

Procédure Complète : Sauvegarde Automatisée Sécurisée avec Chiffrement GPG et Transfert SSH

📖 Objectif :

Configurer un système de sauvegarde automatisé qui :

- ✓Sauvegarde les fichiers d'un serveur web et la base de données MySQL
- ✓Chiffre les sauvegardes avec GPG
- ✓Transfère les fichiers sur une machine distante via SSH sans mot de passe
- ✓Automatise le processus avec crontab

1. Configuration des Répertoires et Machines

📖 Serveur (192.168.27.10) :

- Héberge les fichiers web dans /var/www/mangaverse2.0
- Contient la base de données mangadb
- Stocke temporairement les sauvegardes dans /~/backup/
- Exécute les scripts backup_db.sh et backup_file.sh

📖 Machine Distante (192.168.27.12) :

- Reçoit les fichiers sauvegardés dans /~/backup_projets/
- Déchiffre les sauvegardes pour s'assurer de leur intégrité via un script

2. Configuration SSH Sans Mot de Passe

Sur le serveur, générer une clé SSH :

```
ssh-keygen -t rsa -b 4096
```

📖 Appuyer sur Entrée à chaque question (pas de passphrase)

```

administrateur@administrateur-VirtualBox:~$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/administrateur/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/administrateur/.ssh/id_rsa
Your public key has been saved in /home/administrateur/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:TL4jXr+u5GLZWJ2recTY6H3W0awH+xLDtq60SEUNv4 administrateur@administrateur-VirtualBox
The key's randomart image is:
+---[RSA 4096]-----+
|
|      +
|     o o
|    . o .
|   + = + o
|  So * E .
|   .o = X o|
|  . =o . B #|
|  . =o+  + X +|
|   ..+o+=o. *o|
+-----[SHA256]-----+

```

Copier la clé publique sur la machine distante :

```
ssh-copy-id osboxes@192.168.27.12
```

Tester la connexion SSH :

```
ssh osboxes@192.168.27.12
```

✓ Si pas de mot de passe demandé, la configuration est bonne.

3. Génération d'une Clé GPG pour le Chiffrement

Depuis la machine cliente (qui fera office de serveur gpg) :

Commande : `gpg --full-generate-key`

```

administrateur@administrateur-VirtualBox:~$ gpg --full-generate-key
gpg (GnuPG) 2.4.4; Copyright (C) 2024 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

```

☐ Options à choisir :

- Type : (1) RSA et RSA
- Taille : 4096 bits
- Expiration : 2 ans
- Nom : Backup Serveur
- Email : wasterans@gmail.com

Lister les clés pour obtenir l'ID : `gpg --list-keys`

```
administrateur@administrateur-VirtualBox:~$ gpg --list-keys
gpg: vérification de la base de confiance
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: profondeur : 0  valables : 1  signées : 0
      confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
/home/administrateur/.gnupg/pubring.kbx
-----
pub   rsa4096 2025-05-24 [SC]
      948BDD62615169158FEB964CAA0A81BAFAA6F34E
uid           [  ultime ] test (test) <test@test.com>
sub   rsa4096 2025-05-24 [E]
```

Exporter la clé publique :

```
gpg --export -a "wasterans@gmail.com" > gpg-public-key.asc
```

Transférer la clé publique :

```
scp gpg-public-key.asc osboxes@192.168.15.104:/home/osboxes/
```

Importer la clé sur la machine distante :

```
gpg --import /home/osboxes/gpg-public-key.asc
```

4. Script de Sauvegarde de la Base de Données (backup_db.sh)

Créer le script : `nano /home/osboxes/Documents/backup_db.sh`

Contenu du script :

```
#!/bin/bash

# Variables

DB_USER="adminmanga"

DB_PASSWORD="naruto"

DB_NAME="mangaverse"

BACKUP_DIR="/home/aftec/backup"

DATE=$(date +%Y-%m-%d)

GPG_KEY="wasterans@gmail.com"

ARCHIVE_NAME="mangaverse_backup_$(date +%Y-%m-%d).sql"
```

Créer le répertoire de sauvegarde s'il n'existe pas

```
mkdir -p $BACKUP_DIR
```

Effectuer la sauvegarde

```
mysqldump -u $DB_USER -p$DB_PASSWORD $DB_NAME > $BACKUP_DIR/$DB_NAME-$DATE.sql
```

#Chiffrer la sauvegarde

```
gpg --always-trust --output $BACKUP_DIR/$DB_NAME-$DATE.sql.gpg --encrypt --recipient $GPG_KEY $BACKUP_DIR/$DB_NAME-$DATE.sql
```

=== SUPPRIMER LE FICHIER NON CHIFFRÉ ===

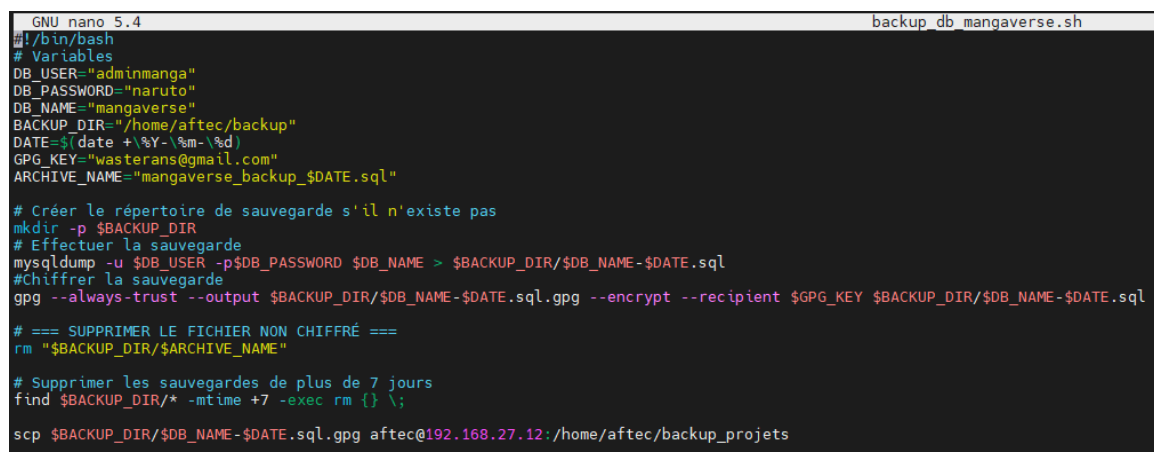
```
rm "$BACKUP_DIR/$ARCHIVE_NAME"
```

Supprimer les sauvegardes de plus de 7 jours

```
find $BACKUP_DIR/* -mtime +7 -exec rm {} \;
```

```
scp $BACKUP_DIR/$DB_NAME-$DATE.sql.gpg
```

aftec@192.168.27.12:/home/aftec/backup_projets



The screenshot shows a terminal window with the title 'GNU nano 5.4' and 'backup_db_mangaverse.sh'. The script content is as follows:

```
#!/bin/bash
# Variables
DB_USER="adminmanga"
DB_PASSWORD="naruto"
DB_NAME="mangaverse"
BACKUP_DIR="/home/aftec/backup"
DATE=$(date +%Y-%m-%d)
GPG_KEY="wasterans@gmail.com"
ARCHIVE_NAME="mangaverse_backup_$DATE.sql"

# Créer le répertoire de sauvegarde s'il n'existe pas
mkdir -p $BACKUP_DIR

# Effectuer la sauvegarde
mysqldump -u $DB_USER -p$DB_PASSWORD $DB_NAME > $BACKUP_DIR/$DB_NAME-$DATE.sql

#Chiffrer la sauvegarde
gpg --always-trust --output $BACKUP_DIR/$DB_NAME-$DATE.sql.gpg --encrypt --recipient $GPG_KEY $BACKUP_DIR/$DB_NAME-$DATE.sql

# === SUPPRIMER LE FICHIER NON CHIFFRÉ ===
rm "$BACKUP_DIR/$ARCHIVE_NAME"

# Supprimer les sauvegardes de plus de 7 jours
find $BACKUP_DIR/* -mtime +7 -exec rm {} \;

scp $BACKUP_DIR/$DB_NAME-$DATE.sql.gpg aftec@192.168.27.12:/home/aftec/backup_projets
```

Rendre le script exécutable : `chmod +x ~/scripts/backup_db.sh`

5. Script de Sauvegarde des Fichiers (backup_file.sh)

Créer le script : nano ~/scripts/backup_file.sh

Contenu du script :

```
#!/bin/bash
```

```
# === VARIABLES ===
```

```
SOURCE_DIR="/var/www/mangaverse2.0"
```

```
BACKUP_DIR="/home/aftec/backup"
```

```
DATE=$(date +%Y-%m-%d')
```

```
ARCHIVE_NAME="mangaverse_backup_$(DATE).tar.gz"
```

```
ENCRYPTED_ARCHIVE="$ARCHIVE_NAME.gpg"
```

```
GPG_KEY="wasterans@gmail.com"
```

```
DEST_USER="aftec"
```

```
DEST_IP="192.168.27.12"
```

```
DEST_DIR="/home/aftec/backup_projets"
```

```
# === CRÉER LE DOSSIER DE BACKUP SI NÉCESSAIRE ===
```

```
mkdir -p "$BACKUP_DIR"
```

```
# === SAUVEGARDE : COMPRESSION ===
```

```
tar -czf "$BACKUP_DIR/$ARCHIVE_NAME" -C "$SOURCE_DIR" .
```

```
# === CHIFFREMENT ===
```

```
gpg --yes --always-trust --output "$BACKUP_DIR/$ENCRYPTED_ARCHIVE" --encrypt --recipient "$GPG_KEY" "$BACKUP_DIR/$ARCHIVE_NAME"
```

```
# === SUPPRIMER LE FICHIER NON CHIFFRÉ ===
```

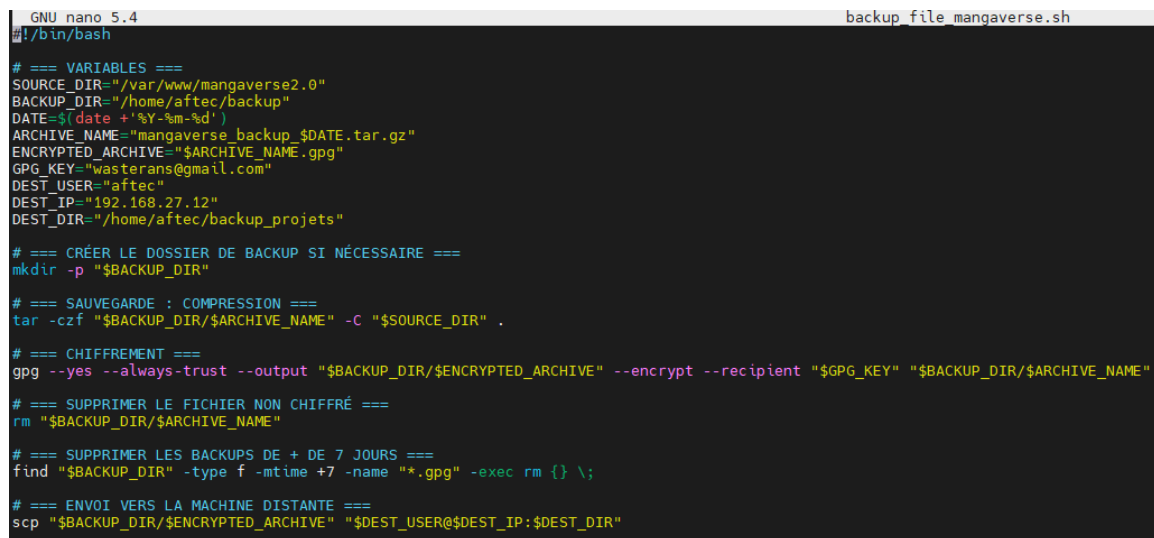
```
rm "$BACKUP_DIR/$ARCHIVE_NAME"
```

```
# === SUPPRIMER LES BACKUPS DE + DE 7 JOURS ===
```

```
find "$BACKUP_DIR" -type f -mtime +7 -name "*.gpg" -exec rm {} \;
```

```
# === ENVOI VERS LA MACHINE DISTANTE ===
```

```
scp "$BACKUP_DIR/$ENCRYPTED_ARCHIVE" "$DEST_USER@$DEST_IP:$DEST_DIR"
```



```
GNU nano 5.4 backup_file_mangaverse.sh
#!/bin/bash

# === VARIABLES ===
SOURCE_DIR="/var/www/mangaverse2.0"
BACKUP_DIR="/home/aftec/backup"
DATE=$(date +%Y-%m-%d)
ARCHIVE_NAME="mangaverse_backup_$DATE.tar.gz"
ENCRYPTED_ARCHIVE="$ARCHIVE_NAME.gpg"
GPG_KEY="wasterans@gmail.com"
DEST_USER="aftec"
DEST_IP="192.168.27.12"
DEST_DIR="/home/aftec/backup_projets"

# === CRÉER LE DOSSIER DE BACKUP SI NÉCESSAIRE ===
mkdir -p "$BACKUP_DIR"

# === SAUVEGARDE : COMPRESSION ===
tar -czf "$BACKUP_DIR/$ARCHIVE_NAME" -C "$SOURCE_DIR" .

# === CHIFFREMENT ===
gpg --yes --always-trust --output "$BACKUP_DIR/$ENCRYPTED_ARCHIVE" --encrypt --recipient "$GPG_KEY" "$BACKUP_DIR/$ARCHIVE_NAME"

# === SUPPRIMER LE FICHIER NON CHIFFRÉ ===
rm "$BACKUP_DIR/$ARCHIVE_NAME"

# === SUPPRIMER LES BACKUPS DE + DE 7 JOURS ===
find "$BACKUP_DIR" -type f -mtime +7 -name "*.gpg" -exec rm {} \;

# === ENVOI VERS LA MACHINE DISTANTE ===
scp "$BACKUP_DIR/$ENCRYPTED_ARCHIVE" "$DEST_USER@$DEST_IP:$DEST_DIR"
```

Rendre le script exécutable : `chmod +x ~/scripts/backup_file.sh`

6. Tester la Sauvegarde et le Transfert

Exécuter les scripts :

```
/~/scripts/backup_db.sh
```

```
/~/scripts/backup_file.sh
```

Vérifier les fichiers reçus sur la machine distante :

```
ls -l ~/~/backup_file/
```

Tester le déchiffrement :

```
gpg --output ~/~/backup_file/html-backup.tar.gz --decrypt
/home/osboxes/backup_file/html-backup-XXXX.tar.gz.gpg
```

7. Automatiser avec crontab

Ouvrir crontab : `crontab -e`

Ajouter les lignes suivantes :

```
0 2 * * * /home/osboxes/Documents/backup_db.sh
```

```
5 2 * * * /home/osboxes/Documents/backup_file.sh
```

✓ Les sauvegardes seront exécutées automatiquement tous les jours à 2h avec un intervalle de 5 minutes pour éviter les surcharges

```
GNU nano 5.4
# 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
0 2 * * * /home/aftec/scripts/backup_db_mangaverse.sh
5 2 * * * /home/aftec/scripts/backup_file_mangaverse.sh
10 2 * * * /home/aftec/scripts/backup_db_easyroom.sh
15 2 * * * /home/aftec/scripts/backup_file_easyroom.sh
```