Building an iSCSI Target

Set up a couple of volumes to be exported

To build an iSCSI target, we need to create some space on the file system which will be exported. This can be a file, which is slow, a device, or a zfs volume. The latter has several advantages in that it is very fast, and can take advantages of zfs by taking snapshots, using checksums to maintain reliability, and using compression. (De)compression will be performed on the iSCSI server itself, so the initiators (clients) will not have their resources utilized.

```
# Set volmode=dev is preferred for iSCSI volumes
# create a zvol with compression on and checksum off. Dangerous
zfs create -V 10G -o volmode=dev -o checksum=off -o compression=on
storage/virtual_1
# create a ZFS Volume named virtual_2 to be exported.
zfs create -V 10G -o volmode=dev storage/virtual_2
# zfs volumes are available under /dev/zvol
zfs list
ls /dev/zvol/storage
```

Create Configuration File

Now, we need the file which defines what our target will have available to the initiators. This is stored in /etc/ctl.conf. Create it with 0600 permissions since it should not be read by anyone but root.

First we define a portal group. This allows us to have the same information for the targets we create below. In this case, we are saying initiators can perform discovery with no authentication, and we are listening on all IPv4 addresses on port 3260 (the default).

Following that, we define one or more targets. The target names, by convention, are

iqn.yyyy-mm.domain:targetname

where domain is reversed (like reverse dns).

I have included some basic information in both targets below, basically allowing any initiator to use the target, then defining a single lun (lun 0 is required). There are many, many other things that can be done here.

ctl.conf

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0:3260
    listen [::]:3260
}
```

```
target iqn.2019-02.com.example:virtual-1 {
        alias "Virtual 1"
    auth-group no-authentication
    portal-group pg0

    lun 0 {
                path /dev/zvol/storage/virtual_1
        }
}

target iqn.2019-02.com.example:virtual-2 {
        alias "Virtual 2"
    auth-group no-authentication
    portal-group pg0

    lun 0 {
                path /dev/zvol/storage/virtual_2
        }
}
```

Start Service

Now that the volumes are created and the configuration file is set up and secure, we are ready to start the iSCSI target for the first time. We'll enable the service at boot, then start it up and get a list of the devices which are available.

```
# enable iSCSI target on boot
echo 'ctld_enable="YES"' >> /etc/rc.conf
# start the service for the first time
service ctld start
# list the devices. Note the -v gives you additional information
ctladm devlist -v
```

Maintenance

You can add/remove/modify targets of the iscsi target with only brief interruptions in the services.

Add new Volume

1. Add new volume to file system
2. Add new *target* stanza in /etc/ctl.conf
3. have ctld reread configuration with **service ctld reload**

Remove existing volume

1. Ensure no initiators are actively accessing the volume
2. Remove *target* stanza from /etc/ctl.conf
3. restart the service with **service ctld reload**

Add space to existing volume

1. Add space to volume with **zfs set volsize=15G storage/virtual_1**
2. Restart the service with **service ctld reload**
3. Depending on how the target is utilized, you may need to perform tasks on initiators

Take a snapshot of a volume

```
zfs snap storage/virtual_1@somelabel
```

creates a snapshot of volume storage/virtual_1 with the label somelabel. It is available at

/dev/zvol/storage/virtual_1@somelabel

To roll back the snapshot, you must first stop access to the volume, then rollback, then re-enable access.

```
# comment out stanza in /etc/ctl.conf that exports virtual-1
service ctld reload
# virtual_1 no longer exported
zfs rollback storage/virtual_1@somelabel
# uncomment stanza for virtual_1
service ctld reload
# original version of virtual_1 is now exported.
```

But be careful doing that on a running system.

References

- https://www.freebsd.org/cgi/man.cgi?query=ctl.conf&sektion=5&manpath=freebsd-release-ports
- https://www.freebsd.org/doc/handbook/network-iscsi.html

From:
https://kb.unixservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://kb.unixservertech.com/unix/iscsi/iscsi_target

Last update: **2026/01/17 03:24**