

# INSTALLATION DE BACKUPPC

**BackupPC** est un logiciel très intéressant, très facile à installer, et à gérer et qui permet bien sûr de faire de bonnes sauvegardes de nos PCs. Aujourd'hui, le backup de nos machines notamment serveur de données est aussi (sinon même plus) important que la « sécurité » de ces machines. (Par exemple, si votre disque dur plante, vous avez beau avoir le meilleur firewall du monde, le meilleur IDS de la planète, vous êtes \_\_\_\_). Et BackupPC est un logiciel qui joue bien se rôle ; avec son interface WEB, et ces nombreuses méthodes de sauvegardes, il arrive à me satisfaire complètement. (Il faut reconnaître aussi que je suis pas un gars bien compliqué.)

Démarrons donc notre installation :

## 1- Installation :

Installation du serveur de base Debian Etch ou tout ce que vous voulez.

Paquets à rajouter :

- **apticron** : pour les annonces de mises à jour
- **beep** : pour pouvoir faire du bruit en cas de problème et alerter les gens
- **git** : pour la gestion de changement de configurations
- **logcheck** : checker les logs à la recherche de choses bizarres
- **ntp** : pour synchroniser l'heure
- **partimage** : copie de partition, me permet de faire du backup de backup
- **rsync** et **libfile-rsyncperl** +++: requis par notre méthode de sauvegarde des Pcs (rsync)
- **snmpd** : Surtout pour monitorer la partition qui sert de bakup
- **sshd** +++: faire du SSH
- **sudo** : pour faire du sudoers
- **tcpdump** : faire de l'analyse de trafic quand c'est bizare
- **backuppc** +++ : Rôle de la machine.

Les paquets marqués +++ sont obligatoires pour cette installation.

Sous Debian, les install de tous ces trucs se font avec : `aptitude install <paquet>`

Sous Redhat et autres Fedora et CentOS, essayer avec : `yum install <paquet> // Pas testé`

Donc installation très très facile.

## 2- Config de ce sympathique BackupPC :

Toute la config se trouve dans le répertoire **/etc/backuppc/**. Il faut donc commencer par s'y positionner. `# cd /etc/backuppc/`

La config même sophistiquée n'est pas du tout compliquée, malgré la longueur du fichier principal de configuration. Et en général, les options par défaut sont tout à fait valides et marchent bien.

Donc, faisons un peu de customisation en ouvrant le fichier **/etc/backuppc/config.pl**

Changeons ces options :

```

$Conf{WakeupSchedule} = [1..24]; // Réveil toutes les heures de la journée
$Conf{MaxOldLogFiles} = 28; // L'âge max des logs
$Conf{DfMaxUsagePct} = 92; // à 92% de saturation de disque: ==> envoi de rapport, plus de
backup régulier
$Conf{BackupPCUser} = 'backuppc';
$Conf{FullPeriod} = 13.97; // Full backup toutes les 2 semaines (par PC bien sûr)
$Conf{IncrPeriod} = 0.97; // Incrémental backup tous les jours
$Conf{FullKeepCnt} = 6; // 6 backup complets, or 1 backup complet toutes les 2 semaines , donc
$Conf{FullAgeMax} = 93; // Age maxi des backup-complet = 93 jours, donc plus de 3 mois
$Conf{FullKeepCntMin} = 1; // Garder au minimum un backup-complet, même s'il est plus
vieux que 93 jours.
$Conf{IncrKeepCnt} = 15; // On garde 15 sauvegardes incrémentales, ce qui nous fait 15 jours
$Conf{IncrAgeMax} = 30; // 30 jours max pour l'âge de celles-ci
$Conf{IncrKeepCntMin} = 3; // On garde 3 sauvegardes incrémentales quelques soient leur âge.
$Conf{BlackoutPeriods} = [
    {
        hourBegin => 7.5,
        hourEnd => 18.5,
        weekDays => [1, 2, 3, 4, 5],
    },
]; // En gros, la période où il faut pas faire trop de backup si les machines ne répondent pas bien
aux pings. Le logiciel fait des pings périodiques et garde des statistiques. Il sait donc quand une
machine et/ou le réseau vers la machine est un peu saturé.
$Conf{XferMethod} = 'rsync'; // La méthode de sauvegarde par défaut (il y'a aussi tar, rsyncd,
smb)
$Conf{PingMaxMsec} = 999; // Délai max pour un ping, avant de considérer que le PC n'est pas
joignable et donc ne peut se faire backup.
$Conf{Language} = 'fr' // l'interface Web et les messages en français
$Conf{EMailUserDestDomain} = '@yerbynet.com' //

```

Voilà ! FIN de la config de BackupPC qu'il faut juste relancer :

```

# invoke-rc.d backuppc restart // Sous debian et compagnies
# service backuppc restart // Sous fedora et consorts

```

### 3- Supplément BackuPC (Interface Web) :

Inclure la config web de BackupPC dans notre Apache en rajoutant une ligne de Include dans la config de Apache :

```

Include /etc/backuppc/apache.conf

```

Ligne que j'ai rajoutée dans /etc/apache2/apache.conf

Puis, réactualiser apache :

```

# apache2ctl graceful // Sous debian
# apachectl graceful // Sous redhat

```

Ensuite, modifiez le mot de passe de l'utilisateur backuppc de l'interface web :

```

# htpasswd -b /etc/backuppc/htpasswd backuppc <nouveaumotdepasse>

```

Ensuite, on arrive à se connecter à l'interface Web de BackupPC avec l'utilisateur **backuppc** et le <nouveaumotdepasse> saisi avec htpasswd.

#### 4- Ajout des PCs à backuper :

D'abord, générer une clé SSH pour l'utilisateur système **backuppc**. En effet, il y'a un utilisateur du système, **backuppc** qui est créé lors de l'installation de BackupPC et c'est **backuppc** qui exécute le programme BackupPC.

```
# su - backuppc // Se connecter en tant que backuppc
```

```
# ssh-keygen // Générer la clé SSH pour backuppc : la clé se trouve par défaut dans le répertoire /var/lib/backuppc/.ssh/
```

Une fois que la clé est générée, on peut commencer à enregistrer des machines cibles à backuper.

Pour ajouter une machine, on édite le fichier **/etc/backuppc/hosts** : puis on rajoute un hôte par ligne en respectant la syntaxe du fichier.

Exemples:

```
<nomdemachinecible>adresseIPcible> 0 backuppc [<autreutilisateur>]
roger.yerbynet.com 0 backuppc
222.3.2.1 0 backuppc thomas
```

Puis, recharger la config de backuppc : `# invoke-rc.d backuppc reload`

Nos nouveaux PCs sont donc pris en charge. Il faut se rendre ensuite sur l'interface Web de backuppc : `http://<IPServeurBackupPC>/backuppc`

Dans la liste déroulante de gauche, on voit bien apparaître les machines nouvellement rajoutées. On sélectionne une machine, puis on lance une sauvegarde complète pour celle-ci en cliquant sur le bouton « Démarrer la sauvegarde complète ». Evidemment, cela ne marche pas, mais cela a pour effet de créer la structure arborescente pour ce PC dans **/var/lib/backuppc/pc/**. Donc en gros, on notera la présence du répertoire **/var/lib/backuppc/pc/<nomdemachinecible>/**.

On peut ensuite ajouter dans ce répertoire un fichier **config.pl** pour customiser un peu la sauvegarde pour cette machine : par exemple, en changeant la méthode de sauvegarde, ou en précisant des répertoires à exclure ou à rajouter dans la sauvegarde, des fichiers à exclure, ...

Ensuite, copier la clé SSH de **backuppc** sur le root de la machine cible :

```
# ssh-copy-id -i /var/lib/backuppc/.ssh/id-rsa.pub root@<nomdemachinecible>
```

Puis en tant que utilisateur **backuppc**, essayer de se connecter à la cible.

```
(# su - backuppc )
```

```
# ssh -l root <nomdemachinecible> // Normalement, ça doit se connecter sans demander de mot de passe, et ça permet aussi de recevoir le fingerprint de la machine.
```

Attention : il faut donc qu'il y'ait du SSH installé sur la machine cible, ainsi que rsync si la méthode utilisée pour le backup est rsync.

Ensuite, on repart sur l'interface Web, puis on relance le backup de la cible qui devrait normalement marcher.

#### 5- Restauration avec BackupPC :

C'est bien de sauvegarder, mais il faut aussi savoir restaurer. Sinon, ça ne sert à rien de sauvegarder.

Ce qui est bien avec BackupPC, c'est que tout est facile, et je me demande bien pourquoi j'écris ce document. Mais bon, puisqu'on l'a commencé ...

Pour restaurer (également très facile), il suffit d'aller sur l'interface Web, de choisir le PC à restaurer, puis la sauvegarde qu'on veut (en fonction de la date de sauvegarde), puis le répertoire, et enfin le fichier à restaurer (peut être un répertoire aussi). On sélectionne, puis on click sur « Restaurer ... », on choisit la méthode de restauration, et on restaure tout simplement comme de l'eau à boire.

FIN de BackupPC

## Complément : BACKUP de BackupPC :

On ne peut pas faire de backup, sans faire de backup de ce backup. Sinon, ça ne servirait à rien. Le jour où le serveur de backup tombe en panne, ou brûle, ou tombe dans de l'eau; ou explose, ... qu'allons nous faire ? On aura l'air trop con. Bref ! On fait donc un backup de backup, voire même un backup de backup de backup...

Avec notre serveur de backup, nos backups se trouvent dans `/var/lib/backuppc/`, c'est donc se répertoire qu'il faut sauvegarder.

Plusieurs techniques existes, mais en gros, ce qu'il faut retenir, c'est qu'il est juste un peu intelligent de ne pas avoir le backup du backup pas dans le même local que le backup. Après on peut faire tout ce qu'on veut pour réaliser ce papa de backup ; par exemple, en mettant un autre serveur dans un autre local qui vient copier les données régulièrement, ou bien faire des sauvegardes périodiques sur disques durs externes qu'on cache ailleurs, ou même combiner les deux méthodes, et avoir des bandes magnétiques aussi... Bon, nous ce qu'on choisit, c'est la plus simple, parce que déjà, on est quand même au niveau de backup de backup, sinon ça commence à devenir trop sophistiqué, et là on s'y perd tous. (Je commence à dire des bêtises là : OBJECTION)

Alors, pour sauvegarder le BackupPC sur un disque dur externe, on fait tout simplement de la copie de partition avec partimage.

```
# partimage <options> /<partition> /<disquedurexterne>/<fichierimage>
```

En gros, y'a que ça.

**Détails** : Avec un disque dur externe USB de 700 Go reconnu en tant que `/dev/sdc` : (toutes les commandes) :

```
# fdisk /dev/sdc // avec ça, je vais créer une partition /dev/sdc1
```

```
# pvcreate /dev/sdc1 // Créer un PV
```

```
# vgcreate usb0 /dev/sdc1 // Création du VG usb0
```

```
# lvcreate -L 698G -n backup usb0 // Création du LV backup dans le VG usb0
```

On a donc une partition LVM `/dev/usb0/backup` sur notre disque dur externe.

```
# mkfs.ext3 /dev/usb0/backup // On formate la nouvelle partition en ext3. On pourrait le faire aussi en ext2. Bon non, on garde ext3, y'a de la journalisation, ça pourrait être utile.
```

```
# mkdir /sauvegarde // On crée un répertoire
```

```
# mount -t ext3 /dev/usb0/backup /sauvegarde // On la monte, cette partition du disque externe
```

Supposons donc que notre **/var/lib/backuppc** est en fait le point de montage de **/dev/sata0/backup**.  
# *lvcreate --snapshot --size 10G --name backupsave /dev/sata0/backup // C'est là que c'est très intéressant. Partimage ne peut pas utiliser une partition montée et donc en cours d'utilisation. Heureusement que LVM permet de prendre une vue (snapshot) d'une partition à un instant t donné, et c'est cette vue qui sera sauvegardée. Lvcreate crée un snapshot (snapshot, genre une photo de la partition), size = 10G précise que les changements opérés sur la partition seront sauvegardés dans un espace de 10Go. En effet, snapshot, ne copie pas vraiment les données de la partition dans un autre espace disque, il crée juste un endroit où il garde les traces des modifications opérées sur la partition. Toute modification faite sur la partition réelle est enregistrée, ce qui permet de voir la partition dans son état au moment de la création du snapshot (et donc d'avoir notre snapshot). (C'est plus malin quand même).*

Ensuite, on crée un répertoire pour monter et démonter notre gentil snapshot. C'est con mais, c'est comme ça ; sans ça, partimage qui vient juste après ne veut pas marcher.

```
# mkdir /tmp/quedale  
# mount /dev/sata0/backupsave /tmp/quedale && umount /dev/sata0/backupsave
```

Démarrons maintenant notre sauvegarde de sauvegarde avec partimage :

```
# partimage [options] save /dev/sata0/backupsave /sauvegarde/image
```

Où [options] égal : -b -B '=' : pour avoir le mode batch et non interactif avec l'interface bidon là !!

-z0 : pas de compression (peut être z1 ou z2 ==> compression)

-o (--overwrite) : Effacer les sauvegardes précédentes

-d (--nodescr) : Pas de demande de description pour aucune image

-M (--nonmbr) : Pas de backup de MBR

-c (--nocheck) : Pas de check de partition avant de commencer à travailler

Notre backup de backup est donc faite et se trouve sur le disque dur externe USB. On peut donc démonter la partition, rendre le lvm inactif et retirer le disque pour le conserver ailleurs loin du serveur.

```
# umount /sauvegarde  
# vgchange -an /dev/usb0
```

## **Restauration du backup de backup :**

Brancher le disque dur USB

Monter la partition du disque :

```
# vgchange -ay /dev/usb0  
# mount /dev/usb0/backup /sauvegarde
```

Puis restaurer :

```
# partimage [options] restore /dev/<autrepartition> /sauvegarde/image
```

ou bien :

```
# partimage [options] restore /dev/sata0/backup /sauvegarde/image // restauration directe de la partition de backuppc
```

Ensuite, on monte la partition **/dev/<autrepartition>** et on retrouve nos backups de PCs.

FIN de FIN de Backup de backup

*[Bon ! Maintenant il faut faire des scripts de sauvegardes automatiques si cela vous intéresse, et faire un peu de crontab. C'est plus intéressant en tout cas. Encore plus intéressant, utiliser udev pour lancer des scripts quand les disques se branchent. Peut être utile de tester les disques au*

*moment où on les branche. Comme ça, en cas de problème, on cherche à résoudre avant que le programme de backup de backup (avec partimage) se lance. Peut être que si je suis inspiré je ferai une documentation de ça une prochaine fois]*

Sources :

<http://wiki.auf.org/wikiteki/ZAO/Dakar/Configuration/Backup> +++++ ;-)  
clin d'oeil à Jérôme. Toutes ces idées sont de lui.

<http://backuppc.sourceforge.net/faq/BackupPC.html> +++++

[http://www.klipz.fr/tutoriel\\_backuppc.html](http://www.klipz.fr/tutoriel_backuppc.html) +++

<http://doc.ubuntu-fr.org/backuppc> -

© Octobre 2008  
Roger YERBANGA  
[www.yerbynet.com](http://www.yerbynet.com)