrouver un jour passé en lisant son nom.

```
if [ $(date +%d) = "01" ] ; then
    lastsav=$(date --date '3 months ago' +%Y-%m);
    lastindex=$(date --date '3 months ago' +%m-%Y);
    ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f fichier_sauvegarde-*${lastsav}*.tar.gz"
    ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f index-${lastindex}*"
fi
```

Ancien code pour effacer les sauvegardes de mois-2

```
if [ $(date +%d) = "01" ] ; then
    # le 1er janvier => supprimer ceux de l'an passé (novembre)
```

Script sh sauvegarde incrémentale déportée tar.gz via ssh

```
if [ $(date +%m) = "01" ] ; then
oy=$(expr `date +%Y` - 1); # l'année à effacer
echo "Nous sommes le 1er janvier : on efface les sauvegardes déportées de novembre de l'année passée
($oy)";
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f fichier_sauvegarde-*$oy-11*.tar.gz"
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f index-11-$oy"
fi
# le 1er février => supprimer ceux de l'an passé (décembre)
if [ $(date +%m) = "02" ] ; then
oy=$(expr `date +%Y` - 1); # l'année à effacer
echo "Nous sommes le 1er février : on efface les sauvegardes déportées de décembre de l'année passée
($oy)";
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f fichier_sauvegarde-*$oy-12*.tar.gz"
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f index-12-$oy"
fi
# en temps normal, chaque 1er du mois de mars à décembre (donc sauf janvier et février) => supprimer ceux
de mois - 2
if [ $(date +%m) != "01" ] && [ $(date +%m) != "02" ] ; then
om=$(expr `date +%m` - 2); # le mois a effacer
```

Là, `expr` renvoie **un** chiffre, de 0 à 10, alors que les mois vont de 01 à 12, sur deux chiffres.
Donc il faut corriger la variable `om`, sinon le `rm` ne correspondra pas aux fichiers de sauvegardes.

```
if [ $om -lt 10 ] ; then
om="0$om";
fi
echo "Nous sommes le 1er $(date +%B %Y) : on efface les sauvegardes déportées du mois de $(date +%B %Y'
-d $om/01/`date +%Y`)" ;
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f fichier_sauvegarde-*$(date +%Y)-$om*.tar.gz"
ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; rm -f index-$om-$(date +%Y)"
fi
```

```
fi # fin traitement début de mois
```

```
# et création d'une nouvelle sauvegarde
# copie simple : prend trop de place
# gzip : concatène le répertoire dans un seul fichier ==> ne peut être déconcaténé !

# pas de sauvegarde le week end (jours 6 et 7 de la semaine)
if [ $(date +%u) != 6 ] && [ $(date +%u) != 7 ]; then
# sauvegarde incrémentale si le fichier existe. il est effacé tout les premiers du mois
echo "Debut sauvegarde le $(date +%A %d/%m/%Y) à $(date +%H:%M)";
# sauf répertoires répertoire1/l_A et répertoire1/l_B
# le comportement de la commande $(date +format) à l'intérieur des parenthèses du SSH étant imprévisible,
```

Script sh sauvegarde incrémentale déportée tar.gz via ssh

```
# il vaut mieux stocker le résultat dans une variable)
# ici, la date du jour : le nom complet du jour (%A) et la date complète année-mois-jour (%F), séparés par
un tiret
datesav=$(date +%A-%F);
# envoi du flux incrémental, récupération par split, renommage du fichier splité (pas réellement splité en
fait)
# le -n rend l'archive "seekable", il n'y aura donc en théorie pas besoin de parcourir toute l'archive pour
atteindre un fichier
tar -n -g /chemin/index/local/index-$(date +%m-%Y) -czf - /repertoire1 --exclude="repertoire1/1_A"
--exclude="repertoire1/1_B" | (ssh user@serveur_externe "cd /chemin/fichier/sauvegarde; split -b 2T -d -
fichier_sauvegarde-$(datesav).tar.gz.; mv fichier_sauvegarde-$(datesav).tar.gz.00
fichier_sauvegarde-$(datesav).tar.gz")
# et envoi de l'index (en cas de défaillance de la machine A, il faudra l'index pour pouvoir exploiter les
sauvegardes sur B)
scp /chemin/index/local/index-$(date +%m-%Y) user@serveur_externe:/chemin/fichier/sauvegarde/
echo "Fin sauvegarde le $(date +%A %d/%m/%Y) à $(date +%H:%M)";
fi # fin si pas week end
```

bien sûr, on pourrait "jouer" davantage en mettant un script "générique" qui récupérerait (wget ?) du serveur B des paramètres (quoi sauvegarder, quand, , comment "profil" et où), et après la sauvegarde, demanderait (wget, encore, ou ssh) au serveur B de cataloguer la sauvegarde.