
Rabanov
Rabanov Hotel vendée
Version <1.0>

HOTEL
RABANOV



Mise en place du Plan de Reprise d'Activité

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

Historique des révisions

Date	Version	Description	Auteur
25/04/2016	<1.0>	Mise en place du pra	Brice Harismendy

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

Table des matières

1. Introduction
 - 1.1 Contexte du projet
 - 1.2 Objectifs du document
 - 1.3 Portée
2. Éléments de configuration
 - 2.1 Installation Mysqldump
 - 2.2 Ajout de l'utilisateur de sauvegarde sous linux
 - 2.3 Génération des clés
 - 2.4 On envoie la clé sur le serveur de sauvegarde :
 - 2.5 Création du répertoire de sauvegarde
3. Le script de sauvegarde
 - 3.1 Création du script
 - 3.1.1 Décompression du tar.gz 7
 - 3.2 transformation du script en commande linux
4. Automatisation de la sauvegarde
5. Conclusion

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

Mise en place du serveur WEB

1. Introduction

L'hotel Rabanov vient d'être construit et un pra est nécessaire il faut le sécuriser et rédiger une documentation technique permettant de le reconfigurer en cas de grave problème. Le pra sert à remettre en place le serveur en cas de grave problème avec une sauvegarde tous les jours de tout le répertoire web

1.1 Contexte du projet

Un groupe Hotelier international a lancé sur la cote vendéenne la construction d'un nouvel hotel 4 étoiles de luxe « Le Rabanov »

1.2 Objectifs du document

Ce document a pour objectif de permettre la reconfiguration rapide du pra web de l'hotel

1.3 Portée

Cette documentation s'adresse à l'équipe technique de l'hotel Rabanov

2. Éléments de configuration

2.1 Installation Mysqldump

-Installer "mysql-client-(la version 5.5 car antérieure ne sont plus fonctionnelles) :

root@debian: apt-get install mysql-client-5.5

```
root@debian:~# apt-get install mysql-client-5.5
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
mysql-client-5.5 est déjà la plus récente version disponible.
mysql-client-5.5 passé en « installé manuellement ».
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
```

on peut désormais sauvegarder en local avec cette commande :
mysqldump --opt -u root -p "gsb">/home/sauvegarde/gsb.sql

2.2 Ajout de l'utilisateur de sauvegarde sous linux

On va créer un utilisateur qui aura les droits nécessaires pour sauvegarder sur le serveur

adduser sauvegarde
adduser sauvegarde www-data

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

remplissez le questionnaire en fonction de vos choix

```

root@debian:~# adduser sauvegarde
Ajout de l'utilisateur « sauvegarde » ...
Ajout du nouveau groupe « sauvegarde » (1002) ...
Ajout du nouvel utilisateur « sauvegarde » (1001) avec le groupe « sauvegarde »
...
Création du répertoire personnel « /home/sauvegarde »...
Copie des fichiers depuis « /etc/skel »...
Entrez le nouveau mot de passe UNIX :
Retapez le nouveau mot de passe UNIX :
passwd : le mot de passe a été mis à jour avec succès
Modification des informations relatives à l'utilisateur sauvegarde
Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée
    Nom complet []: sauvegarde
    N° de bureau []:
    Téléphone professionnel []:
    Téléphone personnel []:
    Autre []:
Cette information est-elle correcte ? [O/n]o

```

pour ce qui est des droits après test j'ai réussi à sauvegarder les données dans le répertoire perso de l'utilisateur sauvegarde

2.3 Génération des clés

Pour envoyer le fichier sans mot de passe on va utiliser des clés public/privé
sur le serveur de développement on exécute la commande suivante dans le fichier root

root@debian: ssh-keygen -t dsa → elle crée le couple de clés on prend les paramètres par défaut (pas de mot de passe)

2.4 On envoie la clé sur le serveur de sauvegarde :

Pour pouvoir envoyer nos fichiers sans mot de passe on envoie la clé publique sur le serveur de sauvegarde :

dans l'ordre :

sur le serveur de sauvegarde :

root@debian: mkdir /root/.ssh

sur le serveur émetteur des sauvegardes :

root@debian: scp /root/.ssh/id_dsa.pub root@192.168.28.22:/root/.ssh/authorized_keys

sur le serveur de sauvegarde :

root@debian: chmod 0600 /root/.ssh/authorized_keys

on peut maintenant envoyer des fichiers avec la commande suivante :

root@debian: scp -B /home/sauvegarde/salut root@192.168.28.22:/home/

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

-B : mode bash ça évite qu'il demande un mot de passe et utilise les clés
 scp : secure copy
 /home/sauvegarde/ : repertoire contenant le fichier "salut" (le fameux "hello world" pour tester l'envoi de fichier)
 root@192.168.28.22 : connection SSH login@ip du serveur
 :/home/ : repertoire de destination sur le serveur destination

2.5 Création du repertoire de sauvegarde

Sur le serveur linux de sauvegarde nous allons créer un repertoire de sauvegarde :

```
root@debian: mkdir /srv/sauvegarde/
```

j'ai fait le choix de mettre les sauvegardes dans ce repertoire par simplicité

3. Le script de sauvegarde

3.1 Création du script

Voici le script de sauvegarde (il ne fait qu'une sauvegarde complète) :
 nous allons créer un fichier shell dans le dossier /home/sauvegarde et ensuite on va le remplir de code.

```
root@debian: nano /home/sauvegarde/sauvegarde.sh
```

```
#!/bin/bash → interpréteur pour les commandes
```

```
echo lancement de la sauvegarde !!!! → message indiquant le début de la procédure de sauvegarde
```

```
rm -rf /home/sauvegarde/backup_* → supprime tout les repertoire dont le nom commence par  
« backup_ » dans le dossier qui sert a entreposer la sauvegarde.
```

```
mkdir /home/sauvegarde/backup_`date +%d_%m_%G` → crée un repertoire commençant par  
« backup_ » suivi du jour (%d) suivi du mois (%m) et de l'année(%G) le tout séparé par des « _ »
```

```
cd backup* → on entre dans le fichier tout juste créé
```

```
cp -R /home/Lmattina /home/sauvegarde/backup_* → copie de l'utilisateur Lmattina
```

```
cp -R /home/Nmoreau /home/sauvegarde/backup_* → copie de l'utilisateur Nmoreau
```

```
cp -R /var/www/public_html/ /home/sauvegarde/backup_* → copie du dossier public_html contenant  
tout les sites web
```

```
mysqldump --opt -u Save 'gsb'>/home/sauvegarde/backup_`date +%d_%m_%G`/BackupGsb.sql
```

```
mysqldump --opt -u Save 'mysql'>/home/sauvegarde/backup_`date +%d_%m_%G`/BackupMysql.sql
```

```
→ (pour les deux lignes précédentes) sauvegarde des base de donnée grace a l'utilisateur « Save »
```

```
cd /home/sauvegarde/
```

```
tar -zcvf backup_`date +%d_%m_%G`.tar.gz backup_`date +%d_%m_%G`/ → compression du fichier  
contenant les données a sauvegarder
```

```
rm -rf /home/sauvegarde/backup_`date +%d_%m_%G` → suppression du repertoire contenant les donnée
```

GSB	Version: <1.0>
Mise en place du pra	Date: 25/04/2016

a sauvegarder (elle sont dans le fichier compressé)

scp -B /home/sauvegarde/backup_* root@192.168.28.22:/srv/sauvegarde/ → envoi du fichier de sauvegarde sur le serveur distant

echo sauvegarde terminé !!! → message signifiant la fin de la sauvegarde

l'avantage de ce script est qu'il fait une sauvegarde même si il ne l'envoie pas sur le serveur distant

3.1.1 Décompression du tar.gz

Pour décompresser le fichier de sauvegarde il suffit de faire :

```
tar -zxvf Lefichier.tar.gz
```

exemple :

```
tar -zxvf backup_30_11_2015.tar.gz
```

3.2 transformation du script en commande linux

root@debian: mv sauvegarde.sh sauvegarde → on renome le script pour enlever l'extension

root@debian: mv sauvegarde /usr/bin/ → on met le script dans un dossier connus de la variable PATH (on aurait pus également pus la modifier)

maintenant notre script est devenue un fichier qui s'exécute automatiquement quand on écrit 'sauvegarde' dans le terminal.

4. Automatisation de la sauvegarde

Jusqu'à présent notre script sauvegarde fonction manuellement nous allons l'automatiser grâce a crontab

root@debian: crontab -e → '-e' pour 'edit' ça permet d'éditer le fichier des tâches automatisé

on rajoute l'entrée suivante à la suite des commentaires :

```
0 20 * * * sauvegarde
```

explication :

0 : a la minute 0

20 : a la 20^{ème} heure de la journée

1^{er} * : chaque du mois

2^{ème} * : chaque mois

3^{ème} * : chaque jour de la semaine

il faut maintenant relancer cron :

```
root@debian: /etc/init.d/cron restart
```

5. Conclusion

Le plan de reprise d'activité est maintenant opérationnel il nous permettra de restaurer les données en cas de panne critique du serveur de production