Faire un Script Crontab de sauvegarde

Sur la VM de Base de données, créer un utilisateur de backup :
Aller dans mariadb
mysql -u root -p

```
root@gsb-bdd:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 29
Server version: 11.8.3-MariaDB-0+deb13u1 from Debian -- Please help get to 10k
tars at https://github.com/MariaDB/Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Puis :
CREATE USER 'backup_user'@'192.168.107.224' IDENTIFIED BY ‘MOT_DE_PASSE’;

```
MariaDB [(none)]> CREATE USER 'backup_user'@'192.168.107.224' IDENTIFIED BY [redacted];
Query OK, 0 rows affected (0,008 sec)

MariaDB [(none)]>
```

GRANT SELECT, LOCK TABLES, SHOW VIEW, EVENT, TRIGGER ON gsb_bdd.* TO 'backup_user'@'192.168.107.224';

```
MariaDB [(none)]> GRANT SELECT, LOCK TABLES, SHOW VIEW, EVENT, TRIGGER ON gsb_bdd.* TO 'backup_user'@'192.168.107.224';
Query OK, 0 rows affected (0,006 sec)

MariaDB [(none)]>
```

FLUSH PRIVILEGES;

```
MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,007 sec)

MariaDB [(none)]>
```

Sur la VM Base de données, faire :
ufw allow from 192.168.107.224 to any port 3306

```
root@gsb-bdd:~# ufw allow from 192.168.107.224 to any port 3306
Rule added
root@gsb-bdd:~#
```

Sur la VM Apache :
Pour la sécurité :
Faire un fichier de conf d’utilisateur sur la VM APACHE avec les infos utilisateurs :

nano /root/.my.cnf

```
[mysqldump]
user=backup_user
password=[redacted]
host=192.168.107.225
port=3306
```

Préparer le script bash :
nano /usr/local/bin/backup_mysql.sh

```
#!/bin/bash


DB_HOST="192.168.107.225"
DB_PORT="3306"
DB_NAME="gsb_bdd"

BACKUP_DIR="/var/backups/mysql"
RETENTION_DAYS=30
DATE=$(date +"%Y-%m-%d_%H-%M-%S")
BACKUP_FILE="${BACKUP_DIR}/${DB_NAME}_${DATE}.sql.gz"

LOG_FILE="/var/log/mysql_backup.log"


log() {
    echo "[$(date '+%Y-%m-%d %H:%M:%S')] $1" | tee -a "$LOG_FILE"
}


mkdir -p "$BACKUP_DIR"
chmod 750 "$BACKUP_DIR"

log "Début du backup de la base : ${DB_NAME}"


mysqldump \
    --host="$DB_HOST" \
    --port="$DB_PORT" \
    --single-transaction \
    --routines \
    --triggers \
    --events \
    "$DB_NAME" | gzip > "$BACKUP_FILE"


if [ $? -eq 0 ] && [ -s "$BACKUP_FILE" ]; then
    log "✓ Backup réussi : ${BACKUP_FILE} ($(du -sh "$BACKUP_FILE" | cut -f1))"
else
    log "✗ ERREUR : Le backup a échoué ou le fichier est vide !"
    rm -f "$BACKUP_FILE"
    exit 1
fi


DELETED=$(find "$BACKUP_DIR" -name "*.sql.gz" -mtime +${RETENTION_DAYS} -print -delete | wc -l)
log "Nettoyage : ${DELETED} fichier(s) supprimé(s) (> ${RETENTION_DAYS} jours)"

log "Backup terminé"
```

Le fichier fermé et enregistré, on s'assurer des droits corrects du dossier

chmod 750 /usr/local/bin/backup_mysql.sh

```
root@gsb-symfony:/var/www/html# chmod 750 /usr/local/bin/backup_mysql.sh
```

On s'assure que root possède bien le dossier
chown root:root /usr/local/bin/backup_mysql.sh

```
root@gsb-symfony:/var/www/html# chown root:root /usr/local/bin/backup_mysql.sh
root@gsb-symfony:/var/www/html#
```

On crée le
crontab -e

```
no crontab for root - using an empty one
Select an editor.  To change later, run select-editor again.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny

Choose 1-2 [1]: 1
```

Appuyer sur 1 pour l'éditeur nano

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
```

A la fin de la ligne on ajoute les spécifités du cron (ici tous les vendredi à 11H)
0 11 * * 5 /usr/local/bin/backup_mysql.sh

```
# m h  dom mon dow   command
0 11 * * 5 /usr/local/bin/backup_mysql.sh
```

On ferme le fichier

Valider avec “Entrée”