

Rapport du Projet Master 1 ASR BORGBACKUP

DIALLO Boubacar

Encadrant: Monsieur Pascal PETIT

I) maquette

Station1 : 192.168.10.2

Serveur2 : 192.168.10.1

II) installation de Borgbackup sur les deux machines

sudo apt update && sudo apt install borgbackup -y

```
--remote-path PATH      use PATH as borg executable on the remote (default: "borg")
--remote-ratelimit RATE deprecated, use '--upload-ratelimit' instead
--upload-ratelimit RATE set network upload rate limit in kiByte/s (default: 0=unlimited)
--remote-buffer UPLOAD_BUFFER deprecated, use '--upload-buffer' instead
--upload-buffer UPLOAD_BUFFER set network upload buffer size in MiB. (default: 0=no buffer)
--consider-part-files   treat part files like normal files (e.g. to list/extract them)
--debug-profile FILE    Write execution profile in Borg format into FILE. For local use a Python-compatible file can be generated by suffixing FILE with ".pyprof".
--rsh RSH               Use this command to connect to the 'borg serve' process (default: 'ssh')
```

required arguments:

<command>	
benchmark	benchmark command
break-lock	break repository and cache locks
check	verify repository
compact	compact segment files / free space in repo
config	get and set configuration values
create	create backup
debug	debugging command (not intended for normal use)
delete	delete archive
diff	find differences in archive contents
export-tar	create tarball from archive
extract	extract archive contents
info	show repository or archive information
init	initialize empty repository
key	manage repository key
list	list archive or repository contents
mount	mount repository
prune	prune archives
recreate	Re-create archives
rename	rename archive
serve	start repository server process
umount	umount repository
upgrade	upgrade repository format
with-lock	run user command with lock held
import-tar	Create a backup archive from a tarball

III) Configuration du serveur de sauvegarde (Serveur2)

* Sur Serveur2, nous allons :

Créer un utilisateur dédié à la sauvegarde pour éviter d'utiliser root.

sudo adduser --disabled-password --gecos "" borg

* Créer un dossier de stockage des sauvegardes :

```
sudo mkdir -p /backup/borg
```

```
sudo chown borg:borg /backup/borg
```

* Configurer SSH pour l'accès sécurisé depuis Station1

éditer le fichier `/etc/ssh/sshd_config`

```
GNU nano 7.2 /et
# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
```

Nous allons créer le Dossier pour stocker la clé de connexion en suite changé les permissions et de propriétaire

```
sudo mkdir -p /home/borg/.ssh
```

```
sudo touch /home/borg/.ssh/authorized_keys
```

```
sudo chmod 700 /home/borg/.ssh
```

```
sudo chmod 600 /home/borg/.ssh/authorized_keys
```

```
sudo chown -R borg:borg /home/borg/.ssh
```

Pour permettre que la connexion par clé

Configuration du client (Station1)

Sur Station1, nous allons :

Créer une paire de clés SSH : une clé privée et une autre public

```
ssh-keygen -t ed25519 -C "borg-backup"
```

```
root@debian:~# ls -l ~/.ssh/
total 16
-rw----- 1 root root 399  8 févr. 11:29 id_ed25519
-rw-r--r-- 1 root root  93  8 févr. 11:29 id_ed25519.pub
-rw-r--r-- 1 root root  78  8 févr. 11:53 known_hosts
```

Nous allons copier l'id de la clé publique dans le dossier authorized_keys

copier la cle dans le serveur ~/.ssh/authorized_keys en mode append-only pour eviter d'autre manipulation

```
borg@debian:~# nano /home/borg/.ssh/authorized_keys
GNU nano 7.2 /home/borg/.ssh/authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEMs4q7YgenoSgSwBkMUFzUPTuIwdp91x1R+kRXuXck+ root@debian
command="borg serve --append-only", ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIEMs4q7YgenoSgSwBkMUFzUPTuIwdp91x1R+kRXuXck+ root@debian
```

Tester la connexion SSH

```
ssh borg@192.168.10.1
```

```
root@debian:~# ssh borg@192.168.10.1
Linux debian 6.1.0-30-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86_64
```

```
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Sat Feb  8 11:53:00 2025 from 192.168.10.2
```

```
borg@debian:~$ █
```

Problème de Sécurité à partir de station j'ai acces à mon dossier de sauvegarde ce qui vraiment pose un souci de securité

Une Solution

a) Désactiver le shell pour l'utilisateur borg

b) paramétré mon repository en mode open-only

sudo usermod --shell /usr/sbin/nologin borg

```
root@debian:~# ssh borg@192.168.10.1 "ls /"  
This account is currently not available.  
root@debian:~#
```

Pour Reactivé le shell

sudo usermod --shell /bin/bash borg

```
borg@debian:/backup/borg$ cd ..
borg@debian:/backup$ cd ..
borg@debian:/$ exit
déconnexion
Connection to 192.168.10.1 closed.
root@debian:~# ssh borg@192.168.1.20 "ls /"
ssh: connect to host 192.168.1.20 port 22: Network is unreachable
root@debian:~# ssh borg@192.168.10.1 "ls /"
backup
bin
boot
dev
etc
home
initrd.img
initrd.img.old
lib
lib64
lost+found
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
Vbox
vmlinuz
vmlinuz.old
root@debian:~# █
```

j'arrive à supprimer la sauvegarde :

```
root@debian:~# borg list borg@192.168.10.3:/backup/borg_repo
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
documents-2025-02-15_13:12      Sat, 2025-02-15 13:12:32 [6d586d3d4a732e2d51c03a5f94ecd53f498822e7c74505dcf8b2f872e779f4c3]
documents2-2025-02-15_13:28    Sat, 2025-02-15 13:28:31 [51bb79218a576bb7fa9284bc00b1126925489dc043fd712fb1eaede5bf1813be]
root@debian:~# cd restauration_repo/
root@debian:~/restauration_repo# borg extract borg@192.168.10.3:/backup/borg_repo::documents-2025-02-15_13:12
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
root@debian:~/restauration_repo# cd ..
root@debian:~# borg delete borg@192.168.10.3:/backup/borg_repo::documents-2025-02-15_13:12
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
root@debian:~# borg list borg@192.168.10.3:/backup/borg_repo
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
documents2-2025-02-15_13:28    Sat, 2025-02-15 13:28:31 [51bb79218a576bb7fa9284bc00b1126925489dc043fd712fb1eaede5bf1813be]
```

```

root@debian:~# ssh borg@192.168.10.1
Linux debian 6.1.0-30-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 14 10:50:02 2025 from 192.168.10.2
This account is currently not available.
Connection to 192.168.10.1 closed.
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:81:3b:ed brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.2/24 brd 192.168.10.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe81:3bed/64 scope link
        valid_lft forever preferred_lft forever
root@debian:~# ssh borg@192.168.10.1
Linux debian 6.1.0-30-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.124-1 (2025-01-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 14 10:52:25 2025 from 192.168.10.2
borg@debian:~$ █

```

Nous allons Initialiser Borg à partir de station1

```
borg init --encryption=repokey borg@192.168.10.1:/backup/borg
```

Dans serveur2 verification

```

root@debian:~# su borg
borg@debian:/root$ cd /backup/borg/
borg@debian:/backup/borg$ ls
config data hints.9 index.9 integrity.9 nonce README
borg@debian:/backup/borg$ █

```

Teste d'une sauvergarde manuelle de mon Dossier Documents dans station1

```
borg create --stats --progress \
```

```
borg@192.168.10.1:/backup/borg::"backup-{now:%Y-%m-%d_%H-%M-%S}" \
```

```
~/Documents
```

Avec exclusions de certains fichier comme les caches


```
root@debian:~# borg extract borg@192.168.10.1:/backup/borg::backup-2025-02-09_13-07-59 /root/Documents/
Enter passphrase for key ssh://borg@192.168.10.1/backup/borg:
root@debian:~# cd /root/Documents/
root@debian:~/root/Documents# ls
booba.txt 'Captures d'écran' test
root@debian:~/root/Documents# █
```

Automatisons les sauvegardes

Sur Station1, nous allons créer le script pour pouvoir automatiser les sauvegardes

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

Dans le script pour éviter de stocker le mot de passe en clair j'ai décidé de le stocker dans un fichier

```
.borg_passphrase
```

```
echo "123" | sudo tee /root/.borg_passphrase > /dev/null
```

```
sudo chmod 600 /root/.borg_passphrase
```

```
GNU nano 7.2 /usr/local/bin/borg-backup.sh
#!/bin/bash

# Variables
export BORG_PASSCOMMAND="cat /root/.borg_passphrase"
REPO="borg@192.168.10.1:/backup/borg"

echo "Début de la sauvegarde..."

# Création d'une nouvelle sauvegarde
borg create --verbose --stats --compression lz4 \
  $REPO::"backup-{now:%Y-%m-%d_%H-%M}" \
  ~/Documents

echo "✅ Sauvegarde terminée."

# Nettoyage des anciennes sauvegardes
borg prune --keep-daily=7 --keep-weekly=4 --keep-monthly=6 $REPO

echo "Nettoyage terminé."
```

Rendons le script exécutable

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

```

GNU nano 7.2 /usr/local/bin/backup.sh
~/bin/bash

# Définition des variables
BORG_REPO="borg@192.168.10.3:/backup/borg_repo"
BORG_PASSPHRASE_FILE="/root/.borg_passphrase"

# Vérification de l'existence du fichier de passphrase
if [[ ! -f "$BORG_PASSPHRASE_FILE" ]]; then
  echo -X Erreur : Le fichier de passphrase Borg n'existe pas ! -> &2
  exit 1
fi

# Chargement de la passphrase
export BORG_PASSPHRASE=$(cat "$BORG_PASSPHRASE_FILE")

# Création de la sauvegarde
echo "🚀 Démarrage de la sauvegarde..."

borg create --compression lz4 --progress --exclude '*.tmp' --exclude 'tmp/' --exclude 'cache/' borg@192.168.10.3:/backup/borg_repo::"documents2-(now:%Y-%m-%d_%H:%M)" ~/Documents

# Nettoyage des anciennes sauvegardes
echo "✓ Suppression des anciennes sauvegardes..."
borg prune -v --list $BORG_REPO \
  --keep-daily=7 --keep-weekly=4 --keep-monthly=6

# Vérification de l'intégrité des sauvegardes
echo "✅ Vérification de l'intégrité des sauvegardes..."
borg check --verify-data $BORG_REPO

echo "🏁 Sauvegarde terminée avec succès !"

```

avec les paramètres

j'Utilise la compression lz4

Affiche la progression

Exclut les fichiers temporaires et les dossiers cache

Sauvegarde le dossier Documents avec un nom incluant la date et l'heure

Nettoyage des anciennes sauvegardes :

Garde les 7 dernières sauvegardes quotidiennes

Conserve 4 sauvegardes hebdomadaires

Maintient 6 sauvegardes mensuelles

Exécuter manuellement le script :

```

root@debian:~# borg list borg@192.168.10.3:/backup/borg_repo
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
documents5-2025-02-15_17:04 Sat, 2025-02-15 17:04:06 [200fc9c22b70e7f1f0a7bd2acd815cf5cdc4472ca140eced5a363ecb473fbb95]
documents4-2025-02-15_17:15 Sat, 2025-02-15 17:15:57 [0e995f954966edcaae0bec4567b75a8f24d3a1133e6f87a36f2682ea8825080]
root@debian:~# borg list borg@192.168.10.3:/backup/borg_repo
Enter passphrase for key ssh://borg@192.168.10.3:/backup/borg_repo:
documents5-2025-02-15_17:04 Sat, 2025-02-15 17:04:06 [200fc9c22b70e7f1f0a7bd2acd815cf5cdc4472ca140eced5a363ecb473fbb95]
documents4-2025-02-15_17:15 Sat, 2025-02-15 17:15:57 [0e995f954966edcaae0bec4567b75a8f24d3a1133e6f87a36f2682ea8825080]
root@debian:~# echo "123" | sudo tee /root/.borg_passphrase > /dev/null
root@debian:~# sudo chmod 600 /root/.borg_passphrase
root@debian:~# sudo nano /usr/local/bin/backup.sh
root@debian:~# sudo chmod +x /usr/local/bin/backup.sh
root@debian:~# backup.sh
🚀 Démarrage de la sauvegarde...
✓ Suppression des anciennes sauvegardes...
Keeping archive (rule: daily #1): documents2-2025-02-15_19:46 Sat, 2025-02-15 19:46:52 [071b2546d5c98c8dfae2db27426e82fefe32519aa8e5b5c34b1f7aa9af865921]
Pruning archive (1/1): documents4-2025-02-15_17:15 Sat, 2025-02-15 17:15:57 [0e995f954966edcaae0bec4567b75a8f24d3a1133e6f87a36f2682ea8825080]
Keeping archive (rule: daily[oldest] #2): documents5-2025-02-15_17:04 Sat, 2025-02-15 17:04:06 [200fc9c22b70e7f1f0a7bd2acd815cf5cdc4472ca140eced5a363ecb473fbb95]
✅ Vérification de l'intégrité des sauvegardes...
🏁 Sauvegarde terminée avec succès !
root@debian:~# █

```

Nous allons sauvegarde de façon automatique en un instant donné par exemple chaque 10 minute

Ouvrons la crontab :

crontab -e

tous les 2 minutes

```
GNU nano 7.2 /tmp/crontab.sJ3uoF/crontab *
# 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
*/2 * * * * /usr/local/bin/borg-backup.sh >> /var/log/borg-backup.log 2>&1
```

Resultat en consultant mon fichier .log

root@debian:~# tail -f /var/log/borg-backup.log

```

root@debian:~# cat /var/log/borg-backup.log
Début de la sauvegarde...
Creating archive at "borg@192.168.10.1:/backup/borg::backup-2025-02-09_15-14"
-----
Repository: ssh://borg@192.168.10.1/backup/borg
Archive name: backup-2025-02-09_15-14
Archive fingerprint: 63f41bb134b115d9191bc6297258dbf74371f5e3981314bf13f4d0e262891256
Time (start): Sun, 2025-02-09 15:14:04
Time (end): Sun, 2025-02-09 15:14:04
Duration: 0.00 seconds
Number of files: 0
Utilization of max. archive size: 0%
-----

```

	Original size	Compressed size	Deduplicated size
This archive:	595 B	589 B	589 B
All archives:	38 B	81 B	2.91 kB

```

-----

```

	Unique chunks	Total chunks
Chunk index:	8	9

```

-----
[✓] Sauvegarde terminée.
Nettoyage terminé.
Début de la sauvegarde...
Creating archive at "borg@192.168.10.1:/backup/borg::backup-2025-02-09_15-16"
-----
Repository: ssh://borg@192.168.10.1/backup/borg
Archive name: backup-2025-02-09_15-16
Archive fingerprint: bc198d71b48750b9bf89e6aa848fc24b8084264f75786a41886ba1252bb38740
Time (start): Sun, 2025-02-09 15:16:04
Time (end): Sun, 2025-02-09 15:16:04
Duration: 0.00 seconds
Number of files: 0
Utilization of max. archive size: 0%
-----

```

	Original size	Compressed size	Deduplicated size
This archive:	595 B	588 B	588 B
All archives:	38 B	81 B	2.91 kB

```

-----

```

	Unique chunks	Total chunks
Chunk index:	8	9

```

-----

```

configuration zfs

```
sudo zpool create backup_pool /dev/sdb disque sdb (10,5 Go)
```

Créez un dataset pour le dépôt Borg

```
sudo zfs create backup_pool/borg_repo2
```

Définir le point de montage

```
sudo zfs set mountpoint=/backup/borg_repo2 backup_pool/borg_repo2
```

Activez la compression

```
sudo zfs set compression=lz4 backup_pool/borg_repo2
```

Vérifiez la configuration

```
zfs list backup_pool/borg_repo
```

Changer le propriétaire du dataset

```
sudo chown -R borg:borg /backup/borg_repo
```

```
# Vérifiez les droits
```

```
ls -ld /backup/borg_repo
```

```
root@debian:/# ls -ld /backup/borg_repo
```

```
drwxr-xr-x 3 root root 2 15 févr. 23:48 /backup/borg_repo
```

```
Création manuelle d'un snapshot (test)
```

```
sudo zfs snapshot backup_pool/borg_repo@initial
```

Rendre les snapshots immuables

script nano /usr/local/bin/zfs_snapshot.sh

```
GNU nano 7.2 /usr/local/bin/zfs_snapshot.sh
#!/bin/bash

# Définition des variables
ZFS_POOL="backup_pool/borg_repo2"
SNAPSHOT_NAME="${ZFS_POOL}@$(date +%Y-%m-%d-%H%M)"

# Créer un snapshot
echo "■ Création du snapshot ZFS: $SNAPSHOT_NAME"
sudo zfs snapshot "$SNAPSHOT_NAME"

# Lister les snapshots du plus récent au plus ancien (tri par date décroissante)
SNAPSHOTS=$(sudo zfs list -H -t snapshot -o name -s creation -r "$ZFS_POOL")

# Garder seulement les 3 derniers snapshots
KEEP=3
COUNT=0

echo "🔍 Analyse des snapshots existants :"
while IFS= read -r SNAPSHOT; do
    if [ $COUNT -lt $KEEP ]; then
        # Appliquer un hold sur les 3 plus récents
        sudo zfs hold borg_protect "$SNAPSHOT" 2>/dev/null || true
        echo "[x] Protégé : $SNAPSHOT"
    else
        # Retirer le hold et supprimer les anciens
        sudo zfs release borg_protect "$SNAPSHOT" 2>/dev/null || true
        sudo zfs destroy -v "$SNAPSHOT"
        echo "[x] Supprimé : $SNAPSHOT"
    fi
    ((COUNT++))
done <<< "$SNAPSHOTS"

echo "✅ Opération terminée. Snapshots restants :"
sudo zfs list -t snapshot -o name,creation -s creation
```

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

```
# Définition des variables
```

```
ZFS_POOL="backup_pool/borg_repo2"
```

```
SNAPSHOT_NAME="${ZFS_POOL}@$(date +%Y-%m-%d-%H%M)"
```

```
# Créer un snapshot
```

```
echo "Création du snapshot ZFS: $SNAPSHOT_NAME"
```

```
sudo zfs snapshot "$SNAPSHOT_NAME"

# Lister les snapshots du plus récent au plus ancien (tri par date décroissante)
SNAPSHOTS=$(sudo zfs list -H -t snapshot -o name -S creation -r "$ZFS_POOL")

# Garder seulement les 3 derniers snapshots

KEEP=3

COUNT=0

echo " Analyse des snapshots existants : "

while IFS= read -r SNAPSHOT; do

    if [ $COUNT -lt $KEEP ]; then

        # Appliquer un hold sur les 3 plus récents

        sudo zfs hold borg_protect "$SNAPSHOT" 2>/dev/null || true

        echo "[✓] Protégé : $SNAPSHOT"

    else

        # Retirer le hold et supprimer les anciens

        sudo zfs release borg_protect "$SNAPSHOT" 2>/dev/null || true

        sudo zfs destroy -v "$SNAPSHOT"

        echo " Supprimé : $SNAPSHOT"

    fi

    ((COUNT++))

done <<< "$SNAPSHOTS"

echo " Opération terminée. Snapshots restants : "

sudo zfs list -t snapshot -o name,creation -s creation
```

lister les snapshot

```
root@debian:/# zfs list -t snapshot
```

```
NAME                                USED AVAIL  REFER MOUNTPOINT
backup_pool/borg_repo2@2025-02-16-1530 89.5K   -   550K -
backup_pool/borg_repo2@2025-02-16-1531  0B     -   556K -
backup_pool/borg_repo2@2025-02-16-1532  0B     -   556K -
root@debian:/#
```

zfs list -t snapshot

liste es snapshot supprimer :

```
root@debian:/# zpool history | grep destroy
```

verifier les log

```
tail -f /var/log/zfs_snapshots.log
```

```
root@debian:/# tail -f /var/log/zfs_snapshots.log
[✓] Protégé : backup_pool/borg_repo2@2025-02-16-1246
[✓] Protégé : backup_pool/borg_repo2@2025-02-16-1224
will destroy backup_pool/borg_repo2@2025-02-16-1222
will reclaim 0B
[✗] Supprimé : backup_pool/borg_repo2@2025-02-16-1222
[✓] Opération terminée. Snapshots restants :
NAME                                CREATION
backup_pool/borg_repo2@2025-02-16-1224 dim. févr. 16 12:24 2025
backup_pool/borg_repo2@2025-02-16-1246 dim. févr. 16 12:46 2025
backup_pool/borg_repo2@2025-02-16-1247 dim. févr. 16 12:47 2025
^C
root@debian:/# █
```

tenter de supprimer le repo2

```

root@debian:/# sudo rm -rf /backup/borg_repo2
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/181': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/118': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/112': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/165': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/7': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/211': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/32': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/45': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/38': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/157': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/120': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/77': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/99': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/223': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/254': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/229': Système de fichiers accessible en lecture seulement
rm: impossible de supprimer '/backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/data/0/93': Système de fichiers accessible en lecture seulement

```

Restauration d'un snapshots par copie pas par rollback pour éviter d'écraser les données ajoutés après le snapshot

```

backup_pool1/borg_repo2@2025-02-16-1532  55      -      330K      -
root@debian:/# cp -a /backup/borg_repo2/.zfs/snapshot/2025-02-16-1532/* /backup/borg_repo2/
root@debian:/# cd backup
root@debian:/backup# ls
borg_repo2
root@debian:/backup# cd borg_repo2/
root@debian:/backup/borg_repo2# ls
borg_repo2_tmp  config  data  hints.257  index.257  integrity.257  nonce  README
root@debian:/backup/borg_repo2# █

```

Résumé :

Borg étant un logiciel de sauvegarde avancé qui offre plusieurs fonctionnalités essentielles pour optimiser le stockage et la gestion des archives. Parmi ses principales caractéristiques, on trouve.

La Déduplication → Optimise l'espace et la vitesse des sauvegardes en stockant uniquement les blocs de données nouveaux ou modifiés.

Gestion des anciennes sauvegardes (prune) → Permet de conserver uniquement les sauvegardes nécessaires et d'économiser de l'espace disque.

Borg permet l'accès aux archives passées et offre des outils pour les explorer et les restaurer facilement.

Un bon usage de prune et compact assure une gestion efficace des sauvegardes en évitant une accumulation inutile de données.