

ZFS Cheat Sheet – l'antisèche ZFS

V1.4 - Hervé Suaudeau – herve.suaudeau@cnrs.fr – SPPIN (CNRS UMR 8003) – octobre 2026



INSTALLER OPENZFS (DEBIAN 13)

Sur Debian 13, il faut ajouter 'contrib' en fin de ligne de `/etc/apt/sources.list` puis exécuter :

```
apt update; apt install linux-headers-amd64 \
zfsutils-linux zfs-dkms zfs-zed
```

et redémarrer.

Tester si le module est chargé : `modprobe zfs`

PRÉPARER LES DISQUES

DISQUES

Créer les alias simples des disques

But : ajouter au fichier `/etc/zfs/vdev_id.conf` des lignes au format :

```
alias disk0 pci-0000:07:00.0-sas-0x12210002000000-lun-0 #sdc
alias disk1 pci-0000:07:00.0-sas-0x12210003000000-lun-0 #sdd
alias disk2 ...
```

Étapes :

- Récupérer les UUID de chaque disque :
`ls -lh /dev/disk/by-id/`
- Ajouter les alias des UUID dans `/etc/zfs/vdev_id.conf`
- Activer les modifications :
`udevadm trigger`
- Vérifier la prise en compte :
`ls -al /dev/disk/by-vdev/`

Script pour générer les alias (à adapter) :

```
nb=1
for ScsiId in /dev/disk/by-id/scsi-*; do
    SdId=$(basename "$(readlink -f $ScsiId)")
    ScsiIdName=$(basename "$ScsiId")
    echo "alias disk$nb $ScsiIdName #${SdId}"
    nb=$((nb+1))
done
```

done

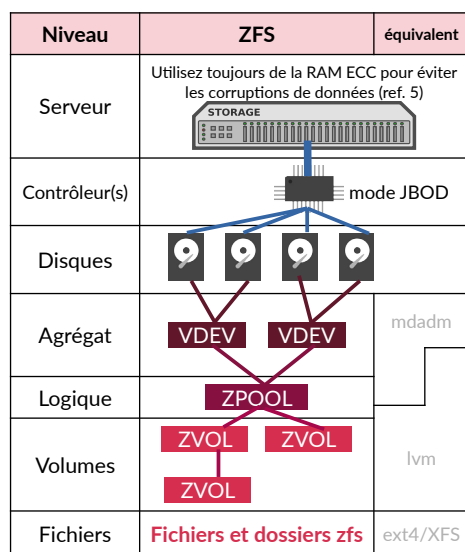
Effacer les disques (optionnel)

Instruction d'effacement des MBR :

```
params="bs=512 count=1"
dd if=/dev/zero of=/dev/disk/by-vdev/disk0 $params
dd if=/dev/zero of=/dev/disk/by-vdev/disk1 $params
...
```

ARCHITECTURE ZFS

DISQUES



CRÉER POOLS ET VDEVs

ZPOOL

Nombre de VDEV

Les disques peuvent être répartis sur plusieurs VDEVs avec les conséquences suivantes :

	1 VDEV	2 VDEV et +
Optimisation de l'espace	Bonne. Un seul jeu de disque de parité	Réduite. Chaque VDEV doit avoir ses propres disques de parité
Temps de reconstruction	⚠ Maximal	Divisé par le nombre de VDEV
Performance R/W		Optimale car parallélisée

⚠ La reconstruction est une opération qui stresse fortement les disques pendant plusieurs jours sur de gros VDEV. Minimisez ce temps !

Exemple de création de pool

Créer un pool nommé « mypool » contenant 2 VDEV de 6 disques en raid6 (8 disques utiles) :

```
zpool create -o ashift=12 mypool \
    raidz2 disk0 disk1 disk2 disk3 disk4 disk5 \
    raidz2 disk6 disk7 disk8 disk9 disk10 disk11
```

Niveaux RAID ZFS

L'instruction `zpool` permet de configurer la redondance et l'entrelacement des données :

Niveau	Mot clef ZFS	Commentaire
RAID0	Aucun	Disques agrégés (stripe)
RAID1	« mirror »	Miroirs de disques
RAID10	Pas de mot clef mais il est possible de configurer avec l'instruction suivante : <code>zpool create mypool mirror disk0 disk1 \ mirror disk2 disk3</code>	
RAID5	« raidz »	1 disque de parité par pool.
RAID6	« raidz2 »	2 disques de parité par pool
	« raidz3 »	3 disques de parité par pool
	« draid »	Nouveau : utile pour + 100 disques [3]. Peu éprouvé ?

Recommandations (ref. 3).

- Mettre 3 à 9 gros disques par VDEV, au max. 12
- `raidz` est à éviter pour des disques > 1Tio
- Éviter de remplir vos disques à plus de 80 %, jamais au-delà de 90 % (effondrement des performances, risque au changement de disque)

CONFIGURER LES POOLS

ZPOOL

Réglages conseillés

- Désactiver la déduplication (délétère pour la RAM) : `zfs set dedup=off mypool`
- Activer la compression :
`zfs set compression=lz4 mypool`
- Désactiver l'access-time et la synchronisation immédiate (meilleures performances) :
`zfs set atime=off sync=disabled mypool`
- Autres valeurs recommandées (Stocker les attributs étendus dans les inodes afin d'obtenir plus d'IO) :
`zfs set xattr=sa dnodesize=auto mypool`
- Ajouter un disque SSD de cache (cache L2ARC) :
`zpool add mypool cache disk_nvme1`

Réglages optionnels

Ajouter un disque de spare :

```
zpool add mypool spare disk12
```

CHANGER UN DISQUE EN PANNE

ZPOOL

- Repérer le disque avec : `zpool status -v`
- Désactiver disque : `zpool offline tank diskX`
- Remplacer physiquement le disque.
- Repérer le nouveau disque :
`ls -lh /dev/disk/by-id/`
- Dans `/etc/zfs/vdev_id.conf`, créer un alias `diskXb` de l'UUID du nouveau disque.
- Prise en charge des modifications :
`udevadm trigger`
- Identifier le guid de l'ancien disque : `zdb`
- Remplacer le disque dans le pool :
`zpool replace -f mypool ancien_guid diskXb`

SCRIPT DE SURVEILLANCE

ZPOOL

Si `zfs-zed`, `postfix` et `mailutils` ont été installés, il suffit de rajouter au fichier `/etc/zfs/zed.rc` :

```
ZED_EMAIL_ADDR="your@email.addr"
ZED_NOTIFY_VERBOSE=1
ZED_SYSLOG_TAG="zed"
```

puis de redémarrer zed : `systemctl restart zed`

CONFIGURER LES VOLUMES

ZVOL

- Créer et monter un volume :
`zfs create mypool/data1`
`zfs set mountpoint=/data1 mypool/data1`
`zfs mount -a`
- Option à rajouter en cas de partage Samba :
`zfs create -o casesensitivity=mixed mypool/data1`
- Optionnel : mettre en lecture seule le volume.
`zfs set readonly=on mypool/data1`

INSTANTANÉS

ZVOL

- Lister les instantanés : `zfs list -t snapshot`
- Les instantanés sont sur `/data1/.zfs/snapshot`
- Créer un instantané tagué à la date du jour (simple ou récursif) :
`zfs snapshot mypool/data1@2026-01-11`
`zfs snapshot -r mypool/data1@2026-01-11`
- Détruire un instantané (ou de façon récursive) :
`zfs destroy mypool/data1@2026-01-11`
`zfs destroy -r mypool/data1@2026-01-11`
- Connaître la différence d'un instantané avec le volume en cours ou un autre instantané :
`zfs diff mypool/data1@backup1`
`zfs diff mypool/data1@backup1 data1@backup2`
- Restaurer un volume à partir du dernier instantané : `zfs rollback mypool/data1@lastback`
Pour restaurer de plus vieux instantané et effacer les intermédiaires, utiliser l'option `-r`.
- Créer une copie modifiable d'un instantané :
`zfs clone mypool/data1@day mypool/data2@test`

SE RENSEIGNER

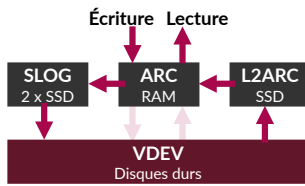
Voir l'état des pools : `zpool status -v`

Voir la liste de volumes : `zfs list`

Liste des commandes : `zfs` ou `zpool`

Doc d'une commandes : `man zfs-commande`

OPTIMISER LES CACHES

DISQUES
ZPOOL

RAM (ARC : adjustable Replacement Cache)

- La RAM contient le cache de lecture principal et les transactions en cours d'écriture. Maximisez-les pour profiter des meilleures performances.
- Vérifier la taille allouée (50 % de la RAM par défaut) : `arc_summary -s arc | grep Target`
- Il faut l'ajuster dans `/etc/modprobe.d/zfs.conf` en fonction des usages RAM des autres applications (si aucunes mettre 75%) puis rebooter. Exemple de contenu pour 192 GiO d'ARC (valeur obtenue par 192×1024^3) : options `zfs zfs_arc_max=206158430208`

Cache lecture (L2ARC : level 2 ARC)

- Le cache de lecture niveau 2 (L2ARC) est une extension sur SSD de l'ARC.
- Maximisez l'achat de RAM plutôt que le L2ARC. Il faut privilégier les SSD Nvme performants et des disques en parallèle pour plus de rapidité.
- Attribuer trop de cache L2ARC réduit les performances : Prévoir 1GiO de cache ARC supplémentaire en RAM pour chaque 50GiO de cache disque [3].

Taille minimum	Taille maximum
2 GiO par TiO de disque [5] et au moins 2 fois la RAM	10 à 20 fois la RAM [3]

- Pour ajouter un disque en cache :
`zpool add mypool cache disk_nvme1`

Cache écriture (SLOG pour le ZIL)

- Le cache SLOG (*separate intent log*) est la transposition sur SSD du cache d'écriture ZIL (*ZFS Intent Log*) du pool ZFS sur disque.
- Son utilisation est recommandée pour les écritures synchrones intensives (bases de données, NFS, virtualisation, sauvegardes). Il n'est généralement pas bénéfique pour les usages plus courants sauf à vouloir sécuriser les écritures asynchrones [3].
- Les disques de cache devront être assemblés obligatoirement en miroir (la perte du ZIL pouvant entraîner la corruption du pool). Il faut privilégier les SSD conçus pour l'écriture intensive, voire des NVDIMMs [1].
- TrueNAS préconise des disques entre 8 et 32GiO avec - pour améliorer leur durée de vie - de l'over-provisioning (attribuer une partition plus petite que l'espace disque total).
- Pour configurer un SLOG sur deux partitions :
`zpool add mypool log mirror nvme1p1 nvme2p1`

OPTIMISER LES RÉGLAGES

ZPOOL

Ashift

`zpool create -o ashift=XX` permet de fixer la tailles des secteurs. La surévaluer a beaucoup moins d'impact que de la sous-évaluer :

- `ashift=12` : secteurs de 4K, généralement la bonne valeur
- `ashift=13` : secteurs de 8K (quelques SSDs modernes)

Pour connaître la taille des secteurs des disques :

`lsblk --nodeps -o NAME,PHY-SEC`

Recordsize

Par défaut à 128KiO, cette valeur peut être ajustée en fonction des cas d'usages (voir plus loin) :

`zfs set recordsize=16K mypool/home`

Redundant_metadata

Lorsque les données ne sont pas trop critiques, il est possible de réduire la duplication des métadonnées pour accélérer les accès (typiquement pour les bases de données) :

`zfs set redundant_metadata=most mypool/dir`

Cas d'usages

Optimiser certains paramètres permet d'améliorer les performances en fonction de cas particuliers [6]. Toujours tester l'impact de ces changements.

	Bases de données	Virtualisation	Backup, médias
Géométrie	mirror (plus rapide sur les IOPs)		raidz, raidz1, ou raidz3
Volumes	Faire des volumes séparés pour les données et les logs (WAL sur Postgre)	Volume RAW <code>zfs create -V 50G mypool/raw1</code> Option <code>-s</code> pour un volume « sparse » quigrossi à l'usage. Proxmox gère cela.	Faire autant de volumes et réglages que d'usage
recordsize	16K (MySQL, PostgreSQL, KVM, Proxmox)		1M pour les grands fichiers
Compression	<code>zfs set compression=lz4 mypool/zvol</code> (rapide)		zstd-3, zstd-6 (normal ou archivage)
Autres	https://openzfs.github.io/openzfs-docs/Performance%20and%20Tuning/Workload%20Tuning.html#database-workloads		

SCRUB

ZPOOL

Le scrub vérifie et corrige les erreurs en lisant tous les blocs et en validant leurs sommes de contrôle.

Lancement automatique : Debian lance un scrub tous les mois : `/etc/cron.d/zfsutils-linux`

Lancement manuel :

- Lancement sur un pool : `zpool scrub mypool`
- Voir l'état d'avancement : `zpool status mypool`
- Éventuel arrêt : `zpool scrub -s mypool`

CHIFFREMENT

ZPOOL
ZVOL

Il faut rajouter les option `encryption` et `keyformat` au moment de la création du pool. Exemple :

```
zpool create -o ashift=12 -O encryption=on \
-O keyformat=passphrase mypool \
raidz disk0 disk1 disk2 disk3 disk4 disk5
```

Puis rentrer une phrase de passe.

NB : pour chiffrer un volume plutôt qu'un pool :

```
zfs create -o encryption=on -o \
keyformat=passphrase mypool/dir
```

Déverrouillage automatique

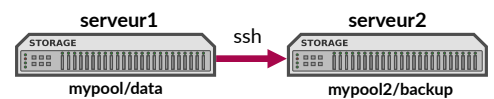
Exemple de script de configuration :

```
sudo mkdir -p /etc/zfs/keys
sudo chmod 700 /etc/zfs/keys
echo "passphrase" > /etc/zfs/keys/pass.txt
Pour charger la clef de déchiffrement au boot,
rajouter dans le crontab (crontab -e) la ligne :
@reboot /usr/sbin/zfs load-key -L
file:///etc/zfs/keys/pass.txt
mypool; /usr/sbin/zfs mount -a
```

Ne pas oublier de sauvegarder `pass.txt`.

ENVOI PAR RÉSEAU

ZVOL



Le but est ici d'envoyer un volume d'une machine à une autre par le réseau en bénéficiant de la vitesse de la copie incrémentielle par bloc.

Première sauvegarde

- Sur **serveur2** créer le volume de réception (ne pas le monter pour ne pas modifier l'instantané cible) : `zfs create -p -o mountpoint=none mypool2/data`
- Sur **serveur1** créer un instantané de transfert : `zfs snapshot mypool/data@day1`
- Lancer la sauvegarde initiale : `zfs send -vP mypool/data@day1 | ssh root@serveur2 zfs receive -F mypool2/backup`

Pour envoyer un volume, ses sous-volumes et leurs instantanés, il faut ajouter l'option `-r` à `zfs snapshot` et l'option `-R` à `zfs send`.

Sauvegarde incrémentielle

- Attendre quelques temps.
- Créer un nouvel instantané : `zfs snapshot mypool/data@day2`
- Envoi de la différence sur serveur2 : `zfs send -i day1 mypool/data@day2 | ssh remotehost zfs recv -F mypool2/backup`

Sauvegarde chiffrée

Pour recevoir une sauvegarde chiffrée sans la déchiffrer (zero knowledge backup, option `-w`) :

```
zfs send -vPw mypool/data@day1 | ssh root@serveur2
zfs receive -F mypool2/backup_not_decrypted
```

Cloner encore plus simplement

Avec `syncoid` (`apt install sanoid`) le clonage entre deux machines, de deux pools montés et leur snapshots, s'effectue très simplement avec une seule ligne :

```
sudo syncoid mypool1/data \
root@serveur2:mypool2/data -sshport=22 -r
```

QUOTAS

ZVOL

Mettre un quota :

`zfs set quota=160T mypool/myvol`

ASTUCES DIVERSES

- Renommer un pool : `zpool export oldname ; zpool import oldname newname`
- Connaitre la taille des volumes (ne pas utiliser `df -h`) : `zfs list -o space`
- Partager et accéder par NFS un volume :
`zfs set sharenfs=on mypool/data/share`
Accès par `mount -t nfs serveur1:/share /mnt/share`

RÉFÉRENCES

- Groupe de Travail ZFS : <https://resinfo-gt.pages.in2p3.fr/zfs/doc/>
- Wiki officiel : https://openzfs.org/wiki/Main_Page
- Documentation ZFS de TrueNAS : <https://www.truenas.com/docs/references/> et <https://www.truenas.com/docs/scale/gettingstarted/scalehardwareguide/>
- Pages man : `man zpoolconcepts`
- Doc Debian : <https://wiki.debian.org/ZFS>
- OpenZFS official documentation : <https://openzfs.github.io>