

Mémoire technique

Utilitaire d'archivage

Justin Bossard

Antoine Feuillette

Octobre 2024

Table des matières

1	Utilisation des variables	2
1.1	Variables du fichier de configuration	2
1.1.1	Configuration générale	2
1.1.2	Serveur d'archivage	2
1.1.3	Envoi de mails	2
1.2	Variable utilisée dans le script	3
2	Organisation	3
2.1	Script (<code>archive.sh</code>)	3
2.1.1	Fonction d'écriture de logs	3
2.1.2	Fonction d'envoi de mails	4
2.1.3	Fonction de combinaison	5
2.2	Fichier de configuration (<code>archive.conf</code>)	5
2.3	Fichier de journaux (<code>archive.log</code>)	6
2.4	Fichier de somme de contrôle (<code>.prevChecksum</code>)	6
3	Principe de fonctionnement	6
4	Justification des choix techniques	6
4.1	Envoi de mails	6
4.2	Serveur Web	7

1 Utilisation des variables

1.1 Variables du fichier de configuration

Le fichier de configuration `archive.conf` comprend les variables suivantes :

1.1.1 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, redirige 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.

1.1.2 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.

1.1.3 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.

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

serveurHote Serveur SMTP qui gère l'envoi de mails.

port Port sur lequel contacter le serveur. En général, on a :

- 25 : sans chiffrement
- 465 : chiffrement implicite (TLS/SSL)
- 587 : chiffrement explicite (STARTTLS)

mailEnvoyeur Mail envoyeur des informations de l'utilitaire, enregistré sur le serveur renseigné dans **serveurHote**.

motDePasse Mot de passe associé au mail pour s'identifier sur **serveurHote**.

1.2 Variable utilisée dans le script

Toutes les variables du fichier de configuration sont utilisées dans le script. On ne définit qu'une variable dans le script, comme tel :

```
currentChecksum=$(sha256sum ./*.sql | cut -d ' ' -f1)
```

Cela permet de stocker la somme de contrôle du dump SQL en cours de traitement pour pouvoir la comparer avec la somme de contrôle sauvegardée.

2 Organisation

On a l'organisation de fichier suivante :

```
.
|- archive.sh      # Script
|- archive.conf    # Fichier de configuration
|- archive.log     # Logs du script, possibilité de modifier l'emplacement
|- .prevChecksum  # Somme de contrôle du précédent fichier
```

2.1 Script (archive.sh)

Pour simplifier la lecture et l'écriture du programme, nous avons écrit une fonction pour écrire les logs, une fonction pour envoyer un mail, et une fonction qui combine les deux.

2.1.1 Fonction d'écriture de logs

```
1 function ecrireLog() {
2     if [[ $1 -eq 0 ]]; then
3         echo "[ $(date +%T - %d %b %Y) ] : Succès, checksum=$2" >> "$emplacementLog"
4     else
5         [[ $logStdout -eq 0 ]] && echo "$2"
6         echo "[ $(date +%T - %d %b %Y) ] : Échec, $2" >> "$emplacementLog"
7     fi
8 }
```

Cette fonction admet deux paramètres, le premier définit le succès ou l'échec de l'opération, le second dépend du premier :

- en cas de succès, il s'agit de la somme de contrôle du fichier SQL sauvegardé
- en cas d'échec, il s'agit du motif de l'échec

On ajoute ensuite à la fin du fichier de log la date et l'heure d'écriture, si l'opération est un succès ou non, suivi de la checksum dans le premier cas et du motif d'échec dans le second.

2.1.2 Fonction d'envoi de mails

```
1 function envoyerMail() {
2     if [[ ${#mailDestinataires[*]} -ne 0 && (($1 -eq 1 && $envoyerMail -eq 1) \
3         || $envoyerMail -eq 2) ]]; then
4
5         if [[ $muttrcUtilisateur -eq 0 ]]; then
6             echo "$([[ $1 -eq 0 ]] && echo L'opération de ce jour est un succès \
7                 || echo $2)" | \
8             mutt -x \
9                 -s "$([[ $1 -eq 0 ]] && echo $objSucces || echo $objEchec)" \
10                "$([[ $joindreLog -eq 2 || ($1 -eq 1 && $joindreLog -eq 1) ]]" \
11                && echo "-a $emplacementLog --)" \
12                "$(echo ${mailDestinataires[*]})"
13
14         else
15             echo "$([[ $1 -eq 0 ]] && echo L'opération d'archivage de ce jour \
16                 est un succès. || echo $2)" | \
17             mutt -nx \
18                 -e "set from = \"$mailEnvoyeur\"" \
19                 -e "set smtp_pass = \"$motDePasse\"" \
20                 -e "set smtp_url = \"smtps://$mailEnvoyeur@$serveurHote:$port\"" \
21                 -e "set send_charset = \"utf-8\"" \
22                 -s "$([[ $1 -eq 0 ]] && echo $objSucces || echo $objEchec)" \
23                 "$([[ $joindreLog -eq 2 || ($1 -eq 1 && $joindreLog -eq 1) ]]" \
24                 && echo "-a $emplacementLog --)" \
25                 "$(echo ${mailDestinataires[*]})"
26         fi
27     fi
28
29     [[ $? -ne 0 ]] && ecrireLog 1 "erreur lors de l'envoi du mail."
30 }
```

Cette fonction admet également deux paramètres, le premier définit le succès ou l'échec de l'opération, le second, n'est utilisé qu'en cas d'échec, et contient alors le motif de celui-ci.

Le booléen utiliser par le premier **if** permet d'envoyer un mail si la liste de destinataires du fichier de configuration n'est pas vide *et* que on est en situation d'envoyer un mail (échec et `envoyerMail=1`, ou `envoyerMail=2`).

On vérifie ensuite si l'utilisateur veut utiliser sa configuration de **mutt** ou non. Si ce n'est pas le cas, on précise que **mutt** ne doit pas utiliser le fichier de configuration de l'utilisateur (option `-n`), et on passe les paramètres nécessaires à l'envoi d'un mail en argument (option `-e "set option = \"valeur\"" :`

- **from** permet de définir l'identité de l'expéditeur. On utilise l'adresse mail renseignée dans le fichier de configuration
- **smtp_pass** le mot de passe qui permet de se connecter au serveur SMTP
- **smtp_url** Définit le serveur auquel se connecter et le protocole de connexion (**smtps** ici). On renseigne également le port de connexion défini dans la configuration.
- **send_charset** Définit l'encodage du corps du mail et des pièces jointes. On choisit **utf-8** pour pouvoir prendre en compte l'ensemble des caractères Unicode.

On définit le sujet du mail avec l'option **-s**, qui varie en fonction de l'aboutissement de l'opération (on utilise la syntaxe : `[[succès ?]] && oui || non`).

Ensuite, selon les préférences de l'utilisateur, on joint ou non le fichier de log. Pour cela, on vérifie s'il faut joindre systématiquement le log, ou s'il faut le joindre en cas d'échec et que c'en est un. Si la condition est vraie, on renvoie l'option **-a** avec le chemin de log, sinon on ne renvoie rien.

Enfin, on affiche tous les éléments de la liste des destinataires.

2.1.3 Fonction de combinaison

```

1 function combo() {
2     if [[ "$1" -eq 0 ]]; then
3         ecrireLog 0 "$2"
4         envoyerMail 0
5     else
6         ecrireLog 1 "$2"
7         envoyerMail 1 "L'opération d'archivage de ce jour \
8             a échoué pour le motif suivant : $2"
9     fi
10 }
```

Cette fonction combine simplement les deux fonctions précédentes, et permet d'unifier les motifs d'arrêts de fonctions. En premier argument, on définit le succès (0) ou l'échec (1) de l'opération. En second, on donne la checksum du fichier concerné en cas de succès, ou le motif de l'arrêt en cas d'erreur.

2.2 Fichier de configuration (**archive.conf**)

Le fichier de configuration est organisé en plusieurs sections :

1. Configuration générale
2. **Configuration du serveur d'archivage** Permet de configurer la connexion à un serveur SFTP via SSH
3. **Configuration des mails** Permet de configurer les infos générales des mails : quand envoyer, objets, destinataires...
 - **Configuration serveur SMTP** Permet de définir comment se connecter au serveur d'envoi SMTP spécialement pour ce script, si nécessaire

2.3 Fichier de journaux (archive.log)

Le fichier de journalisation a la syntaxe suivante pour une ligne (en utilisant une version édulcorée de la syntaxe bash pour succès ou échec) :

```
[ heure - date ] : [[ Succès, checksum || Échec, cause de l'échec ]]
```

Une nouvelle entrée par ligne, en ordre décroissant d'ancienneté.

2.4 Fichier de somme de contrôle (.prevChecksum)

Le fichier `.prevChecksum` n'a pas vocation à être consulté par l'utilisateur, d'où le fait qu'il soit caché. Il permet de stocker la somme de contrôle en 256 bits du dump SQL précédent, pour pouvoir déterminer si des modifications sont advenues.

3 Principe de fonctionnement



4 Justification des choix techniques

4.1 Envoi de mails

Pour configurer l'envoi des mails, on utilise mutt :

- `mailutils` ou `mutt` : pour l'interaction avec l'utilisateur, envoyer et lire des mail

4.2 Serveur Web