# Internet Small Computer System Interface (ISCI)

In the **iSCSI** world, you’ve got two types of agents:

* an **iSCSI** **target** provides some storage (here called **server**),
* an**iSCSI** **initiator** uses this available storage (here called **client**).

Creates 2 IQN(iscsi Qualified Name) – One for Server and One for Client.

|  |  |
| --- | --- |
| **SERVER** | **CLIENT** |
| Package: targetcliv | Package: iscsi-utils |
| Service: target | Service: iscsi, iscsid |
| Firewall Port: 3260/tcp |  |
|  | Initiator: /etc/iscsi/initiatorname |

**Tool:** >> target

Most of the **target** configuration is done interactively through the **targetcli** command. This command uses a directory tree to access the different objects.

## Server Side

Install the following packages:

# **yum install -y targetcli**

Activate the **target** service at boot:

# **systemctl enable target**

Start the **target** service:

# **systemctl start target**

**Firewall**

# **firewall-cmd --permanent –add-port=3260/tcp**

# **firewall-cmd reload**

**getent services 3260**

**Create Disk**

# **fdisk /dev/vdb**

**n >> 5GB >> /dev/vdb1**

Execute the targetcli command:

**# targetcli**

**/> ls**

**o- backstores .................................................**

**o- block .......................................................**

**o- fileio............................................................**

**o- psccsi ......................................................**

**o- ramdisk ......................................................**

**o- iscsi .................................................**

**o- loopback .................................................**

Now use an existing logical disk (**/dev/vdb1**) as a block-type backing store for storage object “**data**”

/> cd backstores/block

/backstores/block> create dev=/dev/vdb1 name=data

Create a target

/backstores/block> **cd /iscsi**

/iscsi> **create wwn=iqn.2017-09.com.example.com:server4**

Created target iqn.**2017-09.com.example.com:server4**.

Created TPG 1.

Below **tpg1**, three objects have been defined:

* **acls** (**a**ccess **c**ontrol **l**ists: restrict access to resources),
* **luns** (**l**ogical **u**nit **n**umber: define exported resources),
* **portals** (define ways to reach the exported resources; consist in pairs of IP addresses and ports).

o- tpg1 ........................................ [no-gen-acls, no-auth]

o- acls ................................................... [ACLs: 0]

o- luns ................................................... [LUNs: 0]

o- portals ............................................. [Portals: 1]

Create ACL for client machine (It’s the IQN which clients use to connect).

/iscsi> **cd /iscsi/iqn.2017-09.com.example.com:server4/tpg1/acls**

/iscsi/iqn.20...sk1/tpg1/acls> **wwn=iqn.2017-09.com.example.com:desktop4**

Create a LUN under the target, The LUN should use the previously mentioned backing storage object named “data”

/iscsi/iqn.20 > **cd /iscsi/iqn.2017-09.com.example.com:server4/tpg1/luns**

/iscsi/iqn.20...sk1/tpg1/luns> **create /backstores/block/data**

Created LUN 0.

Create a portal with server IP

/iscsi/iqn.20.. > **cd /iscsi/iqn.2017-09.com.example.com:server4/tpg1/portals**

/iscsi/iqn.20...sk1/tpg1/portals> **create ip\_address=172.25.4.11**

**Save Configuration**

/iscsi/iqn.20...sk1/tpg1/portals> **cd /**

/> **saveconfig**

/> **exit**

# **systemctl restart target**

## Client Side

Install the following packages:

# **yum install -y iscsi\***

Activate the services at boot:

# **systemctl enable isci iscsid**

Start the services:

# **systemctl isci iscsid**

[root@node1 ~]# vi /etc/iscsi/initiatorname.iscsi

**InitiatorName=iqn.2017-09.com.example.com:desktop4**

Discover the target using below command.

[root@node1 ~]# iscsiadm --mode discoverydb --type sendtargets --portal 172.25.4.11 --discovery

[root@node1 ~]# systemctl restart iscsid.service

Login to the discovered target.

[root@node1 ~]# iscsiadm --mode node --targetname **iqn.2017-09.com.example.com:server4** --pprtal 172.25.4.11:3260 --login

Mount

[root@node1 ~]# lsblk

[root@node1 ~]# fdisk /dev/sda

[root@node1 ~]# mkfs.xfs /dev/sda1

[root@node1 ~]# blkid

[root@node1 ~]# mkdir /iscsi

[root@node1 ~]# vim /etc/fstab

UUID=3c... /iscsi xfs defaults,\_netdev 0 0

[root@node1 ~]# mount -a

# NFS

## Server Side

[root@node1 ~]# yum install nfs-utils krb5-workstation

[root@node1 ~]# systemctl enable nfs-server

[root@node1 ~]# systemctl start nfs-server

[root@node1 ~]# firewall-cmd --permanent –add-service=nfs

[root@node1 ~]# firewall-cmd --reload

[root@node1 ~]# mkdir /nfsshare

[root@node1 ~]# vim /etc/exports

/nfsshare \*.example.com(ro)

[root@node1 ~]# exportfs –r

[root@node1 ~]# exportfs

/nfsshare \*.example.com

[root@node1 ~]# mkdir –p /nfssecure/protected

[root@node1 ~]# vim /etc/exports

/nfsecure \*.example.com(rw,sec=krb5p)

[root@node1 ~]# exportfs –r

[root@node1 ~]# exportfs

[root@node1 ~]# lab nfskrb5 setup

[root@node1 ~]# wget –O /etc/krb5.keytab http://.....serverx.keytab

[root@node1 ~]# systemctl enable nfs-secure-server

[root@node1 ~]# systemctl start nfs-secure-server

[root@node1 ~]# systemctl restart sssd

[root@node1 ~]# chown ladpuserx /nfssecure/protected

[root@node1 ~]# vim /etc/sysconfig/nfs

RPGNFSDRAGS=”-V 4.2” ######Line 13

[root@node1 ~]# systemctl restart nfs-server nfs-secure-server

## Client Side

[root@desktop1 ~]# yum install nfs-utils krb5-workstation

[root@desktop1 ~]# systemctl enable nfs-secure

[root@desktop1 ~]# systemctl start nfs-secure

[root@desktop1 ~]# lab nfskrb5 setup

[root@desktop1 ~]# wget –O /etc/krb5.keytab http://.....desktopx.keytab

[root@desktop1 ~]# mkdir /public

[root@desktop1 ~]# mkdir /secure

[root@desktop1 ~]# vim /etc/fstab

ServerX.example.com:/nfsshare /public nfs defaults 0 0

ServerX.example.com:/nfssecure /secure nfs defaults,sec=krb5p,vers=4.2 0 0

[root@desktop1 ~]# mount -a

[root@desktop1 ~]# systemctl restart nfs-secure.service

# Samba

[root@node1 ~]# yum install samba\*

[root@node1 ~]# systemctl enable smb

[root@node1 ~]# systemctl start smb

[root@node1 ~]# firewall-cmd --permanent –add-service=samba

[root@node1 ~]# firewall-cmd --reload

[root@node1 ~]# user add Stallmen

[root@node1 ~]# smbpasswd –as Stallmen

[root@node1 ~]# smbpasswd –e Stallmen

[root@node1 ~]# mkdir /sambadir

[root@node1 ~]# chmod 2775 /sambadir

[root@node1 ~]# chggrp Stallmen /sambadir

[root@node1 ~]# user add Watson

[root@node1 ~]# smbpasswd –as Watson

[root@node1 ~]# smbpasswd –e Watson

[root@node1 ~]# user add Eaton

[root@node1 ~]# smbpasswd –as Eaton

[root@node1 ~]# smbpasswd –e Eaton

[root@node1 ~]# mkdir /opstack

[root@node1 ~]# chmod 2775 /opstack

[root@node1 ~]# chggrp Watson /opstack

[root@node1 ~]# vim /etc/samba/smb.conf

Workgroup = TESTGROUP

hosts allow=172.25.

[data]

Path=/sambadir

Readonly=yes

Valid user = Stallmen

Browseable = yes

[cluster]

Path=/opstack

Valid users=Watson,Eaton

Write list = @Watson

[root@node1 ~]# semanage fcontext –a –t samba\_share\_t ‘/sambadir(/.?’

[root@node1 ~]# restorecon –v /sambadir

[root@node1 ~]# semanage fcontext –a –t samba\_share\_t ‘/opstack(/.?’

[root@node1 ~]# restorecon –v /opstack

[root@node1 ~]# systemctl restart smb

# Client Side

[root@node1 ~]# yum install cifs-utils samba\*

[root@node1 ~]# smbclient –L //server4.example.com –U Stallmen

[root@node1 ~]# mkdir /mnt/smbspace

[root@node1 ~]# vim /cred.txt

Username=Watson

Password=SaniTago

[root@node1 ~]# vim /etc/fstab

//server4.example.com/cluster /mnt/smbspace cisf defaults,credentials=/cred.txt,multiuser,sec=ntmlssp 0 0

[root@node1 ~]# mount –a

[root@node1 ~]# cisfcreds add server4.example.com –u Watson

# Dynamic WebPage

[root@node1 ~]# yum http\* mod\*

[root@node1 ~]# systemctl enable httpd

[root@node1 ~]# systemctl start httpd

[root@node1 ~]# firewall-cmd --permanent –add-service=http

[root@node1 ~]# firewall-cmd --permanent –add-service=https

[root@node1 ~]# firewall-cmd --reload

[root@node1 ~]# mkdir /var/ww/scripts

[root@node1 ~]# cd /var/www/scripts

[root@node1 ~]# wget http://....../webapp.wsgi

[root@node1 ~]# semanage port –a –t http\_port\_t –p 8961

[root@node1 ~]# firewall-cmd --permanent –add-port=8961/tcp

[root@node1 ~]# firewall-cmd --reload

[root@node1 ~]# vim /etc/httpd/conf.d/dynamic.conf

<VirtualHost \*:8961>

DocumentRoot /var/www/scripts

ServerName wsgiX.example.com

WSGIScriptAlias / /var/www/scripts/webapp.wsgi

</VirtualHost>

Listen 8961

<Directory /var/www/scripts>

Required all granted

</Directory>

[root@node1 ~]# semanage fcontext –a –t http\_sys\_script\_exec\_t ‘/var/www/scripts/webapp.wsgi(/.\*)?’

[root@node1 ~]# restorecon –v /var/www/scripts/webapp.wsgi

[root@node1 ~]# systemctl restart httpd

[root@node1 ~]# firefox http://wsgi.example.com:8961