FreeBSD zfs iscsi howto

Table of Contents

L'utilisation de ZFS sur FreeBDS offre la possibilité d'utiliser un stockage externe basé sur ZFS via iSCSI. Pour cela il suffit de, créer un volume zfs `zfs create -V <size> ...` puis configurer le partage iscsi du périphérique `/dev/zvol/path/to/device` dans **ctl.conf** [1].

Création d'un volume zfs

Créer un pool RAID-Z, en spécifiant les disques à ajouter au pool :

```
zpool create storage raidz da0 da1 da2
```

ZFS implémente RAID-Z, une variante du standard RAID-5 qui offre une meilleure répartition de la parité et élimine le “trou d'écriture RAID-5” dans lequel les données et les informations de parité deviennent incohérentes après un redémarrage inattendu. ZFS prend en charge trois niveaux de RAID-Z qui offrent différents niveaux de redondance en échange de niveaux décroissants de stockage utilisable. Les types sont nommés RAID-Z1 à RAID-Z3 en fonction du nombre de périphériques de parité dans la matrice et du nombre de disques susceptibles de tomber en panne tant que le pool reste opérationnel.

Dans une configuration RAID-Z1 (**raidz**) avec quatre disques de 1 To chacun, le stockage utilisable est de 3 To et le pool pourra toujours fonctionner en mode dégradé avec un disque défaillant. Si un disque supplémentaire est mis hors ligne avant que le disque défaillant ne soit remplacé et réargenté, toutes les données du pool peuvent être perdues.

Dans une configuration RAID-Z3 (**raidz3**) avec huit disques de 1 To, le volume fournira 5 To d'espace utilisable et pourra toujours fonctionner avec trois disques défectueux. Sun ne recommande pas plus de neuf disques dans un seul vdev. Si la configuration comporte plus de disques, il est recommandé de les diviser en vdev distincts et les données du pool seront réparties entre eux.

Puis créer un volume zfs:

```
zfs create -V 2t -o compression=on storage/virtual-win10-workstation
```

Un volume est un type spécial de jeu de données. Plutôt que d'être monté en tant que système de fichiers, il est exposé en tant que périphérique de bloc sous `/dev/zvol/poolname/` dataset. Cela permet au volume d'être utilisé pour d'autres systèmes de fichiers, de sauvegarder les disques d'une machine virtuelle ou d'être exporté à l'aide de protocoles comme iSCSI ou HAST.

Un volume peut être formaté avec n'importe quel système de fichiers ou utilisé sans système de fichiers pour stocker des données brutes. Pour l'utilisateur, un volume semble être un disque ordinaire. Mettre des systèmes de fichiers ordinaires sur ces zvols fournit des fonctionnalités que les disques ou systèmes de fichiers ordinaires n'ont normalement pas. Par exemple, l'utilisation de la propriété compression sur un volume de 250 Mo permet la création d'un système de fichiers FAT compressé

Ajout du volume à ctl.conf.

Ce qui suit est un exemple de fichier de configuration /etc/ctl.conf simple:

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}

target iqn.2012-06.com.example:target0 {
    auth-group no-authentication
    portal-group pg0

    lun 0 {
        path /dev/zvol/storage/virtual-win10-workstation
        size 2T
    }
}
```

La première entrée définit le groupe de portails pg0. Les groupes de portail définissent les adresses réseau sur lesquelles le démon ctld écoutera. L'entrée `discovery-auth-group no-authentication` indique que tout initiateur est autorisé à effectuer la découverte de cible iSCSI sans authentification. Les lignes suivantes configurent ctld pour écouter toutes les adresses IPv4 (listen 0.0.0.0) et IPv6 (listen [::]) sur le port par défaut de 3260.

Il existe un groupe de portails intégré appelé **default** pour lequel, la découverte de la cible est toujours refusée, alors qu'avec pg0, elle est toujours autorisée.

La deuxième entrée définit une cible unique. La cible a deux significations possibles : une machine servant iSCSI ou un groupe nommé de LUN. Cet exemple utilise cette dernière signification, où

`iqn.2012-06.com.example:target0` est le nom de la cible. Ce nom de cible convient à des fins de test. Pour une utilisation réelle, remplacer com.example par le vrai nom de domaine, inversé. Le 2012-06 représente l'année et le mois d'acquisition du contrôle de ce nom de domaine, et target0 peut être n'importe quelle valeur. N'importe quel nombre de cibles peut être défini dans ce fichier de configuration.

La ligne `auth-group no-authentication` permet à tous les initiateurs de se connecter à la cible spécifiée et le groupe de portails pg0 rend la cible accessible via le groupe de portails pg0.

La section suivante définit le LUN. Pour l'initiateur, chaque LUN sera visible en tant que périphérique de disque distinct.

Plusieurs LUN peuvent être définis pour chaque cible. Chaque LUN est identifié par un numéro (LUN 0 est obligatoire):

```
lun 0 {
path /dev/zvol/storage/domain-controller-1-iscsi-target
}
lun 1 {
path /dev/zvol/storage/domain-controller-2-iscsi-target
}
lun 2 {
path /dev/zvol/storage/virtual-win10-workstation
}
```

La ligne `path /dev/zvol/storage/virtual-win10-workstation` définit le chemin complet vers un zvol supportant le LUN. Ce chemin doit exister avant de démarrer **ctld**. La deuxième ligne est facultative et spécifie la taille du LUN.

L'exemple précédent est intrinsèquement non sécurisé car il n'utilise aucune authentification, accordant à quiconque un accès complet à toutes les cibles.

Pour exiger un nom d'utilisateur et un mot de passe pour accéder aux cibles, modifier la configuration comme suit :

```
auth-group ag0 {
    chap username1 secretsecret
    chap username2 anothersecret
}
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}
target iqn.2012-06.com.example:target0 {
    auth-group no-authentication
```

```
    portal-group pg0
    lun 0 {
        path /dev/zvol/storage/virtual-win10-workstation
    }
}
```

La section `auth-group` définit les paires nom d'utilisateur et mot de passe. Un initiateur essayant de se connecter à **iqn.2012-06.com.example** : target0 doit d'abord spécifier un nom d'utilisateur et un secret définis. Cependant, la découverte de la cible est toujours autorisée sans authentification. Pour exiger l'authentification de la découverte de la cible, il faut définir `discovery-auth-group` sur un nom de groupe d'authentification défini au lieu de `no-authentication`.

Il est courant de définir une seule cible exportée pour chaque initiateur. Pour résumer la syntaxe ci-dessus, le nom d'utilisateur et le mot de passe peuvent être spécifiés directement dans l'entrée cible :

```
target iqn.2012-06.com.example:target0 {
    portal-group pg0
    chap username1 secretsecret

    lun 0 {
            path /dev/zvol/storage/virtual-win10-workstation
        }
}
```

Activation du service

Pour s'assurer que le démon ctld est démarré au démarrage, ajoutez cette ligne à /etc/rc.conf :

```
root@majestix /etc # cat /etc/rc.conf
....
....
#iSCSI
ctld_enable="YES"
....
....
```

```
service ctld start
```

ou si le démon est déjà activé

```
service ctld reload
```

Ça y est, la cible iscsi peut déjà être utilisée.

1)

ctl.conf - Fichier de configuration du démon cible CAM Target Layer / iSCSI