

Sauvegarder une BDD

Réaliser une sauvegarde d'une BDD

Pour créer une sauvegarde (**backup** en Anglais) de la base de données sur PostgreSQL c'est très simple car il existe un utilitaire `pg_dump` pour créer une sauvegarde même si la base données est en cours d'utilisation.

Les sauvegardes peuvent être produits dans des formats de script ou de fichier d'archive. elles sont constitué des fichiers en texte clair contenant les commandes SQL nécessaires pour reconstruire la base de données dans l'état où elle se trouvait au moment où elle a été enregistrée.

Tout d'abord nous allons créer un répertoire de sauvegarde pour centraliser les sauvegarde de base de données. Puis nous allons créer une sauvegarde pour vérifier que tout fonctionne. Ensuite nous allons créer un script de sauvegarde de BDD qui s'exécutera périodiquement.

Création du répertoire des sauvegardes

```
sudo mkdir -p /var/backup
```

Créer une sauvegarde

Commande pour créer une sauvegarde. Remplacer les variables par les bonnes valeur avant de l'exécuter.

<code>\${DB_USER}</code>	Utilisateur ayant accès à la base de données
<code>\${DB_PASSWORD}</code>	Mot de passe de l'utilisateur
<code>\${DB_HOST}</code>	Host de la base de données : Adresse IP ou nom de domaine
<code>\${DB_NAME}</code>	Nom de la base de données

```
/usr/bin/pg_dump postgresql://${DB_USER}:${DB_PASSWORD}@${DB_HOST}/${DB_NAME} | gzip -c > /var/backup/"${DB_NAME}-`date +%d-%m-%y_%T`".sql.gz;
```

Le fichier générer sera sous le nom suivant : `${DB_NAME}-${DATE}_${HEURE}.sql.gz`

Automatiser la sauvegarde

Le processus de remplacement des variables pour réaliser une sauvegarde peut-être long et fatigant. Il est serait plus pratique de créer un script regroupant les démarches précédentes.

Création du script

```
sudo nano save-database
```

Contenu du script

```
#!/bin/bash

#####
# SAVE DATABASE by gpatruno #
#####

# définition des variables
backupdir='/etc/backup';
datefolder=`date +%m-%y`;
datescript=`date +%d-%m-%y_%T`;
DB_USER=postgres
DB_NAME=dbname
DB_PASSWORD=dbpassword
DB_HOST=localhost

echo "Répertoire de destination : $backupdir/$datefolder"
mkdir -p $backupdir/$datefolder;

echo "PostgreSQL Backup for database : ${DB_NAME} start at " `date +%T`;
echo "Exporting ${DB_NAME}";

# Pour conserver la version non compressé du fichier 'gzip -c fichier-de-sortie.ext > fichier-
de-sortie.ext.gz'
# Sinon compresse le fichier fichier-de-sortie et la remplace par la version compressé nommé
avec l'extention *.gz 'gzip fichier-de-sortie.ext.gz'
echo "/usr/bin/pg_dump postgresql://${DB_USER}:${DB_PASSWORD}@${DB_HOST}/${DB_NAME} | gzip -c
> $backupdir/$datefolder/"${DB_NAME}-${datescript}.sql.gz;"
/usr/bin/pg_dump postgresql://${DB_USER}:${DB_PASSWORD}@${DB_HOST}/${DB_NAME} | gzip -c >
```

```
$backupdir/$datefolder/"${DB_NAME}-${datescript"}.sql.gz;  
  
echo "done";  
echo `date +%d-%m-%y` " "`date +%T`;
```

Donner le droit d'exécution

```
sudo chmod +x save-database
```

Exécuter le script

```
./save-database
```

L'exécution du script donne la sortie suivante :

```
Répertoire de destination : /var/backup  
PostgreSQL Backup for database : dbname start at 16:38:07  
Exporting dbname  
/usr/bin/pg_dump postgresql://postgres:dbpassword@localhost/dbname | gzip -c >  
/etc/backup/dbname-03-12-21_16:38:07.sql.gz;  
done  
03-12-21 16:38:07
```

Exécution régulière

Afin de mettre en place une exécution régulière du script nous allons utiliser l'outil Linux appelé `Cron`. C'est un programme disponible sur les systèmes de type Unix (Linux, Mac Osx ...) permettant de planifier des tâches régulières.

Ouvrir Crontab

```
crontab -e
```

Mettre le contenu suivant

```
# Exécution du script tous les jours à 2h du matin  
0 2 * * * /bin/sh /script/save-database
```

La tâche `Cron` ci-dessus sera exécutée tous les jours à 2 heures du matin et lancera un script `save-database` avec l'interpréteur `bash`. Ceci maintiendra la sauvegarde chaque jour.

Revision #2

Created 5 July 2022 13:28:53 by gpatruno

Updated 19 January 2023 15:59:46 by gpatruno