

Utilitaire d'archivage

Documentation utilisateur

Justin Bossard

Antoine Feuillette

Table des matières

1	Dépendances	2
1.1	Ordinateur client	2
1.2	Serveur d'archivage	2
2	Utilisation	2
3	Configuration du script	2
3.1	Configuration générale	2
3.2	Serveur d'archivage	3
3.3	Envoi de mails	3
3.3.1	Serveur SMTP	3
4	Configuration de l'environnement	3
4.1	Configuration du SSH pour une exécution en autonomie	3
4.2	Configuration du Cron pour une exécution quotidienne	4

1 Dépendances

1.1 Ordinateur client

On considère un système GNU/Linux muni des commandes de bases, donc avec [GNU coreutils](#). Ce script dépend en plus de :

- [wget](#) : pour télécharger l'archive
- [unzip](#) : pour extraire le dump SQL de l'archive
- [tar](#) : pour créer la nouvelle archive
- [OpenSSH](#) : en client uniquement, pour communiquer avec le serveur de sauvegarde
- [Mutt](#) : pour communiquer par mail. Si la fonctionnalité de mail est désactivé, cette dépendance peut-être omise.

1.2 Serveur d'archivage

Le serveur d'archivage doit également être un système GNU/Linux muni de [GNU coreutils](#), ainsi que de [GNU findutils](#) (pour pouvoir supprimer les fichiers trop anciens). [OpenSSH](#) doit également être installé et le serveur accessible et configuré.

2 Utilisation

Pour exécuter le script, il faut lui donner les bonnes permissions (`chmod +x /path/to/archive.sh`), puis l'exécuter (`/path/to/archive.sh`). À chaque exécution, il faut saisir deux fois le mot de passe de l'utilisateur SSH renseigné dans la configuration. Pour que le script soit totalement autonome, voir [plus loin](#).

Le script est configurable via le fichier `archive.conf`, voir la section suivante pour plus de détails.

Il ne faut rien ajouter dans le dossier pour le bon fonctionnement du script.

De plus, l'archive renseignée doit contenir uniquement le dump SQL, sans sous-dossier ni autres fichiers.

3 Configuration du script

Ce script est entièrement configurable via les variables présentes dans le fichier `archive.conf`.

3.1 Configuration générale

emplacementLog Définit où les logs sont enregistrés. Attention, l'utilisateur qui exécute le script doit avoir les droits d'écriture dans le dossier parent.

Par défaut sur `./archive.log`.

logStdout En cas d'échec, rediriger le motif à la sortie standard (0) ou pas (1).

Par défaut sur 0.

archiveURL Définit l'emplacement de l'archive via une URL, accessible depuis l'ordinateur client.

3.2 Serveur d'archivage

adresseArchivage Adresse IP du serveur d'archivage, qui doit être accessible via SSH. Si le port 22 est utilisé, pas besoin de le préciser.

usernameSSH Nom d'utilisateur à utiliser sur le serveur d'archivage

pathSSH Chemin sur lequel enregistrer les archives sur le serveur. Le chemin doit déjà exister et être accessible en lecture et écriture pour l'utilisateur renseigné à **usernameSSH**.

dureeConservation Durée à partir de laquelle les anciennes archives seront supprimées, en jours.

Par défaut sur 30.

3.3 Envoi de mails

envoyerMail Dans quel cas envoyer un mail, jamais (0), en cas d'échec de l'exécution (1) ou toujours (2).

Par défaut sur 1.

mailDestinataires=(dest1@mail.org dest2@mail.org) Destinataires du mail, séparés par des espaces. Si cette liste est vide, et peut importe la valeur de **envoyerMail**, le programme quittera sans envoyer de mail et sans erreur.

objSucces Objet du mail à envoyer en cas de succès.

Par défaut sur Archivage du \$(date +%d %B %Y') réussi.

objEchec Objet du mail à envoyer en cas d'échec.

Par défaut sur Archivage du \$(date +%d %B %Y') échoué.

joindreLog Dans quelle situation joindre le fichier de logs complet, jamais (0), en cas d'échec (1) ou toujours (2).

Attention, il s'agit du fichier de log entier, le motif d'échec, si c'est le cas, est toujours indiqué dans le corps du message.

Par défaut sur 1

muttrcUtilisateur Utiliser le ~/.muttrc de l'utilisateur (0) ou non (1).

Par défaut sur 1.

3.3.1 Serveur SMTP

Ces options n'auront aucune incidence si **muttrcUtilisateur=0**.

“

4 Configuration de l'environnement

4.1 Configuration du SSH pour une exécution en autonomie

Pour plus d'informations : <https://www.cyberciti.biz/tips/ssh-public-key-based-authentication-how-to.html>.

Pour une exécution entièrement autonome, il est nécessaire d'authentifier sa machine par clés SSH auprès du serveur d'archivage. Pour cela, depuis la machine sur laquelle on va exécuter le script :

1. `ssh-keygen -t rsa`

Un mot de passe est demandé, si aucun n'est voulu presser Entrée deux fois.

2. `ssh-copy-id -i ~/.ssh/id_rsa.pub $username@$ip`

Avec `$username` le nom d'utilisateur sur le serveur d'archivage et `$ip` l'adresse IP du serveur. Le mot de passe de l'utilisateur du serveur doit être renseigné.

4.2 Configuration du Cron pour une exécution quotidienne

Pour exécuter le script tous les jours à 3h00, avec un cron déjà configuré, après `crontab -e` (depuis un utilisateur qui a les droits nécessaires pour exécuter le script), ajouter à la fin du fichier ouvert :

```
0 3 * * * /path/to/archive.sh
```