

☞ Sous FreeBSD le shell par défaut est `csh`, les commandes données ci-dessous utilise la syntaxe de `bash`

☞ La VM est configurée pour faire du port forwarding sur le serveur ssh depuis le loopback port 2222.

Exercice 1 Création du pool

Part I

1. Créez le *pool* `kko` (disque : `da0`)
2. Transformez le *pool* en un miroir à 2 disques (disque : `da1`)
3. Transformez le *pool* en un mirror à 3 disques (disque : `da2`)
4. Revenir à un miroir à 2 disques (suppression de `da2`)
5. Étendre le *pool* en un “raid10” (disques : `da2` et `da3`)
☞ on peut considérer que les éléments du *pool* fonctionnent en *stripping*.
6. Ajoutez un log séparé (disque : `da4`)
7. Ajoutez un cache (disque `da5`)
8. Ajoutez un hot-spare (disque `da6`)

Part II

9. Détruire le *pool* `kko`
10. Recréer le *pool* `kko` avec la même configuration mais en utilisant une seule commande
11. Y créer quelques fichiers
☞ `for i in 1 2 3 4; do dd count=12345 if=/dev/urandom of=/kko/foo_$i ; done`

Part III

12. Scindez le *pool* pour obtenir le *pool* `kfe`
13. Vérifier que le *pool* `kfe` contient des données identiques au *pool* `kko`
☞ le *pool* n’est pas importé lors de la scission
☞ suivant le type de fichier, pour la comparaison vous pouvez utiliser : `md5`, `sha256`, `diff`, ...

Part IV

14. Transformez le *pool* `kfe` pour qu’il soit résistant à la perte d’un disque (disques à disposition : `da7`, `da8` et `da9`)
15. Simuler la perte d’un des disques, vérifier que les données sont toujours accessibles
☞ utiliser la commande `offline` pour simuler la perte du disque

Exercice 2 Disques défectueux

Part I

1. Créez le *pool* `kko`, choisissez un `raidz2` (disques : `da0`, `da1`, `da2`, `da3`).
2. Désactiver le cache (ARC et L2ARC) :
☞ la désactivation du cache permettra une remontée plus rapide des erreurs disques

```
zfs set primarycache=none kko
zfs set secondarycache=none kko
```
3. Créez quelques fichiers (une centaine de mégaoctets en tout) et sauvegarder leurs sommes de contrôle :

```
for i in 1 2 3 4 5; do dd count=12345 if=/dev/urandom of=/kko/foo_$i ; done
sha256 /kko/foo_* > /tmp/kko.sha
```
4. Simulez des erreurs sur le disque `da0` (20% d’erreurs en lecture, 20% d’erreurs en écriture) :

```
gnop configure -r 20 -w 20 md0.nop
```

5. Vérifiez que les différents fichiers sont toujours accessibles et sans erreurs dans les données :
`shasum -c /tmp/kko.sha`
6. Exécutez la commande `zpool status`
7. Lancez une vérification du *pool* et regarder à nouveau son état
8. Recommencez quelques fois les étapes 5 à 7
9. Recommencez les étapes 4 à 8 en effectuant les substitutions `da0` $\rightarrow$ `da1` et `md0.nop` $\rightarrow$ `md1.nop`.
10. Aquiter les erreurs

Part II

11. Remplacer le disque défectueux `da0` (disque : `da4`)
12. Ajouter un hot spare (disque : `da5`)
13. Lancer une vérification du *pool* et regarder son status
14. Vérifier que le hot-spare a été utilisé pour remplacer le disque deffectueux restant (`da1`)
 ☞ si le hot-spare n'a pas été utilisé, c'est que le système n'a pas estimé qu'il y avait suffisant d'erreurs pour considérer le disque comme invalide : augmenter les erreur de lecture/écriture sur `da1` à 50% et recommencez l'étape 13.

Part III

15. Simulez des erreurs sur les disques `da2`, `da3` et `da4` :
`for i in 2 3 4; do gnop configure -r 20 -w 20 md${i}.nop; done`
16. Vérifiez si les différents fichiers sont accessibles et correctes :
`md5 /kko/*`
17. Lancez une vérification du *pool*
18. Utilisez la commande `zpool status -v` pour trouver les fichiers définitivement corrompus

Exercise 3 Ma petite entreprise

Part I

1. Vous décidez d'utiliser ZFS pour stocker vos données.
 ☞ Créez le *pool* `kko` à partir du disque `da1`.
2. Après avoir failli perdre vos données suite a des secteurs defetueux sur le disque vous decidez d'investir dans un disque supplémentaire `da2`.
 ☞ Transformez le *pool* pour qu'il utilise un miroir a 2 disques
3. Grace à l'utilisation de `smartmontools` vous repérez que le disque `da2` montre des signes de faiblesse.
 ☞ Remplacez le disque `da2` par le disque `da4`
4. L'affaire se developpe bien, il vous faut plus d'espace de stockage, vous achetez un 3^{ème} disque (`da3`).
 ☞ Essayez d'ajouter ce disque au *pool* pour en augmenter la capacité
 ☞ Vous decidez donc de transformer le mirroir en un raidz à 3 disques
 ➤ Casser le miroir
 ➤ Créer un pool raidz dégradé (utilisation d'un périphérique disque créé via un fichier partiellement alloué : `truncate -s 100G /tmp/sparse-file ; mdconfig /tmp/sparse-file`)
 ➤ Recopier les fichiers (`send/recv`)

Part II

Créez des *datasets* pour y héberger :

1. les données : `kko/data` $\rightarrow$ `/kko/data`
 - la compta : `kko/data/compta` $\rightarrow$ `/external/accounting`
 - les produits : `kko/data/products` $\rightarrow$ `/kko/data/products`
2. les site web :
 - `kko/web/www.choco-plaisir.fr` $\rightarrow$ `/web/www.choco-plaisir.fr`
 - `kko/web/www.cacao-passion.fr` $\rightarrow$ `/web/www.cacao-passion.fr`
3. les logs (web uniquement) : `kko/log/web` $\rightarrow$ `/var/log/www`

4. la base MySQL : `kko/db/mysql` → `/var/db/mysql`

Vous devez savoir que :

- les données de la compta sont partagées via samba, donc pour des personnes sous windows, qui sont habitués à un système de fichier ne faisant pas de différences de casses.
- le répertoire produits doit contenir des noms de fichiers au format UTF-8 et vous souhaitez vous assurer que cela soit bien le cas et que la comparaison des noms de fichiers utilise une forme composée)
- les fichiers de log seront compressés afin de minimiser la place qu'ils occupent car vous souhaitez pour des raisons légal les conserver pendant 1 an
- pour optimiser les performances vous connaissez les informations suivantes sur la base de donnée mysql (innodb) :
 - elle n'utilise en interne que des blocs de 16k
 - elle sait très bien gérer son cache de données
 - il est inutile d'enregistrer la date d'accès aux fichiers

Part III

- Après avoir perdu une partie de vos données suite à l'effacement du mauvais répertoire, vous vous dites qu'effectuer des *snapshots* régulièrement pourrait vous sauver la mise la prochaine fois.
 - ☞ écrire un script qui lorsqu'il est exécuté crée une snapshot du pool avec la date courante (ex : `02018-05-31`) et exécuter le script
- un ranconware a chiffré vos données dans `/external/accounting`, et vous réclame 100 bitcoin pour vous fournir la clé de déchiffrement...
 - ☞ restaurer la snapshot d'aujourd'hui
- un employé indelicat a effacé quelques-unes de vos fiches produits et vous ne pouvez pas effectuer de *rollback* car vous avez fait une grosse mise à jour de vos fichiers et modifiés bon nombre de ceux-ci
 - ☞ identifiez les fichiers supprimés ☞ restaurer uniquement les fichiers supprimés

Part IV

1. vous souhaitez faire de profondes modifications à votre site web, et conserver l'ancien tant que la nouvelle version n'a pas été approuvée
 - ☞ effectuez un clone de `www.choco-passion.fr` → `new.choco-passion.fr`
2. le nouveau site est approuvé
 - ☞ supprimez l'ancien et renommez `new.choco-passion.fr` en `www.choco-passion.fr`
 - 👉 des étapes intermédiaires sont nécessaires

Part V

1. Vous souhaitez utiliser des VM pour héberger différents services
 - ☞ créer 3 disques de 30G qui serviront de montage iSCSI pour vos VM
2. L'OS installé dessus étant le même pour toutes les VM, vous décidez d'activer la déduplication

Part VI

- créez un quota de 300Mo pour l'utilisateur joe (`/home/joe`)
- assurez-vous que l'utilisateur admin (`/home/admin`) dispose d'au moins 10Mo

Part VII

```
zfs send a@b | dd count=1000 | zfs recv -s w/b
zfs get receive_resume_token w/b
```