

TP Sauvegarde / Restauration de Base de Données

Application concernée : GLPI

A votre disposition :

- Votre machine GLPI avec sa base de données sous Mysql
- Les fichier shell1.pdf et shell2.pdf

Votre mission : Automatiser la sauvegarde de la base de données glpi en suivant les étapes suivantes :

1. écrire un script bash de sauvegarde,
2. rajouter des fichiers de log,
3. planification avec cron,
4. stratégie de sauvegarde,
5. sauvegarde distante,
6. procédure de restauration

1 Etape 1 : Création de l'environnement de test

Vérifier que les services httpd et mariadb sont bien démarrés :

Pour vérifier le statut de httpd et de mysql :

```
#systemctl status httpd
#systemctl status mariadb
```

Pour démarrer les services de httpd et de mysql :

```
#systemctl start httpd
#systemctl start mariadb
```

Pour qu'ils démarrent au lancement de la machine :

```
#systemctl enable httpd
#systemctl enable mariadb
```

Vérifier que vous accéder à glpi avec un navigateur.

Http://localhost/glpi

Vérifier l'accès en mode commande à la base de données mysql :

Pour entrer dans mysql :

```
mysql -u root -p
```

Dans mysql vérifier la présence de la base de données glpi

```
show databases ;
```

Se positionner dans la nouvelle base

```
use <nom de la base glpi> ;
```

Pour vérifier

```
show tables ;
```

Faire des select pour voir le contenu des tables

```
select id,name,password from glpi_users ;
```

2 Etape2 : Écriture du fichier bash de sauvegarde

Objectif :sauvegarder la base **glpi**. Le fichier de destination sera **/var/sauv/sv_glpi.sql**.

Écrire le script **sauv_bd_glpi** permettant de sauvegarder la base et de compresser le fichier **sv_glpi.sql** généré (utiliser gzip). Sauvegarder le script dans **/root/bash/**

Créer **SI BESOIN** les répertoires suivants :

- Répertoire de stockage du script : **/root/bash**
- Répertoire de destination de la sauvegarde locale : **/var/sauv**

Le nom du script de sauvegarde sera : **sauv_bd_glpi.sh**

L'exécution du fichier bash sera réalisé par l'administrateur du serveur GNU/Linux (compte root). Le nom du serveur mysql est par défaut localhost,

Déroulement du script :

Sauvegarde de la base de données glpi à l'aide de la commande **mysqldump** (vous devez trouver la syntaxe) . Il faut rediriger le résultat du mysqldump vers un fichier nommé sv_glpi.sql

Compresser le fichier sv_glpi.sql à l'aide de la commande gzip (vous devez trouver la syntaxe)

Rendre exécutable votre script, le tester !

```
GNU nano 2.9.8          bash/sauv bd glpi.sh

#!/bin/bash
#=====
host="localhost"
utilisateur="root"
mdp="joliverie"
bdd="dbglpi"
#=====
if [ ! -e /var/sauv/ ]
then
    echo "creation du répertoire /var/sauv/"
    mkdir /var/sauv/
fi
if [ -e /var/sauv/sv_glpi.sql.gz ]
then
    rm -v /var/sauv/sv_glpi.sql.gz
fi
mysqldump -h $host -u $utilisateur -p$mdp $bdd > /var/sauv/sv_glpi.sql
gzip /var/sauv/sv_glpi.sql

[ Lecture de 18 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier  ^C Pos. cur.
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^T Analyse st^_ Aller lig.
```

Le resultat est concluant :

```
[root@glpi151 ~]# ls /var/sauv
sv_glpi.sql.gz
[root@glpi151 ~]#
```

3 Etape 3 : Création de fichiers de log

Vous allez réaliser un fichier de log dans le répertoire /var/log/mysql/ (à créer si besoin)

Le fichier aura dans un premier temps pour nom : sauv_bd_glpi.log

Il sera créer par redirection des messages affichés vers ce fichier. L'opérateur de redirection est > lorsque l'on crée le fichier de destination et >> lorsque l'on ajoute au fichier de destination. Vous reprenez votre script de l'étape 2 pour l'enrichir avec des messages vers le fichier lors de l'exécution.

a) *Début et fin d'exécution*

Au début de l'exécution, il faut envoyer un message :

```
|debut de sauvegarde mardi 19 mars 2013, 08:56:00 (UTC+0100)
```

Idem à la fin

Δ ` s'obtient avec Alt Gr 7 + espace

Exemple :

```
debut="Début de la sauvegarde ; "`date`
echo $debut > /var/log/mysql/sauv_bd_glpi.log
```

b) *Tester la connexion au serveur MySQL avant d'exécuter le mysqldump*

Pré-requis : Tester en ligne de commande le mysqlshow

Tester la connexion à mysql avec un mysqlShow. Si la connexion est établie vous devez trouver le mot Database dans le résultat de cette commande sinon il n'y sera pas.

Expliquer la ligne suivante :

```
Connect=`mysqlshow -h $mysql_server -u root --password=password | grep "Tables"``
```

La rajouter dans votre script avant le mysqldump.

Si \$Connect est = à chaîne vide envoyer un message d'erreur de connexion dans le fichier log, sinon exécuter le mysqldump

```
connect=`mysqlshow -h localhost -u root --password=joliverie | grep "Databases"``
if [ -z "$connect" ]
then
    echo "erreur mysql n'est pas installé ou démarré" >> /var/log/mysql/sauv_
else
    mysqldump -h $host -u $utilisateur -p$mdp $bdd > /var/sauv/sv_glpi.sql
    gzip /var/sauv/sv_glpi.sql
fi
```

4 Étape 4 : Sauvegardes périodiques : planification avec cron

Le daemon (service) **cron** permet de lancer automatiquement des commandes à un horaire fixe.

Complétez la "crontab" de façon à ce que la sauvegarde s'effectue tous les jours, dans 5 minutes.

Vérifiez que cela fonctionne.

```
[root@glpi151 ~]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@glpi151 ~]# crontab -l
*/5 * * * * /bin/bash /root/bash/sauv_bd_glpi.sh
[root@glpi151 ~]# nano bash/sauv_bd_glpi.sh
[root@glpi151 ~]# cat /var/log/mysql/sauv_bd_glpi.log
Début de la sauvegarde : mer. déc. 2 08:40:02 EST 2020
Fin de Sauvegarde : mer. déc. 2 08:40:03 EST 2020
[root@glpi151 ~]# █
```

Il fait la sauvegarde tout les jours, toutes les 5 min

5 Étape 5 : Stratégie de sauvegarde

- Mettre en place une stratégie de sauvegarde mensuelle, hebdomadaire, journalière
- Faire une proposition justifiée et rédigée
- Réfléchir à la problématique d'écrasement des sauvegardes (ne faut-il pas conserver plusieurs sauvegardes en gérant des fichiers différents et les dates ?)

Pour GLPI je pense qu'une sauvegarde complète hebdomadaire est suffisante, car le parc n'est pas modifié tous les jours, cependant une sauvegarde journalière de la gestion des incidents peut être une bonne idée afin de ne pas perdre des informations liées à des tickets en cours.

Il faudrait sauvegarder les fichiers de la gestion d'incident entre 2 sauvegardes complètes et archiver les sauvegardes complètes sur 6 mois afin de respecter les exigences légales d'archivage.

6 Étape 6 : Sauvegarde proposée par glpi

- A partir du menu Administration choisir Maintenance
- Faites la sauvegarde
- Trouvez ou a été créé le fichier de sauvegarde.

```
[root@glpi151 ~]# ls /var/www/html/glpi/files/_dumps/
glpi-backup-9.4.5-2020-12-02-14-24.sql.gz  remove.txt
```

- Quel est la méthode de nommage des fichiers de sauvegardes
C'est glpi-backup-<la version>-<année>-<moi>-<jour>-<je sais pas>.sql.gz
- Cliquer sur le lien Voir/télécharger le fichier pour visualiser son contenu.

The screenshot shows a text editor window titled 'glpi-backup-9.4.5-2020-12-02-14-24.sql' with a toolbar containing 'Ouvrir', 'Enregistrer', and window control buttons. The content of the file is a SQL dump for the GLPI database, dated 2020-12-02 14:24. It includes the structure for two tables: 'glpi_alerts' and 'glpi_apiclients'.

```
#GLPI Dump database on 2020-12-02 14:24

### Dump table glpi_alerts

DROP TABLE IF EXISTS `glpi_alerts`;
CREATE TABLE `glpi_alerts` (
  `id` int(11) NOT NULL AUTO INCREMENT,
  `itemtype` varchar(100) COLLATE utf8_unicode_ci NOT NULL,
  `items_id` int(11) NOT NULL DEFAULT 0,
  `type` int(11) NOT NULL DEFAULT 0 COMMENT 'see define.php ALERT_* constant',
  `date` datetime NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `unicity` (`itemtype`,`items_id`,`type`),
  KEY `type` (`type`),
  KEY `date` (`date`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

### Dump table glpi_apiclients

DROP TABLE IF EXISTS `glpi_apiclients`;
CREATE TABLE `glpi_apiclients` (
  `id` int(11) NOT NULL AUTO INCREMENT,
  `entities_id` int(11) NOT NULL DEFAULT 0,
  `is_recursive` tinyint(1) NOT NULL DEFAULT 0,
  `name` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `date_mod` datetime DEFAULT NULL,
  `is_active` tinyint(1) NOT NULL DEFAULT 0,
  `ipv4_range_start` bigint(20) DEFAULT NULL,
  `ipv4_range_end` bigint(20) DEFAULT NULL,
  `ipv6` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `app_token` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `app_token_date` datetime DEFAULT NULL,
  `dolog_method` tinyint(4) NOT NULL DEFAULT 0,
  `comment` text COLLATE utf8_unicode_ci DEFAULT NULL,
```

- Extraire de la même façon le fichier xml

```

<?xml version="1.0"?>
<dataxml>
  <fields>
    <field>id</field>
    <field>itemtype</field>
    <field>items_id</field>
    <field>type</field>
    <field>date</field>
  </fields>
</rows>
</rows>
<fields>
  <field>id</field>
  <field>entities_id</field>
  <field>is_recursive</field>
  <field>name</field>
  <field>date_mod</field>
  <field>is_active</field>
  <field>ipv4_range_start</field>
  <field>ipv4_range_end</field>
  <field>ipv6</field>
  <field>app_token</field>
  <field>app_token_date</field>
  <field>dolog_method</field>
  <field>comment</field>
</fields>
</rows>
<rows>
  <row>
    <data fieldname="id">1</data>
    <data fieldname="entities_id">0</data>
    <data fieldname="is_recursive">1</data>
    <data fieldname="name">full access from localhost</data>
    <data fieldname="date_mod"></data>
    <data fieldname="is_active">1</data>
    <data fieldname="ipv4_range_start">2130706433</data>
  </row>
</rows>

```

7 Étape 7 : Restauration

Envisager les différents scénarios ;

- Restauration sur le même sgbd, seule l'intégrité de la base a été touchée
- Restauration sur nouveau serveur.
- On veut restaurer toute la base de données
- On veut restaurer seulement des tables
-

Proposer des solutions suivant les cas, tester ces solutions

Si seul l'intégrité de la base a été touché on peut juste exécuter le sql dump avec mysql ce qui va refaire les tables a la date de la sauvegarde

Sur un nouveau serveur il faudra recréé une base de donnée avec le compte et le mdp a l'identique par rapport a l'ancien serveur il faudra aussi installé un glpi vierge mais en restaurant le fichier php qui contien les conf de base de donnée de GLPI et ensuite il suffira d'exécuté le sqldump avec mysql.