

n2m  
tuto dump serveur

BELTRAMO Emeric

Juin 2021

# Partie 1

## DUMP BDD

Pour sauvegarder toutes les bases de données présentes sur le serveur il faut placer le script suivant sur le serveur. Lors de l'exécution ce script fera un dump via mysql et sauvegardera chaque base dans une archive en **.sql.gz** séparée dans le dossier :

**\backup\bdds\datedujour.**

De plus lors de l'exécution le script supprimera automatiquement les sauvegardes de 15 jours. Afin d'ajouter une base à la liste de celles **que l'on ne veut pas sauvegarder** il faut ajouter **| grep -v nomdelabase** à la suite des autres ligne 19

```
1 #!/bin/bash
2
3 #d finitions des variables :
4 datedujour=$(date +%d-%m-%Y)
5 datequinze=$(date -d '15 days ago' +%d-%m-%Y)
6
7 # On afficher les infos date du jour et autre pour le log.
8 echo "Date du jour : $datedujour"
9 echo "Date il y a 15 jours : $datequinze"
10
11 # attributs pour la connexion la base
12 MYSQLUSER="root"
13 MYSQLPASS="K3fmFLa445xG"
14 MYSQLHOST="127.0.0.1"
15 # Sauvegarde de l'ensemble des bases de donnees
16 BACKUP_DIR=/backup/bdds/$datedujour
17 BACKUP_DIR_OLD=/backup/bdds/$datequinze
18 test -d $BACKUP_DIR || mkdir -p $BACKUP_DIR
19 # Get the database list , exclude information_schema
20 DB_LIST=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --password=$MYSQLPASS -e '
21     SELECT schema_name FROM information_schema.schemata' | grep -v
22     information_schema | grep -v mysql | grep -v performance_schema | grep -v sys
23     | grep -v backup)
24 echo ${DB_LIST}
25 for db in ${DB_LIST}
26 do
27     # dump each dabase in a separate file
28     mysqldump -h $MYSQLHOST -u $MYSQLUSER --password=$MYSQLPASS $db | gzip >
29     $BACKUP_DIR/$db.sql.gz
30 done
31
32 # Suppression du re pertoire de 15 jours s'il existe
33 if test -d ${BACKUP_DIR_OLD}; then
```

```
30 echo "suppression des anciens backups de BDD vieux de 15 jours : $datequinze"  
31 rm -rf ${BACKUP_DIR_OLD}  
32 fi
```

## Partie 2

# DUMP FILES

### 2.1 la partie base

Pour rendre la gestion des fichiers et dossiers à sauvegarder facilement paramétrable nous avons fait le choix de créer une base de données **backup** ainsi qu'une table **backup\_list** avec le schema suivant. Pour ajouter un élément à sauvegarder il suffit d'ajouter un tuple dans la base en spécifiant ces informations

```
1 create database backup
2
3 create table backup_list
4 (
5     id int NOT NULL AUTO_INCREMENT,      —identifiant du tuple
6     name varchar(50) not null,           —nom du fichier ou dossier
7     sauvegarder
8     path varchar(500) not null,          —path complet de l'élément
9     sauvegarder
10    saved_path varchar(500) not null,      —path du lieu de sauvegarde
11    day_1 boolean not null,               —true on sauvegarde le fichier le
12    jour n de la semaine
13    day_2 boolean not null,
14    day_3 boolean not null,
15    day_4 boolean not null,
16    day_5 boolean not null,
17    day_6 boolean not null,
18    day_7 boolean not null,
19    nb_persist integer not null,          —nombre total de sauvegarde de l'élé
20    ment conserver
21    parent_directory varchar(50),         —élément non obligatoire
22    permet de spécifier un dossiersupplémentaire ou sauvegarder l'élément dans le
23    but de grouper les éléments a sauvegarder sur un meme site dans le meme
24    dossier
25
26    primary key (id)
27 );
```

## 2.2 la partie script

Un fois la base renseignée il suffit de déposer le script suivant sur le serveur. Lors de son exécution ce dernier va enregistrer les éléments contenues suivants les informations qui y sont liés (lieu de sauvegarde, jours de sauvegarde,...)

```

1 #!/bin/bash
2
3 #d  finitions des variables :
4 datedujour=$(date +%d-%m-%Y)
5 datequinze=$(date -d '15 days ago' +%d-%m-%Y)
6 day_of_week=$(date +%u)
7
8 # On afficher les infos date du jour et autre pour le log.
9 echo "Date du jour : $datedujour"
10
11
12 # attributs pour la connexion la base
13 MYSQLUSER="root"
14 MYSQLPASS="K3fmFLa445xG"
15 MYSQLHOST="127.0.0.1"
16
17
18
19 for file_id in $(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --password=$MYSQLPASS
    -e 'use backup; select id from backup_list')
20 do
21
22     file_persist=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --password=
    $MYSQLPASS -e "use backup; select day_{$day_of_week} from backup_list where
    id={$file_id}")
23     true="1"
24     if [ "$file_persist" -eq "$true" ]; then
25
26         file_name=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --password=
    $MYSQLPASS -e "use backup; select name from backup_list where id={$file_id}")
27         file_path=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --password=
    $MYSQLPASS -e "use backup; select path from backup_list where id={$file_id}")
28         file_saved_path=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --
    password=$MYSQLPASS -e "use backup; select saved_path from backup_list where
    id={$file_id}")
29         file_parent_directory=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER
    --password=$MYSQLPASS -e "use backup; select parent_directory from
    backup_list where id={$file_id}")
30         file_nb_persist=$(mysql -B -s -h $MYSQLHOST -u $MYSQLUSER --
    password=$MYSQLPASS -e "use backup; select nb_persist from backup_list where
    id={$file_id}")
31
32
33         if [ $file_parent_directory == "null" ]; then
34             BACKUP_DIR=/ $file_saved_path/$datedujour
35             BACKUP_OLD=/ $file_saved_path/*
36         else

```

```

37     BACKUP_DIR=/$file_saved_path/$datedujour/
    $file_parent_directory
38     BACKUP_OLD=/$file_saved_path/*/ $file_parent_directory
39     fi
40     test -d $BACKUP_DIR || mkdir -p $BACKUP_DIR
41
42     cd $file_path
43     tar -zcvf $BACKUP_DIR/$file_name.tar.gz $file_name
44
45
46     count=$(find $BACKUP_OLD -name "${file_name}.tar.gz" -type f | wc
-1)
47     echo $count
48
49     if [ "$count" -ge "$file_nb_persist" ]; then
50         oldest=$(find $BACKUP_OLD -name "${file_name}.tar.gz" -
type f -printf '%T+ %p\n' | sort | head -n 1)
51         echo $oldest
52
53         oldest_path=$(cut -d' ' -f2 <<< "$oldest")
54         echo $oldest_path
55         rm -rf $oldest_path
56     fi
57 fi
58 done

```

## Partie 3

# crontab

Afin d'automatiser le processus de sauvegarde nous utilisons crontab. Après l'installation il suffit de faire **crontab -e** pour éditer la configuration et ajouter en fin de fichier. Cela va lancer deux scripts vu auparavant tous les jours à **23h30**. Et sauvegarder les log de l'exécution dans le fichier spécifié. **Attention seuls les logs de la dernière exécutions sont sauvegardés**

```

1 SHELL=/bin/bash
2 30 23 * * * /etc/bdd_dump/mysql-dump.sh > /backup/bdds/log/sauvegarde.log
3 30 23 * * * /etc/files_dump/bash_files.sh > /backup/files/log/sauvegarde.log

```