

Rsync

Last updated by | Cerena Hostains | 11 juin 2025 at 08:58 UTC+2

Sauvegarde avec Rsync

Introduction

Rsync (Remote Synchronization) est un logiciel de synchronisation de fichiers.

Rsync utilise SSH pour le transfert des fichiers entre serveur local et un serveur distant.

Ici, il sera utilisé pour mettre en place un système de sauvegarde distante.

Pré-requis

1. Serveur de backup : pour stocké les sauvegardes.
2. Les différents serveurs qui hébergent les données à sauvegarder.

Installation de Rsync

- Installer Rsync sur votre serveur de backup et sur les serveurs qui auront besoin de sauvegarder leur fichiers ou base de données.

```
apt install rsync
```

- Vérifier l'installation

```
rsync --version
```

Cela affichera la version de Rsync installée sur la machine.

Configuration de la sauvegarde distante

Nous allons configurer une tâche qui copie les fichiers d'un serveur où sont héberger les données à sauvegarder vers le serveur de sauvegarde via SSH.

1. Générer une clé SSH

Générer une paire de clé SSH va permettre de faciliter la connexion SSH sans mot de passe.

Sur le serveur où sont héberger les données à sauvegarder, exécutez la commande suivante pour générer une clé SSH:

```
ssh-keygen -t ed25519
```

Cela va créer une clé publique et privée dans le répertoire ~/.ssh/ par défaut.

2. Copier la clé publique sur le serveur de sauvegarde

Dès que la clé est générée, vous pouvez ajouter la clé publique au fichier des clés autorisées du serveur de sauvegarde pour permettre une connexion SSH sans mot de passe.

Sur un des serveurs de production, exécutez la commande suivante (bien évidemment en remplaçant `user@backup_server` par l'utilisateur et l'adresse de votre serveur de sauvegarde).

```
ssh-copy-id user@backup_server
```

Cette commande va ajouter la clé publique au fichier `~/.ssh/authorized_keys` du serveur de backup.

3. Tester la connexion SSH

Vérifiez si vous pouvez vous connecter au serveur de sauvegarde sans mot de passe en exécutant :

```
ssh user@backup_server
```

Utilisation de Rsync pour les sauvegardes

Il existe plusieurs façon de sauvegarder:

1. Sauvegarde manuelle

```
rsync -avz /chemin/du/dossier/a/sauvegarder/ user@backup_server:/chemin/du/dossier/de/sauvegarde/
```

2. Sauvegarde avec suppression des fichiers obsolètes

Afin de supprimer les fichiers du serveur de sauvegarde qui n'existent plus sur le serveur de production, utilisez l'option `--delete` :

```
rsync -avz --delete /chemin/du/dossier/a/sauvegarder/ user@backup_server:/chemin/du/dossier/de/sauvegarde/
```

Cette commande supprimera les fichiers qui ont été supprimés du répertoire source.

3. Sauvegarde avec un script bash d'automatisation

Pour sauvegarder nos fichiers ou base de données importantes, nous pouvons créer un script bash sur toutes les machines où il faut sauvegarder les données en question.



```
#!/bin/bash

# Répertoire local de sauvegarde
BACKUP_DIR="/var/backups"

# Détails du serveur distant
REMOTE_SERVER="user@192.168.50.10"
REMOTE_DIR="/home/user/backup/zabbix"

# Nom de la sauvegarde avec date
BACKUP_NAME="zabbix.tar.gz"

# Fichier Zabbix à sauvegarder
ZABBIX_CONF="/etc/zabbix/zabbix_server.conf"

# Créer le répertoire local s'il n'existe pas
mkdir -p "$BACKUP_DIR"

echo "[+] Sauvegarde de Zabbix..."
tar -czf "${BACKUP_DIR}/${BACKUP_NAME}" "$ZABBIX_CONF"

if [ $? -ne 0 ]; then
    echo "[!] Échec de la création de l'archive"
    exit 1
fi

echo "[+] Envoi au serveur distant..."
rsync -avz --delete "${BACKUP_DIR}/${BACKUP_NAME}" "$REMOTE_SERVER:$REMOTE_DIR"

if [ $? -ne 0 ]; then
    echo "[!] Échec de l'envoi vers le serveur distant"
    exit 1
fi

echo "[+] Suppression des anciennes sauvegardes locales (plus de 1 jour)..."
find "$BACKUP_DIR" -type f -name "zabbix_*.tar.gz" -mtime +1 -delete

echo "[✓] Sauvegarde terminée avec succès."
```

Ce script peut être exécuter manuellement avec la commande suivante: `bash backup_zabbix.sh`.

Ou il peut être également lancer automatiquement grâce à la crontab :



```
00 20 * * * /root/backup_zabbix.sh >> /var/log/backup.log 2>&1
```

Ici, le script se lance tout les soirs à 21h, il exécute le chemin où se situe le script bash et va mettre les logs d'exécution dans `/var/logs/backup.log`.

`>> /var/log/backup.log 2>&1` → Cela enregistre tout ce que fait le script (y compris les erreurs) dans le fichier `/var/log/backup.log`.

4. Sauvegarde depuis la crontab

Nous pouvons ajouter directement l'environnement complet dans la crontab:



```
00 20 * * * rsync -avz --delete /chemin/du/dossier/a/sauvegarder/ user@backup_server:/chemin/du/dossier/de/
```



Ça garantit que cron trouve bien rsync même s'il s'exécute dans un environnement minimal.