



Rapport TP4 : Configuration complète d'un serveur

DRISSI Mohamed Reda
reda-mohamed@isty.uvsq.fr

11 décembre 2018

Table des matières

I	Stratégie De Sauvegarde	2
	I.1 Exercice 1	2
	I.2 Exercice 2	2
	I.3 Exercice 3	2
	I.4 Exercice 4	2
	I.5 Exercice 5	3
II	Espace de stockage	3
	II.1 Exercice 6	3
	II.2 Exercice 7	3
	II.3 Exercice 8	4
III	Sauvegarde	4
	III.1 Exercice 9	4
	III.2 Exercice 10	5
	III.3 Exercice 11	5
	III.4 Exercice 12	6
	III.5 Exercice 13	6
	III.6 Exercice 14	6
	III.7 Exercice 15	6
	III.8 Exercice 16	6
	III.9 Exercice 17	7
IV	Restauration	7
	IV.1 Exercice 18	7
	IV.2 Exercice 19	7
	IV.3 Exercice 20	8

I Stratégie De Sauvegarde

I.1 Exercice 1

La sauvegarde incrémentale permet d'appliquer les changements qui ont été effectués depuis la dernière sauvegarde (uniquement les nouveaux fichiers, et les parties modifiés des fichiers déjà présents). Puisque cette sauvegarde copie uniquement les changements, elle est considérablement plus rapide que la sauvegarde complète. Le problème avec les sauvegardes incrémentales est que la restauration est faible, puisqu'il faut restorer la dernière sauvegarde complète, puis chaque sauvegarde incrémentale depuis, donc si nous avons une sauvegarde complète d'il y a un mois puis des sauvegardes incrémentales quotidiennes, il faudra restaurer la sauvegarde complète puis une trentaine de sauvegardes incrémentales, ce qui aussi bien évidemment implique des risques de données dans le cas où une ou plusieurs sauvegardes sont corrompues.

I.2 Exercice 2

- Une sauvegarde complète chaque mois qui sera compressé dans un `.tar.bz2`, qui commence à 19h
- Une sauvegarde différentielle chaque semaine qui commence à 19h chaque vendredi
- Une sauvegarde incrémentale toutes les 4 heures

Nous choisissons 19h parce que c'est le temps où la plupart si pas tout le monde a quitté son bureau.

I.3 Exercice 3

- Base de données MariaDB : Nous allons utiliser `mysqldump` au lieu d'une simple copie pour éviter la corruption de transactions en cours. Notre base de données est presque vide, donc elle ne doit pas dépasser les 1Mb.
- Le répertoire `/home` est actuellement presque vide aussi, mais nous supposons un total de 3Gb
- Les données de l'application WordPress doivent être aux alentours de 60Mb au total :
 - `/usr/share/wordpress/` 56Mb
 - `/etc/wordpress` <1Mb
 - `/var/lib/wordpress` <1Mb
- Les données de l'annuaire LDAP contenus dans `/var/lib/ldap` qui eux aussi sont inférieures à 1Mb

I.4 Exercice 4

- Bandes magnétiques :
 - **Avantages** : Grosses capacités, très grande durée de vie.

- **Inconvénients** : Très lent, vulnérables aux aimans, même les plus faibles (ex hauts parleurs)
- Disques durs magnétiques SATA (HDD) :
 - **Avantages** : Disponibles partout, pour de très faibles coûts, relativement fiables, compatibles partout.
 - **Inconvénients** : Usure à cause des parties amovibles, vitesse très lente.
 - Stockages portables (hdd externes, clés usb)
 - **Avantages** : facilité de transport et d'usage
 - **Inconvénients** : capacité limitée, peu fiable, vitesse la plus lente après les bandes magnétiques.
- Disques SSD SATA :
 - **Avantages** : Beaucoup plus rapides que les HDDs, aucune partie amovible, température toujours basse.
 - **Inconvénients** : Prix coûteux, durée de vie relative aux nombres de lectures/ écritures
- Disques SSD NVMe PCIe/M.2 :
 - **Avantages** : Beaucoup plus rapide que tout le reste
 - **Inconvénients** : Beaucoup plus chers que tout le reste

I.5 Exercice 5

Selon les supports, certains sont plus vulnérables à quelques conditions qu'à d'autres, et la plupart sont vulnérables à ce qui nuit à l'informatique de manière générale; telle la température élevée (ou trop basses), l'humidité, la poussière ou bien même les ondes électro-magnétiques.

II Espace de stockage

II.1 Exercice 6

Nous avons déjà ajouter plusieurs disques durs sur virtualbox, ceci est trivial. Nous choisissons un disque de 4Gb, qui dépasse largement notre taille de sauvegarde.

II.2 Exercice 7

Comme dans le TD2 nous allons utiliser parted ou fdisk ou cfdisk pour la création d'une partition. Puis nous exécutons les commandes nécessaires au formatage et configuration de fstab :

```
$ mkfs.ext4 /dev/sdb1
$ mkdir /backups
$ echo " /dev/sdb1 /backups ext4 defaults 0 2" >> /etc/fstab
```

II.3 Exercice 8

Garder un seul support de stockage lie nos données à la fiabilité de ce support. Utiliser une méthode RAID pour une réplication de données ou un NAS, ou n'importe quelle autre méthode sera mieux adapté.

III Sauvegarde

III.1 Exercice 9

Nous utilisons `tree` pour montrer l'arborescence

```
$ tree /backups
/backups
  daily
  ãã daily.0 <- dossier
  ãã daily.1 <- dossier
  increments
  monthly
  ãã monthly_2018-12-11.tar.bz2 <- archive
  weekly
    weekly.0 <- dossier
```

III.2 Exercice 10

Nous évitons d'utiliser la variable `$PATH` parce que le script sera utilisé avec cron.

```
#!/bin/sh
unset PATH
DAY=$((date +%u) - 1))
WEEK=$((date +%w) - 1 ))
MONTH=$((date +%m) - 1 ))
YEAR=$(date +%Y)
RSYNC="/usr/bin/rsync"
MKDIR="/bin/mkdir"
TAR="/bin/tar"
RM="/bin/rm"
SOURCE="/backups/include.txt" #currently contains /home
#can contain cache and other non important stuff
EXCLUDES="/backups/exclude.txt"
INCREMENTS="/backups/increments/$(date +%d-%m-%Y_%H:%M:%S)"
DEST="/backups/daily/daily.$DAY"
WEEKLY="/backups/weekly/weekly.$WEEK"
OPTS="-arf --ignore-errors --delete --files-from=$SOURCE\
--exclude-from=$EXCLUDES"
DOPTS="--backup --backup-dir=$INCREMENTS --log-file=\
/backups/log.daily.$(date +%F).log"
WOPTS="$WEEKLY --log-file=/backups/log.weekly.$(date +%F).log"
# Delete backups not needed
if [ $DAY -eq 0 ]; then
    # The following will only delete the folder if it's already full
    $MKDIR -p /backups/weekly/weekly."$WEEK" && $RM -r\
/backups/weekly/weekly."$WEEK"/*
    $RSYNC $OPTS $WOPTS $DEST
    if [ $WEEK -eq 0 ]; then
        MONTHLY = /backups/monthly/year."$YEAR"/monthly."$MONTH"
        $MKDIR -p
        $TAR -cpjf "$MONTHLY"/monthly_$(date +%F).tar.bz2 /home
    else
        # The following will only delete the folder if it's already full
        $MKDIR /backups/daily/daily."$DAY" && $RM -r\
/backups/daily/daily."$DAY"/*
        $RSYNC $OPTS $DOPTS $DEST
```

Nous pouvons aussi utiliser des utils comme `rsnapshot` ou `duplicity` pour éviter toutes ces configurations.

III.3 Exercice 11

Nous aurons à répartir nos archives en de petits fichiers, la commande split sera parfaite pour ceci.

III.4 Exercice 12

```
mysqldump --all-databases --add-drop-database --add-drop-table \  
--add-drop-trigger > /home/$USER/msqldump_$(date +%d-%m-%Y_%H:%M:%S).bak
```

Ensuite elle sera sauvegardée avec le reste du home

III.5 Exercice 13

Si nous copions les fichiers d'une base de données alors qu'une transaction est en cours, nous risquons de corrompre les données de cette DB. Plusieurs solutions existent :

- L'usage d'une sauvegarde logique avec un outil comme mysqldump.
- Verrouiller la base de données en lecture seule durant la sauvegarde.
- Arrêter la base de données lors de la sauvegarde.
- L'usage de la fonctionnalité snapshot si le système de fichiers le permet (zfs ou btrfs)

III.6 Exercice 14

La taille de la sauvegarde va causer une opération de sauvegarde très lente. Nous avons donc encore plus de risque de corruption de cette base de données. Les solutions que nous pouvons adopter :

- Usage de quelques outils plus rapides.
- Verrouiller la base de données en lecture seule.
- Usage de la fonctionnalité de snapshot, si possible.

III.7 Exercice 15

Ajouter les fichiers de ldap (/var/lib/ldap) dans le fichier "/backups/include.txt"

III.8 Exercice 16

Nous pouvons générer ces checksums avec la commande :

```
find /path/to/backup -type f -exec md5sum {} + > /path/to/backup/checksum.md5
```

Nous pouvons les vérifier avec

```
md5sum -c /path/to/backup/checksum.md5
```

III.9 Exercice 17

```
$ crontab -e  
* 0 * * * /path/to/script
```

IV Restauration

IV.1 Exercice 18

Génération d'un backup complet :

```
SOURCE="/home /var/lib/ldap \  
        /usr/share/wordpress  
        /etc/wordpress/  
        /var/lib/wordpress/"  
DAY= $(( $(date +%u) - 1 ))  
mkdir /backups/daily/daily.$DAY  
mysqldump --all-databases --add-drop-database --add-drop-table \  
--add-drop-trigger >/home/$USER/msqldump_$(date +%d-%m-%Y_%H:%M:%S).ba  
rsync -arf --ignore-errors --delete $SOURCE /backups/daily/daily.$DAY
```

Génération d'un backup incrémental :

```
$ bash /root/backing_up.sh
```

IV.2 Exercice 19

Après avoir transféré les fichiers, nous commençons tout d'abord par bien vérifier nos hashes md5 avec la commande :

```
$ md5sum -c /backups/daily/daily.1/checksum.md5
```

Nous restaurons le backup :


```
DAY= $(($(date +%u) - 1 ))  
rsync -arf --ignore-errors --delete /backups/daily/daily.$DAY /
```

Puis les données de notre base de données :

```
mysql > source /home/adminsys/msqldump_11-12-2018_19:48:35.bak
```

IV.3 Exercice 20

Pour se le faire nous allons faire la même chose que lorsqu'on restaure une sauvegarde normale, le seul changement est de copier le répertoire htop-dev uniquement au lieu du backup complet.

```
rsync -arf --ignore-errors --delete \  
/backups/path/to/needed/backup/home/raj/htop-dev \  
/home/raj
```