

BorgBackup // Server <- - rsync <- - VPS

Le Saint Graal de la sécurité des backups.....	2
L'ARME : BORGBACKUP (Présentation).....	2
PRÉPARATION DU TERRAIN (Installation & SSH).....	3
Installation (Sur le VPS ET le server de backups).....	3
Installation sur le server de backups.....	3
La Clé des Portes (SSH sans mot de passe).....	4
Sur le server de backups (Génération des clés).....	4
L'Infiltration (Copie de la clé vers Pollux).....	4
Le Test (Obligatoire).....	4
L'INITIALISATION (Le "Big Bang").....	5
Conflit ou pas conflit si votre disque est déjà chiffré ? (La Double Couche).....	5
L'Architecture (Le Concept).....	6
Le Script Maître.....	7
Lancement manuel du script.....	8
Pourquoi ça se connecte X fois ?.....	8
L'Automatisation (Le Cron).....	8

Le Saint Graal de la sécurité des backups

- **Le Push (Mauvais)** : Le VPS envoie ses backups vers un server de backups. Si le VPS est compromis (hacké), l'attaquant a les clés pour détruire aussi les backups sur le server de backups.
- **Le Pull (Bon)** : le server de backups se réveille, se connecte au VPS, aspire les données et se referme. Le VPS ne connaît même pas l'existence du server de backups. Si le VPS brûle, Le server de backups regarde le feu de loin en mangeant du popcorn.

Référence Culturelle : Le server de backups agit comme *Ethan Hunt* dans *Mission Impossible*. Il s'infiltré dans le VPS (la chambre forte), vole les données (les backups) et s'exfiltre sans laisser de traces.

L'ARME : BORGBACKUP (Présentation)

Oubliez cp ou rsync classique. BorgBackup (Borg pour les intimes) n'est pas un simple copieur de fichiers. C'est une machine à voyager dans le temps.

- **La Déduplication (Magie Noire)** : Borg découpe vos fichiers en petits morceaux (chunks). Si un morceau existe déjà dans le backup (même dans un autre fichier !), il ne le recopie pas.
 - *Résultat* : Sauvegarder 10 fois un disque de 50 Go ne prend pas 500 Go, mais environ 52 Go. C'est de la compression spatio-temporelle digne du TARDIS.
- **Le Chiffrement (AES-256)** : Tout est chiffré **côté client** avant d'être écrit sur le disque. Même si on vole votre disque dur, sans la phrase de passe, vos données sont illisibles.
- **La Compression** : Il compresse les données à la volée (LZ4, Zstd).

PRÉPARATION DU TERRAIN (Installation & SSH)

Nous devons installer Borg sur les deux machines (L'agent infiltré sur le VPS, et le Cerveau sur le server de backups).

Installation (Sur le VPS ET le server de backups)

```
sudo apt update && sudo apt install rsync
```

Installation sur le server de backups

```
sudo apt update && sudo apt install borgbackup
```

La Clé des Portes (SSH sans mot de passe)

Le server de backups doit pouvoir entrer dans le VPS sans taper de mot de passe à 3h du matin.

Sur le server de backups (Génération des clés)

Si vous n'avez pas déjà de clé SSH

```
ssh-keygen -t ed25519 -f .ssh/vps_ls backupsKey -C "vps_serverBackups"
```

Appuyez sur Entrée pour tout (pas de passphrase pour cette clé spécifique d'automatisation, sinon le script bloquera)

L'Infiltration (Copie de la clé vers Pollux)

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub user@ip_vps
```

Remplacez 'user' et 'ip_pollux' par les infos réelles

Le Test (Obligatoire)

Depuis le server de backups, tentez :

```
ssh user@ip_vps
```

Succès : Vous êtes connecté sans mot de passe. "I'm in."

Échec : On arrête tout et on debug.

L'INITIALISATION (Le "Big Bang")

Nous allons créer le coffre-fort (Repository) sur **Le server de Backups**. C'est ici que seront stockées les archives.

Création du dossier de stockage

```
mkdir -p /mnt/storage/backups/vps
```

Initialisation du Repo Borg (Chiffré)

```
borg init --encryption=repokey /mnt/storage/backups/vps
```

⚠ **AVERTISSEMENT CRITIQUE (Niveau DEFCON 1)** ⚠ Borg va vous demander une **Passphrase**. **NE LA PERDEZ JAMAIS**. Si vous perdez cette phrase, vos backups sont morts. Finis. *Excommunicado*. Il n'y a pas de "Mot de passe oublié". Notez-la dans votre gestionnaire de mots de passe (Bitwarden/Keepass) **IMMÉDIATEMENT**.

Conflit ou pas conflit si votre disque est déjà chiffré ? (La Double Couche)

Réponse courte : Aucun conflit.

Réponse technique : C'est le principe de l'oignon (ou de *Shrek*). Les couches sont indépendantes.

- **Couche 1 (LVM/LUKS)** : C'est le niveau **Matériel/OS**. Il chiffre le *conteneur* (le disque dur physique). Si quelqu'un vole le serveur de backups éteint, il ne peut rien lire. Une fois le serveur de backups allumé et la clé entrée, les données sont "en clair" pour le système d'exploitation.
- **Couche 2 (Borg)** : C'est le niveau **Applicatif/Fichier**. Il chiffre le *contenu* (les archives).

Analogie : LVM, c'est le coffre-fort de la banque (le bâtiment). Borg, c'est une valise verrouillée à *l'intérieur* du coffre-fort. Avoir une valise fermée dans un coffre fermé n'est pas un problème. C'est de la "Défense en Profondeur".

L'Architecture (Le Concept)

Nous appliquons la stratégie du "**Miroir Froid**". Contrairement à une sauvegarde directe, nous procédons en deux temps pour garantir l'intégrité des données et la résilience face aux coupures réseau.

1. **Phase 1 (Le Miroir - Rsync)** : Le serveur de backup se connecte au VPS et synchronise une copie exacte ("miroir") des fichiers critiques vers un dossier temporaire local.
 - *Avantage* : Si la connexion coupe, Rsync reprend là où il s'est arrêté.

2. **Phase 2 (L'Archivage - Borg)** : Atlas archive ce dossier miroir local dans le coffre-fort Borg.
 - *Avantage* : Borg travaille en local (rapide) et ne dépend plus des aléas du réseau pour la déduplication.

Le Script Maître

Ce script doit être placé sur le **server de backup** (ex: /usr/local/bin/backupBorg-vps.sh). Il gère la connexion, le miroir, l'archivage et le nettoyage.

```
#!/bin/bash

# =====
#  PROTOCOL: MIRROR & BACKUP (SERVER <- Borg <- Rsync <- VPS)
#  =====

# --- CONFIGURATION SSH ---
REMOTE_USER="user"
REMOTE_HOST="ip_address"
SSH_PORT="22"
# La clé SSH dédiée à l'automatisation (sans passphrase ou gérée par agent)
SSH_KEY="/home/user/.ssh/id_ed25519"

REMOTE_SSH="$REMOTE_USER@$REMOTE_HOST"

# --- CIBLES SUR LE VPS ---
# Liste des dossiers critiques à sauvegarder
DIRS_TO_BACKUP="/opt/docker /etc/ssh /etc/fail2ban /etc/ufw ..."

# --- CONFIGURATION LOCALE ---
MIRROR_DIR="/mnt/backups/vps/mirror"
REPOSITORY="/mnt/backups/vps"
ARCHIVE_NAME="vps-$(date +%Y-%m-%d_%H-%M)"

# Passphrase du Repo Borg
export BORG_PASSPHRASE='Votre_Passphrase_Indevinable'

# --- SYNCHRONISATION DU MIROIR (RSYNC) ---
echo "[$(date)] 1. Synchronisation Rsync..."
mkdir -p $MIRROR_DIR

# Boucle de traitement par dossier
for dir in $DIRS_TO_BACKUP; do
    echo "[SYNC] Traitement de : $dir"

    rsync -avz --delete --numeric-ids -R \
        --rsync-path="sudo rsync" \
        -e "ssh -p $SSH_PORT -i $SSH_KEY -o StrictHostKeyChecking=no" \
        $REMOTE_SSH:$dir \
        $MIRROR_DIR
done

# --- ARCHIVAGE BORG ---
echo "[$(date)] 2. Archivage Borg..."

# Sécurité : On ne sauvegarde pas un miroir vide
if [ -z "$(ls -A $MIRROR_DIR)" ]; then
    echo "[CRITIQUE] Le miroir est vide ! Arrêt d'urgence."
    exit 1
fi

/usr/bin/borg create --stats --compression lz4 --exclude-caches \
    $REPOSITORY::$ARCHIVE_NAME \
    $MIRROR_DIR

BORG_EXIT=$?

# --- NETTOYAGE (RETENTION) ---
if [ ${BORG_EXIT} -eq 0 ]; then
    echo "[SUCCESS] Backup terminé."
    # Politique : 7 jours, 4 semaines, 6 mois
    /usr/bin/borg prune --keep-daily=7 --keep-weekly=4 --keep-monthly=6 $REPOSITORY
else
    echo "[FAILURE] Erreur Borg."
fi

exit ${BORG_EXIT}
```

Lancement manuel du script

```
/usr/local/bin/backupBorg-vps.sh
```

Pourquoi ça se connecte X fois ?

Le script lance une commande rsync distincte pour chaque dossier (/opt/docker, puis /etc/ssh, etc.).

- **L'Avantage** : Si Rsync plante sur /etc/fail2ban (fichier verrouillé, erreur disque...), le script **continue** et sauvegarde quand même /opt/docker.
- **L'Inconvénient** : On perd 2 secondes à refaire la "poignée de main" SSH à chaque fois.
-

Pour un backup nocturne, on préfère la **fiabilité** à la vitesse pure. On s'en fiche que ça prenne 10 secondes de plus tant que ça ne plante pas.

L'Automatisation (Le Cron)

Une sauvegarde manuelle est une sauvegarde qu'on oublie. L'automatisation est obligatoire.

```
sudo crontab -e
```

```
# lancement tout les jours a 03h30 !
```

```
30 03 * * * root /usr/local/bin/backupBorg-vps.sh >> /var/log/backup_vps.log 2>&1
```

```
# 30 03 * * * : À 03h30, tous les jours.
```

```
# >> /var/log/backup_pollux.log : On écrit le résultat (les "OK", "SUCCESS") dans un fichier journal.
```

```
# 2>&1 : On écrit aussi les erreurs dans ce même journal.
```

La "Magie Borg" (Borg Mount)

C'est la méthode la plus élégante. Au lieu de regarder le miroir (qui est un dossier de travail brut), on regarde l'**archive Borg** finale. Borg possède une fonction géniale qui permet de monter une archive comme un disque dur virtuel.

Créer un point de montage

```
mkdir /tmp/backups-view
```

Monter le dépôt Borg (Il demandera la passphrase)

```
borg mount /mnt/archives/backups/server /tmp/backups-view
```

Naviguer dedans

```
cd /tmp/backups-view  
ls -la
```

On verra toutes tes dates de backup comme des dossiers !

On peut entrer dedans et copier des fichiers.

Quand on a fini (TRÈS IMPORTANT)

```
cd ~  
borg umount /tmp/backups-view
```