DOCUMENTATION TECHNIQUE

Mise en place d'un serveur de sauvegarde

rsync over SSH sur Debian

Rédacteur	Thomas
Système cible	Debian (VM)
Outil utilisé	rsync + SSH + cron
Utilisateur backup	backupuser1

Sommaire

1. Présentation et objectifs

Cette documentation décrit la mise en place d'un serveur de sauvegarde centralisé sous Debian. Les sauvegardes sont réalisées via rsync over SSH, ce qui garantit la confidentialité et l'intégrité des données transférées.

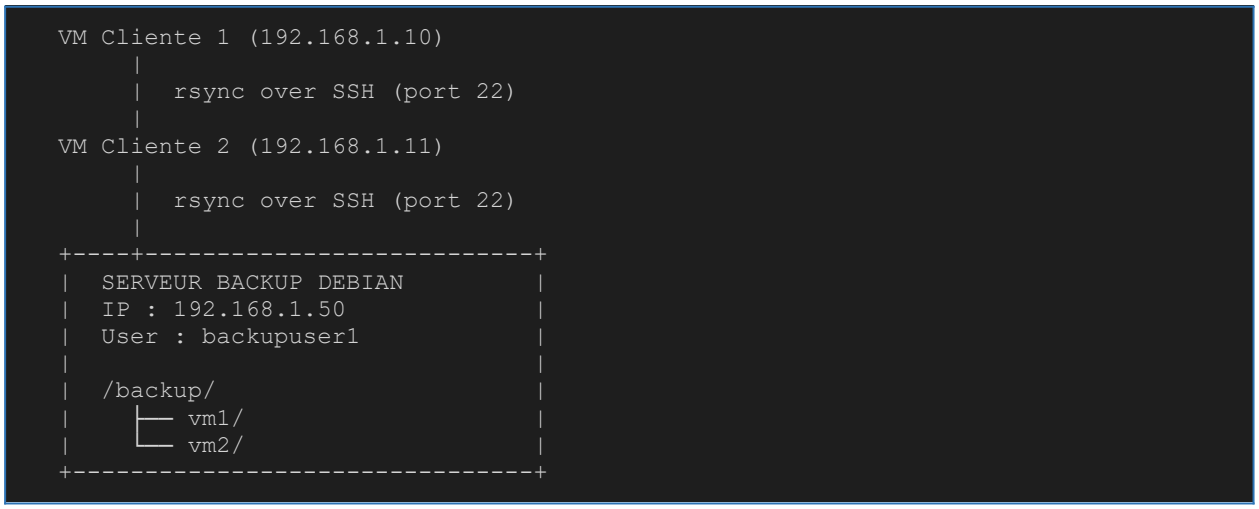
Objectifs

- Centraliser les sauvegardes de plusieurs VMs sur un serveur dédié
- Automatiser les sauvegardes via cron
- Sécuriser les transferts via SSH avec authentification par clé
- Respecter la règle des 3-2-1 (3 copies, 2 supports, 1 hors site)

> *Bonne pratique : la règle 3-2-1 recommande de conserver 3 copies des données, sur 2 supports différents, dont 1 copie hors site.*

2. Architecture réseau

Le schéma suivant décrit l'architecture mise en place :



3. Prérequis

Élément	Détail
Serveur backup	VM Debian 11/12, accès root
VMs clientes	Debian ou toute distribution Linux
Réseau	Connectivité IP entre les VMs
Paquets requis	rsync, openssh-server, openssh-client
Espace disque	Suffisant pour stocker les backups

4. Configuration du serveur de backup

4.1 Mise à jour du système

```
sudo apt update && sudo apt upgrade -y
```

4.2 Installation des paquets

```
sudo apt install rsync openssh-server -y

# Vérifier que SSH est actif
sudo systemctl status ssh
sudo systemctl enable ssh
```

4.3 Création de l'utilisateur backupuser1

Un utilisateur dédié est créé pour isoler les accès de sauvegarde des autres comptes du système.

```
# Créer l'utilisateur backupuser1
sudo useradd -m -s /bin/bash backupuser1
sudo passwd backupuser1

# Créer le répertoire .ssh
sudo mkdir -p /home/backupuser1/.ssh
sudo chown backupuser1:backupuser1 /home/backupuser1/.ssh
sudo chmod 700 /home/backupuser1/.ssh
```

4.4 Création des répertoires de sauvegarde

```
# Créer les dossiers de backup
sudo mkdir -p /backup/vm1
sudo mkdir -p /backup/vm2

# Donner les droits à backupuser1
sudo chown -R backupuser1:backupuser1 /backup
sudo chmod -R 750 /backup

# Vérifier
ls -lh /backup
```

> *Attention : le répertoire /backup doit avoir suffisamment d'espace disque. Vérifiez avec : df -h /backup*

5. Configuration des clients

5.1 Installation de rsync côté client

```
sudo apt update
sudo apt install rsync openssh-client -y
```

5.2 Génération de la paire de clés SSH

L'authentification par clé SSH évite d'avoir à saisir un mot de passe lors des sauvegardes automatiques.

```
# Générer une paire de clés dédiée aux backups
ssh-keygen -t rsa -b 4096 -f ~/.ssh/backup_key -N ""

# Résultat :
# ~/.ssh/backup_key       -> clé privée (à garder secrète)
# ~/.ssh/backup_key.pub   -> clé publique (à envoyer au serveur)
```

5.3 Envoi de la clé publique au serveur backup

```
# Envoyer la clé publique vers backupuser1 sur le serveur
ssh-copy-id -i ~/.ssh/backup_key.pub backupuser1@192.168.1.50

# Tester la connexion (sans mot de passe)
ssh -i ~/.ssh/backup_key backupuser1@192.168.1.50
```

> *Si ssh-copy-id n'est pas disponible, copier manuellement le contenu de backup_key.pub dans /home/backupuser1/.ssh/authorized_keys sur le serveur.*

6. Script de sauvegarde automatique

Le script suivant est à créer sur chaque VM cliente. Il utilise rsync pour synchroniser les données vers le serveur backup.

6.1 Création du script

```
sudo nano /usr/local/bin/backup.sh
```

Contenu du script :

```
#!/bin/bash

# ============================================
#  SCRIPT DE SAUVEGARDE - rsync over SSH
#  Utilisateur : backupuser1
# ============================================

# ---- CONFIGURATION ----
SERVEUR="192.168.1.50"
USER="backupuser1"
CLE="/root/.ssh/backup_key"
SOURCE="/"
DESTINATION="/backup/vm1"
LOG="/var/log/backup.log"
DATE=$(date '+%Y-%m-%d %H:%M:%S')

# ---- DOSSIERS À EXCLURE ----
EXCLUSIONS="--exclude=/proc \
            --exclude=/sys \
            --exclude=/dev \
            --exclude=/tmp \
            --exclude=/run \
            --exclude=/mnt \
            --exclude=/media \
            --exclude=/lost+found"

# ---- LANCEMENT DE LA SAUVEGARDE ----
echo "[$DATE] === Debut de la sauvegarde ===" >> $LOG

rsync -avz --delete \
      -e "ssh -i $CLE -o StrictHostKeyChecking=no" \
      $EXCLUSIONS \
      $SOURCE \
      $USER@$SERVEUR:$DESTINATION >> $LOG 2>&1

CODE=$?
echo "[$DATE] Fin - Code retour : $CODE" >> $LOG

if [ $CODE -eq 0 ]; then
    echo "[$DATE] SUCCES" >> $LOG
else
    echo "[$DATE] ECHEC - Vérifier le log" >> $LOG
fi
```

6.2 Rendre le script exécutable

```
sudo chmod +x /usr/local/bin/backup.sh

# Tester manuellement
sudo /usr/local/bin/backup.sh

# Vérifier le log
cat /var/log/backup.log
```

7. Automatisation avec cron

Cron permet de planifier les sauvegardes automatiquement à intervalles réguliers.

```
# Éditer la crontab de root
sudo crontab -e
```

Exemples de planifications :

```
# Tous les jours à 2h00 du matin
0 2 * * * /usr/local/bin/backup.sh

# Tous les dimanches à 3h00
0 3 * * 0 /usr/local/bin/backup.sh

# Toutes les 6 heures
0 */6 * * * /usr/local/bin/backup.sh
```

> *Syntaxe cron : minute heure jour_du_mois mois jour_de_semaine commande*

8. Vérification et tests

8.1 Vérifier les fichiers reçus sur le serveur

```
# Sur le serveur backup
ls -lh /backup/vm1/

# Espace utilisé
du -sh /backup/vm1/

# Espace disque disponible
df -h /backup
```

8.2 Vérifier les logs

```
# Afficher le log de sauvegarde
cat /var/log/backup.log

# Suivre le log en temps réel
tail -f /var/log/backup.log

# Vérifier les tâches cron exécutées
grep CRON /var/log/syslog | tail -20
```

8.3 Tester une restauration

```
# Restaurer un fichier depuis le serveur backup
rsync -avz \
      -e "ssh -i ~/.ssh/backup_key" \
      backupuser1@192.168.1.50:/backup/vm1/etc/hostname \
      /tmp/restore_test/
```

> *Important : tester régulièrement la restauration. Un backup non testé n'est pas fiable.*

9. Gestion de la rétention

Pour conserver plusieurs versions de backup horodatées, modifier le script pour créer un dossier par date.

```
# Dans backup.sh, remplacer DESTINATION par :
DESTINATION="/backup/vm1/$(date +%Y-%m-%d)"

# Créer le dossier automatiquement
ssh -i $CLE backupuser1@$SERVEUR "mkdir -p $DESTINATION"
```

Pour supprimer automatiquement les backups de plus de 30 jours sur le serveur :

```
# Sur le serveur backup, ajouter dans crontab :
0 4 * * * find /backup/vm1/ -maxdepth 1 -type d -mtime +30 -exec rm -rf {} \;
```

10. Options rsync de référence

Option	Description
-a	Mode archive : conserve permissions, dates, liens symboliques
-v	Verbose : affiche les fichiers transférés
-z	Compression des données pendant le transfert
--delete	Supprime côté destination les fichiers supprimés côté source
--progress	Affiche la progression en temps réel
--dry-run	Simulation sans modification réelle (test)
-e ssh	Utilise SSH comme tunnel de transport
--exclude	Exclut des fichiers ou dossiers de la sauvegarde
--bwlimit	Limite la bande passante utilisée (ex: --bwlimit=1000 = 1Mo/s)

> *Documentation officielle rsync : man rsync ou rsync --help*