

Proxmox Debian als ISCSI Server verwenden

Systemannahme: Debian9 mit ZFS / Proxmox5 mit ZFS

OpenSCSI macht immer wieder Probleme, die Konfiguration ist meinesachtens kompliziert und oft verwirrend. Die Lösung bietet „**targetcli**“. Targetcli ist seit 2.6.38 Bestandteil des Linux Kernels. Da ganze installiert sich in Debian gleich einfach wie in Ubuntu.

```
apt install targetcli (Ubuntu)
apt install targetcli-fb (Debian9)
```

Erstellen des Targets

Um nun eine ISCSI-Freigabe auf unserem Server zu bekommen erstellen wir zunächst ein Zvol.

```
zfs create -V 50gb v-machines/testvm.deepdoc.at
```

Damit wir uns später auch noch auskennen, nehmen wir als zvolname gleich den FQDN unserer VM wo wir den Speicher verwenden wollen. Dieses Zvol ist dann auch wie wenn man es direkt verwenden würde unsere virtuelle Festplatte. Oder was auch immer. Hat man das Zvol geht es an unsere Cli. Um nun mit der Konfuration zu starten, geben wir den Befehl `targetcli` im Terminal ein. Mittels `ls` sieht man die Baumstruktur. Diese ist auch als solche zu behandeln. Man kann mit den default Linuxbefehlen darin navigieren. `help` zeigt die Hilfe, mittels `cd` kann man mit Pfeiltasten navigieren, und mit `saveconfig` wird gespeichert.

Das ganze sieht nun so aus:

```
/> ls
o- /
.....
..... [....]
  o- backstores
.....
..... [....]
    | o- block
.....
.. [Storage Objects: 0]
  | o- fileio
.....
. [Storage Objects: 0]
  | o- pscsi
.....
.. [Storage Objects: 0]
  | o- ramdisk
.....
[Storage Objects: 0]
  o- iscsi
```

```
.....  
..... [Targets: 0]  
o- loopback  
.....  
..... [Targets: 0]  
o- sbp  
.....  
..... [Targets: 0]  
o- vhost  
.....  
..... [Targets: 0]
```

In der Targetcli angekommen

```
cd /backstores/block/  
/backstores/block> create name=testvm.deepdoc.at dev=/dev/zvol/v-  
machines/testvm.deepdoc.at
```

Mit `ls` können wir zwischendurch immer unsere Angaben ansehen:

```
/backstores/block> ls  
o- block  
.....  
..... [Storage Objects: 1]  
o- testvm.deepdoc.at ..... [/dev/zvol/v-  
machines/testvm.deepdoc.at (50.0GiB) write-thru deactivated]
```

Nun weiter mit der Config:

```
cd /iscsi/  
# wir können hier zuerst nur "create" ausführen, das legt automatisch einen  
Namen mit serial an. Danach löschen wir den wieder und legen unseren  
eigenen an, mit dem Hinweis auf die VM.  
/iscsi/> create wwn=iqn.2003-01.org.linux-  
iscsi.testvm.deepdoc.at:sn.8217a0ccccfb
```

Das ganze legt auch das Portal mit an.

```
/iscsi> ls  
o- iscsi  
.....  
..... [Targets: 1]  
o- iqn.2003-01.org.linux-iscsi.testvm.deepdoc.at:sn.8217a0ccccfb  
..... [TPGs: 1]  
o- tpg1  
.....  
. [no-gen-acls, no-auth]  
o- acls
```

```

.....
..... [ACLs: 0]
      o- luns
.....
..... [LUNs: 0]
      o- portals
.....
..... [Portals: 1]
      o- 0.0.0.0:3260
.....
..... [OK]

```

Hier wird angegeben auf welcher IP-Adresse des Servers das Target freigegeben wird. Bei nur einer IP-Adresse wird diese standardmäßig ausgewählt, ansonsten kann sie explizit angegeben werden. Der Port kann ebenso angepasst werden.

```

/iscsi> cd iqn.2003-01.org.linux-
iscsi.testvm.deepdoc.at:sn.8217a0cccfd/tpg1/portals/
/iscsi/iqn.20.../tpg1/portals> delete ip_address=0.0.0.0 ip_port=3260
/iscsi/iqn.20.../tpg1/portals> create
ip_address=2001:470:1e2e:537:ec4:7afe:fefa:c839

```

Konfigurieren der LUN's

- Eine LUN bezeichnet eine Logical Unit (LU).
- Oft wird aber auch der Begriff Logical Unit Number verwendet.
- Sie wird zur Zuordnung für die Ansteuerung von Geräten im SCSI-Bus verwendet.
- Es können je Target ein oder mehrere LUNs bereitgestellt werden.

```

cd /iscsi/iqn.2003-01.org.linux-
iscsi.testvm.deepdoc.at:sn.8217a0cccfd/tpg1/luns
/iscsi/iqn.20...fbd/tpg1/luns> create /backstores/block/testvm.deepdoc.at

```

ACL's setzen

Hiermit wird gesteuert auf welche Interfaces ein Initiator zugreifen darf. Um hier auch den richtigen Namen zu verwenden, sieht man sich auf dem Client (einer Proxmoxnode) die Konfig an. `cat /etc/iscsi/initiatorname.iscsi` Den Namen dort kann man verwenden oder abändern. Nach dem Ändern muss der Client neu gestartet werden.

```
systemctl restart iscsid.service
```

Dies gilt auch für die Änderung via Webmin.

Man kann an dieser Stelle natürlich auch mehr als eine ACL hinzufügen.

```

cd ../acls/
/iscsi/iqn.20...fbd/tpg1/acls> create

```

```
iqn.1993-08.org.debian:node01.deepdoc.at  
/iscsi/iqn.20...fbd/tpg1/acls> create  
iqn.1993-08.org.debian:node02.deepdoc.at
```

Usere Fertig Config sieht nun so aus:

```
o- /  
.....  
..... [...]  
  o- backstores  
.....  
..... [...]  
    | o- block  
.....  
    .. [Storage Objects: 1]  
      | | o- testvm.deepdoc.at ..... [/dev/zvol/v-  
machines/testvm.deepdoc.at (50.0GiB) write-thru activated]  
      | o- fileio  
.....  
      . [Storage Objects: 0]  
        | o- pscsi  
.....  
      .. [Storage Objects: 0]  
        | o- ramdisk  
.....  
      [Storage Objects: 0]  
        o- iscsi  
.....  
        ..... [Targets: 1]  
          | o- iqn.2003-01.org.linux-iscsi.testvm.deepdoc.at:sn.8217a0cccfbd  
          ..... [TPGs: 1]  
            | o- tpg1  
.....  
          [no-gen-acls, no-auth]  
            | o- acls  
.....  
            ..... [ACLs: 2]  
              | | o- iqn.1993-08.org.debian:node01.deepdoc.at  
              ..... [Mapped LUNs: 1]  
                | | | o- mapped_lun0 .....  
                [lun0 block/testvm.deepdoc.at (rw)]  
                | | o- iqn.1993-08.org.debian:node02.deepdoc.at  
                ..... [Mapped LUNs: 1]  
                  | | o- mapped_lun0 .....  
                  [lun0 block/testvm.deepdoc.at (rw)]  
                  | o- luns  
.....  
            ..... [LUNs: 1]  
              | | o- lun0 ..... [block/testvm.deepdoc.at  
(/dev/zvol/v-machines/testvm.deepdoc.at)]
```

```
|      o- portals
.....
.... [Portals: 1]
|      o- [2001:470:1e2e:537:ec4:7afe:fefa:c839]:3260
..... [OK]
o- loopback
.....
..... [Targets: 0]
o- sbp
.....
..... [Targets: 0]
o-
vhost.....
..... [Targets: 0]
```

Einhängen des Targets in Proxmox

Das Einhängen des Targets erfolgt über das Webinterface und wird hier nicht weiter behandelt.

Sehrwohl aber das Aushängen. Deaktiviert man ein ISCSI-Target oder löscht es von PVE, dann ist es zwar dort nicht mehr verfügbar, aber am Host definitiv noch eingehängt. Man muss also die Verbindung trennen, und wenn es nach dem nächsten Neustart des Dienstes oder des Hosts keine neuen Verbindungsversuche geben soll, auch das Target am PVE-Client löschen. Hierfür gibt es einen [Bugbericht](#), der von PVE aber nicht also solcher anerkannt wird. Laut PVE ist das Aushängen von Freigaben immer Aufgabe des Admins, man kann nicht abschätzen ob das Target nun auch wirklich gelöscht werden soll... Meines Achtens sollte das aber schon der Fall sein.

Wie geht man also vor? (Zitat vom Bugbericht)

What could be down when a user removes an iscsi target:

- 1) iscsiadm -m node -T <target> -u
- 2) iscsiadm -m node -T <target> -o delete
- 3) if /etc/iscsi/send_targets/initiator/ which target belongs to is empty
rm -rf /etc/iscsi/send_targets/initiator/
- 4) remove from pve

From:
<https://deepdoc.at/dokuwiki/> - DEEPDOC.AT - enjoy your brain

Permanent link:
https://deepdoc.at/dokuwiki/doku.php?id=virtualisierung:proxmox_kvm_und_lxc:proxmox_debian_als_iscsi_server_verwenden

Last update: 2019/03/09 23:28

