

Plan de sauvegarde

PROJET SYSTÈMES

Charles Abélard | Adèle Brassard | Anthime Doittée | Rania Zaafane

Risques

Les risques pour le site web de l'entreprise sont les suivants :

Problèmes	Prévention	Action si le problème survient
Erreur humaine	Sensibiliser les équipes	
Usure ou panne	Acheter le matériel en avance	Le remplacer
Virus, panne logicielle	Faire des sauvegardes régulières	Restaurer à la dernière sauvegarde
Vol de données	Lister les droits d'utilisateur	Agir en conséquence (licencier si la personne fait partie de l'entreprise)

Les données à sauvegarder

Pour répondre au risque de virus ou de panne, nous réalisons des sauvegardes régulières.

Nous avons décidé d'automatiser les sauvegardes incrémentales toutes les deux heures et celles complètes tous les mois. Ainsi, en cas de problèmes, le chiffre d'affaires de l'entreprise n'en pâtira que peu.

Voici la commande qui automatise cela :

```
GNU nano 2.9.3 /tmp/crontab.ZqGskQ/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 0 * * 0 /home/gonfreecs/Bureau/backup.size.sh
0 */2 * * * /home/gonfreecs/Bureau/incremental.size.sh
```

Nous avons décidé de conserver ces sauvegardes durant 6 mois, en cas de problème qui adviendrait sur la durée.

Les sauvegardes contiendront les fichiers de configuration de HTTP et de www. et seront stockées sur une clef USB.

Les méthodes de sauvegarde

Pour réaliser ces sauvegardes, nous allons utiliser les sauvegardes complètes et incrémentales. Dans un premier temps, la sauvegarde complète va permettre de sauvegarder tous les fichiers et dans un second temps, l'incrémentale va sauvegarder uniquement les fichiers modifiés. Nous avons choisi ces sauvegardes pour éviter toute perte et pour optimiser l'espace de stockage.

Enfin, en cas de problème lors de la sauvegarde, un email sera automatiquement envoyé aux administrateurs afin qu'ils puissent réagir dans les plus brefs délais.

Le script

Voici les scripts qui réaliseront ces tâches :

La sauvegarde incrémentale

Dans un premier temps on va déclarer la variable date afin de ne plus réécrire la date

```
date=`date '+%Y-%m-%d%H%M'`
```

Puis on va déclarer le chemin destination. (Ici une clé USB).

```
BACKUP="media/gonfreecs/CLEF/Backup_inc/$date"
```

Ici la variable, est le chemin du dossier ressources.

```
DOSSIER_Res="var/www"
```

Rsync est une commande qui permet de synchroniser les dossiers, elle peut voir les modifications et ne sauvegarder que les modifications. Par défaut, elle ne copie que des fichiers mais comme nous avons rajouté le drapeau -a cela permet de le faire aussi sur les dossiers. L'argument -a correspond à archive/recursive. Elle rassemble plusieurs arguments :

- -l permet de recréer les liens symboliques.
- -p permet de recopier les permissions.
- -t permet de transférer les dates de modifications.
- -g et -o sont utilisés pour copier les groupes et les owners.

Tous ces drapeaux sont rassemblés dans le drapeau -a. le drapeau -v correspond à verbose. J'ai ajouté les arguments --no-o et --no-g pour ne pas avoir de problème de permission donc de chown. Et pour finir avec les drapeaux, --delete est un drapeau destructif, il va supprimer des dossiers/fichiers à distance. Ce qui veut dire que sans ce drapeau, si nous supprimons un

dossier dans le dossier ressource, il ne supprimera pas dans le dossier destination alors qu'avec ce drapeau on le supprimera. Dans un premier temps, c'est le chemin du dossier ressource et le second, c'est le chemin du dossier destination.

```
rsync -a --no-o --no-g --delete /$DOSSIER_Res /$BACKUP
```

Cette commande va chercher le dossier destination. Le -mtime +x est la durée de vie du fichier/dossier rm et la suppression l'ajout du drapeau -fr permet la suppression des fichiers et dossiers non vides.

```
find /media/projet/CLEF/Backup_inc -mtime +1 -exec rm -fr {} \;
```

Cette variable va permettre d'extraire la disponibilité de la machine. Dans un premier temps, on extrait uniquement la deuxième ligne avec cette commande |sed -n '2p', puis il nous faudra prendre le caractère 4 après un espace, donc avec la commande | tr -s ' ' on se dit que même s'il y a plusieurs espaces, on le compte comme s'il n'y en avait qu'un. Enfin, pour couper le caractère 4 juste avant l'espace on va utiliser cette commande | cut -d ' ' -f 4`.

```
espace=`df / | sed -n '2p' | tr -s ' ' | cut -d ' ' -f 4`
```

On ajoute une condition si la machine à moins de 1GO, alors elle rentre dans la boucle.

```
if [ $espace -lt 1000000 ]
```

Quand la machine aura moins de 1GO, elle va afficher dans créer un fichier log avec la date, et le script va nous envoyer un mail d'erreur.

```
then
    #echo "ERREUR de sauvegarde"
    echo "ERREUR de sauvegarde" >> "/home/gonfreecs/Bureau/log_$date.csv"
    echo "ERREUR de sauvegarde" | mail -s "Alerte_sauvegarde_$date" admin@carnoflux.com
fi
```

Cette variable correspond à la recherche de la dernière sauvegarde dans le dossier destination.

```
NBRE_FILES=find $BACKUP -type f -mtime -1 2>/dev/null | wc -l
```

Cette variable correspond à la recherche de la dernière sauvegarde dans le dossier ressources.

```
NBRE_FILES2=find $DOSSIER_Res -type f -mtime -1 2>/dev/null | wc -l
```

Ici, on a une condition if qui compare le dossier ressources et destination donc si tous les fichiers/dossiers sont dans les deux dossiers alors il n'y a pas de problèmes sinon les erreurs seront stockées dans un fichier log et envoyées par mail.

```
if [ ! -e $BACKUP/$DOSSIER_Res ] && [ "$NBRE_FILES" == 0 ] || [ "$NBRE_FILES2" == 0 ]
then
    echo "Il n'y a pas de problème. \n" >> "/home/gonfreecs/Bureau/log_$date.csv"
else
    echo "Erreur de sauvegarde incrémentale du $date" | mail -s "Erreur de sauvegarde" root
    echo "ERREUR de sauvegarde incrémentale" >> "/home/gonfreecs/Bureau/log_$date.csv"
fi
```

La sauvegarde complète

```
#!/bin/sh
date=`date '+%Y-%m-%d%H%M'`
```

Dans un premier temps on va déclarer la variable date afin de ne plus réécrire la date

```
tar -cvzf /media/gonfreecs/CLEF/Backup_com/backup.tar.gz "/var/www"
```

Ici on compresse la sauvegarde et on l'envoie dans le dossier destination.

```
find /media/gonfreecs/CLEF/Backup_com -mtime +1 -exec rm -fr {} \;
```

Je demande à mon script de supprimer les dossiers/fichiers datant de plus d'un jour (afin de prouver que cela fonctionne la commande, voilà pourquoi je n'ai pas mis 6mois=183j).

```
if [ -e "/media/gonfreecs/CLEF/Backup_com/backup.tar.gz" ]
```

Je rajoute une condition, si mon dossier/fichier existe alors pas de problème.

Intervention en cas de panne

Si un problème survenait, pour restaurer la sauvegarde, veuillez rentrer la commande suivante dans un terminal du serveur HTTP :

```
sudo foremost -w /var/www /media/gonfreecs/CLEF/Backup_(inc ou com)
```