

# EXPORTATION

# Variables mariadb

USER="nom"

PASSWORD="mdp"

DATABASE="nom de la base"

BACKUP\_DIR="lieu d'envoi temporaire du fichier"

DATE=\$(date +%Y-%m-%d\_%H-%M-%S)

mkdir -p "\$BACKUP\_DIR" : **créé le fichier si il n'ai pas créé**

BACKUP\_FILE="\$BACKUP\_DIR/\$DATABASE-\$DATE.sql" : **envoie dans le dossier temporaire et donne le nom à la sauvegarde**

mysqldump -u \$USER -p\$PASSWORD \$DATABASE > \$BACKUP\_FILE : **fais la sauvegarde de la BDD en se connectant sur mariadb**

openssl enc -aes-256-cbc -pbkdf2 -in \$BACKUP\_FILE -out \$BACKUP\_FILE\_CRYPT -pass pass:mdp : **commande pour chiffrer le fichier dans le dossier temp**

scp \$fichier\_chiffrer [user@ipServeur](#):chemin : **Permet d'envoyer le fichier chiffré au serveur distant**

rm \$fichier\_non-chiffrer : **Permet de supprimer le fichier non-chiffré stocké localement**

verif1=`md5sum \$fichier\_chiffrer` : **stock l'empreinte du fichier chiffré local**

ssh [user@ipServeur](#)<<EOF : **On se connecte au serveur a distance**

cd /var/www/html/scrum/saveDB/bash/save : **On se déplace dans le dossier où sont les backup**

if [ "\$verif1" = "`md5sum \$BACKUP\_FILE`" ]; then : **vérifie si les deux fichiers ont la même empreinte**

echo "C good"

else

echo "C pas good"

fi

EOF : **permet d'envoyer un script ou plusieurs commandes à un serveur distant en une seule exécution.**

rm \$fichier\_chiffrer : **Permet de supprimer le fichier chiffré stocké localement**

# IMPORTATION

ssh [user@ipServeur](#) <<'EOF' : **On se connecte au serveur a distance**

cd /var/www/html/scrum/saveDB/bash/save : **On se déplace dans le dossier où sont les backup**

file=\$(ls -t | head -n 1) : **Prend la dernière backup effectué**

nbBackup=\$(ls -t | head -n 1 | wc -l) : **Regarde si on a ou pas une backup**

if [ "\$nbBackup" -eq 0 ]; then

echo "Pas de backup disponible"

else

scp \$file [userClient@ipClient](#) :/var/www/html/scrum/scrum1/importation/ : **Récupère le fichier chiffré**

**du serveur a distance**

if [ \$? -eq 0 ]; then

echo "Importation réussi"

else

echo "Importation échoué"

fi

fi

EOF

inImportation=\$(ls -t | head -n 1 | wc -l) : **stocker le nombre de backup dans le dossier importation**

if [ "\$inImportation" != 0 ]; then : **si le nombre de backup n'est pas 0**

BACKUP\_FILE\_CRYPT=\$(ls -t importation/ | head -n 1) : **Prend le nom de la backup la plus récente**

openssl enc -d -aes-256-cbc -pbkdf2 -in importation/\$BACKUP\_FILE\_CRYPT -out \$BACKUP\_FILE -pass pass:matteocadinot05 : **Déchiffre la backup et crée la backup non-chiffré**

if [ \$? -eq 0 ]; then

rm importation/\$BACKUP\_FILE\_CRYPT : **Supprime la backup chiffré**

mysql -u \$USER -p\$PASSWORD \$DATABASE < \$BACKUP\_FILE : **Restaure la BDD avec**

**la backup**

echo "Restauration effectuée avec succès"

rm \$BACKUP\_FILE : **Supprime la backup non-chiffré localement**

else

echo "Erreur de déchiffrement"

fi

else

echo "Erreur d'importation"

fi

# CRON

Installation de cron

```
apt update && apt install cron -y
```

crontab -e : **fichier où il faut configurer la date, l'heure, la minute, où l'on souhaite exécuter une commande**

crontab -l : **Permet de voir la liste des commandes voulu**