

Лабораторная работа № 13

Тема: Сетевые сервисы на Linux. iSCSI.

Цель работы: Изучение и настройка iSCSI (Internet Small Computer System Interface) на Linux для предоставления сетевого доступа к блочным устройствам.

Необходимое оборудование и программное обеспечение: Виртуальные машины под управлением Linux (CentOS, Ubuntu или др.).

1. Создание целевого устройства (iSCSI таргета) на linux с помощью targetcli.

Установка пакета targetcli:

```
[root@r2 ~]# yum install targetcli
```

Проверим имеющиеся блочные устройства, sda – системный диск, sdb - дополнительный

```
[root@r2 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda           8:0    0   40G  0 disk
├─sda1         8:1    0    1G  0 part /boot
└─sda2         8:2    0   39G  0 part
   ├─centos-root 253:0    0   37G  0 lvm  /
   └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb           8:16    0    8G  0 disk
sr0          11:0    1 1024M  0 rom
```

Подготовка диска. Дополнительный диск нужно отформатировать, для этого установим lvm2 (программа для управления логическими томами в операционных системах Linux):

```
[root@r2 ~]# yum install -y lvm2
Загружены модули: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirror.ps.kz
 * extras: mirror.ps.kz
 * updates: mirror.ps.kz
Пакет 7:lvm2-2.02.187-6.el7_9.5.x86_64 уже установлен
Выполнять нечего
```

Создание физического том (PV) на дополнительном жестком диске:

```
[root@r2 ~]# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created.
```

Создание группы томов (centos-target) и добавление физического тома в нее:

```
[root@r2 ~]# vgcreate centos-target /dev/sdb
Volume group "centos-target" successfully created
```

Создание логического тома (backstore) внутри группы томов(centos-target):

```
[root@r2 ~]# lvcreate -L 8G -n backstore centos-target
Volume group "centos-target" has insufficient free space (2047 extents): 2048 required.
```

Не хватает одного блока (4МБ). Укажем размер в мегабайтах (8192-4=8188)

```
[root@r2 ~]# lvcreate -L 8188M -n backstore centos-target
Logical volume "backstore" created.
```

или можно так:

```
lvcreate -L 6G -n backstore centos-target
```

Проверяем:

```
[root@r2 ~]# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0   40G  0 disk
├─sda1                              8:1    0    1G  0 part /boot
└─sda2                              8:2    0   39G  0 part
   ├─centos-root                    253:0    0   37G  0 lvm  /
   └─centos-swap                    253:1    0    2G  0 lvm  [SWAP]
sdb                                  8:16    0    8G  0 disk
└─centos--target-backstore          253:2    0    8G  0 lvm
sr0                                  11:0    1 1024M  0 rom
```

backstore - это абстрактный слой программного обеспечения, который отвечает за хранение данных и предоставляет интерфейс для доступа к ним через протокол iSCSI.

Backstore выполняет преобразование запросов iSCSI в операции ввода-вывода на устройстве хранения. Он может использовать различные механизмы для хранения данных, такие как файлы, блочные устройства, облачные хранилища и другие.

Запуск targetcli:

```
[root@r2 ~]# targetcli
Warning: Could not load preferences file /root/.targetcli/prefs.bin.
targetcli shell version 2.1.53
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/>
```

Список директорий – ls

```
/> 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]
/>
```

Создаем блочный backstore с именем "disk1", который будет использовать **/dev/centos-target/backstore** в качестве устройства хранения данных:

```
/> cd backstores/block
/backstores/block> create disk1 /dev/centos-target/backstore
Created block storage object disk1 using /dev/centos-target/backstore.
/backstores/block>
```

```

/backstores/block> ..
/backstores> ..
/> ls
o- / ..... [....]
  o- backstores ..... [....]
    | o- block ..... [Storage Objects: 1]
    | | o- disk1 .. [/dev/centos-target/backstore (8.0GiB) write-thru deactivated]
    | |   o- alua ..... [ALUA Groups: 1]
    | |     o- default_tg_pt_gp ..... [ALUA state: Active/optimized]
    | o- fileio ..... [Storage Objects: 0]
    | o- pscsi ..... [Storage Objects: 0]
    | o- ramdisk ..... [Storage Objects: 0]
  o- iscsi ..... [Targets: 0]
  o- loopback ..... [Targets: 0]
/> █

```

IQN (iSCSI Qualified Name) представляет собой уникальное имя, используемое для идентификации iSCSI-объекта в сети. Оно должно быть уникальным для каждого iSCSI-объекта, чтобы избежать конфликтов и позволить идентифицировать объект в сети.

Обычно IQN выглядит как строка в формате:

iqn.<год-месяц>.<обратный домен>:<имя_объекта>

Создание нового iSCSI-объекта (iqn нашего таргета):

```

/> cd iscsi
/iscsi> ls
o- iscsi ..... [Targets: 0]
/iscsi> create iqn.2024-03.com.example:server1-disk1
Created target iqn.2024-03.com.example:server1-disk1.
Created TPG 1.
Global pref auto_add_default_portal=true
Created default portal listening on all IPs (0.0.0.0), port 3260.
/iscsi> █

```

```

/iscsi> ls
o- iscsi ..... [Targets: 1]
  o- iqn.2024-03.com.example:server1-disk1 ..... [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]
/iscsi> █

```

Настройка acl. Переходим в пространство имен iSCSI, далее в конкретный iSCSI-объект, для которого нужно настроить ACL. Указываем имя клиента, которому разрешен доступ, например, iqn.2024.com.example:client1

```

/iscsi> cd iqn.2024-03.com.example:server1-disk1/tpg1/acls
/iscsi/iqn.20...sk1/tpg1/acls> create iqn.2024-03.com.example:client1
Created Node ACL for iqn.2024-03.com.example:client1
/iscsi/iqn.20...sk1/tpg1/acls> █

```

LUN (Logical Unit Number) - это логический блок данных или виртуальный диск, предоставляемый хранилищем данных (например, хранилищем SAN или NAS) посредством протокола iSCSI.

Каждый LUN идентифицируется уникальным идентификатором LUN (LUN ID) и может быть доступен для подключения к удаленным устройствам через сеть. LUN является абстракцией физических накопителей и разделов, предоставляющих пространство для хранения данных удаленным устройствам, таким как серверы или виртуальные машины.

Настройка лунов (создание логического диска, который будет виден как блочное устройство у клиента):

```
/iscsi/iqn.20...sk1/tpg1/acls> cd ..  
/iscsi/iqn.20...r1-disk1/tpg1> cd luns  
/iscsi/iqn.20...sk1/tpg1/luns>  
/ @last bookmarks cd create delete exit get  
help ls pwd refresh set status  
/iscsi/iqn.20...sk1/tpg1/luns> █
```

```
/iscsi/iqn.20...sk1/tpg1/acls> cd ..  
/iscsi/iqn.20...r1-disk1/tpg1> ls  
o- tpg1 ..... [no-gen-acls, no-auth]  
  o- acls ..... [ACLs: 1]  
    | o- iqn.2024-03.com.example:client1 ..... [Mapped LUNs: 0]  
    o- luns ..... [LUNs: 0]  
    o- portals ..... [Portals: 1]  
      o- 0.0.0.0:3260 ..... [OK]  
/iscsi/iqn.20...r1-disk1/tpg1> cd luns  
/iscsi/iqn.20...sk1/tpg1/luns> create /backstores/block/disk1  
Created LUN 0.  
Created LUN 0->0 mapping in node ACL iqn.2024-03.com.example:client1  
/iscsi/iqn.20...sk1/tpg1/luns> █
```

```
/iscsi/iqn.20...sk1/tpg1/luns> cd /  
/> ls  
o- / ..... [...]  
  o- backstores ..... [...]  
    | o- block ..... [Storage Objects: 1]  
    |   | o- disk1 .... [/dev/centos-target/backstore (8.0GiB) write-thru activated]  
    |   |   o- alua ..... [ALUA Groups: 1]  
    |   |     o- default_tg_pt_gp ..... [ALUA state: Active/optimized]  
    | o- fileio ..... [Storage Objects: 0]  
    | o- pscsi ..... [Storage Objects: 0]  
    | o- ramdisk ..... [Storage Objects: 0]  
  o- iscsi ..... [Targets: 1]  
    | o- iqn.2024-03.com.example:server1-disk1 ..... [TPGs: 1]  
    |   o- tpg1 ..... [no-gen-acls, no-auth]  
    |     o- acls ..... [ACLs: 1]  
    |       | o- iqn.2024-03.com.example:client1 ..... [Mapped LUNs: 1]  
    |       |   o- mapped_lun0 ..... [lun0 block/disk1 (rw)]  
    |       o- luns ..... [LUNs: 1]  
    |         | o- lun0 [block/disk1 (/dev/centos-target/backstore) (default_tg_pt_gp  
    |       )]  
    |   o- portals ..... [Portals: 1]  
    |     o- 0.0.0.0:3260 ..... [OK]  
  o- loopback ..... [Targets: 0]  
/> █
```

Настройки таргета завершены, выходим:

```
/> exit  
Global pref auto_save_on_exit=true  
Configuration saved to /etc/target/saveconfig.json  
[root@r2 ~]# █
```

Запускаем таргет и добавляем в автозагрузки:

```
[root@r2 ~]# systemctl start target
[root@r2 ~]# systemctl enable target
Created symlink from /etc/systemd/system/multi-user.target.wants/target.service
to /usr/lib/systemd/system/target.service.
[root@r2 ~]#
```

Смотрим статус, все хорошо:

```
[root@r2 ~]# systemctl status target
● target.service - Restore LIO kernel target configuration
   Loaded: loaded (/usr/lib/systemd/system/target.service; enabled; vendor prese
t: disabled)
   Active: active (exited) since Пн 2024-03-18 20:33:57 +05; 49s ago
   Main PID: 1667 (code=exited, status=0/SUCCESS)

map 18 20:33:57 r2 systemd[1]: Starting Restore LIO kernel target config.....
map 18 20:33:57 r2 systemd[1]: Started Restore LIO kernel target configu...n.
Hint: Some lines were ellipsized, use -l to show in full.
[root@r2 ~]#
```

Что бы посмотреть открытые порты установим net-tools:

```
[root@r2 ~]# yum install -y net-tools
Загружены модули: fastestmirror
Loading mirror speeds from cached hostfile
```

Смотрим список открытых сетевых соединений и прослушивающих портов в системе. Порт 3260 «слушается» на всех адресах:

```
[root@r2 ~]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:2601          0.0.0.0:*                LISTEN      721/zebra
tcp        0      0 127.0.0.1:2604          0.0.0.0:*                LISTEN      736/ospfd
tcp        0      0 0.0.0.0:111             0.0.0.0:*                LISTEN      676/rpcbind
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1177/sshd
tcp        0      0 127.0.0.1:25            0.0.0.0:*                LISTEN      1427/master
tcp        0      0 0.0.0.0:3260            0.0.0.0:*                LISTEN      -
tcp6       0      0 :::1:2606               :::*                    LISTEN      748/ospf6d
tcp6       0      0 :::111                  :::*                    LISTEN      676/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      1177/sshd
tcp6       0      0 :::1:25                  :::*                    LISTEN      1427/master
udp        0      0 0.0.0.0:850             0.0.0.0:*                *          676/rpcbind
udp        0      0 0.0.0.0:67              0.0.0.0:*                *          1178/dhcpd
udp        0      0 0.0.0.0:68              0.0.0.0:*                *          893/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                *          676/rpcbind
udp6       0      0 :::850                  :::*                    *          676/rpcbind
udp6       0      0 :::111                  :::*                    *          676/rpcbind
[root@r2 ~]#
```

Файвоулл может блокировать соединения к таргету, добавим новое правило (разрешение входящих TCP-соединений на порт 3260):

```
[root@r2 ~]# firewall-cmd --add-port=3260/tcp --permanent
success
[root@r2 ~]# firewall-cmd --reload
success
```

Проверяем, порт разрешен:

```
[root@r2 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3 enp0s8 enp0s9
  sources:
  services: dhcpv6-client ssh
  ports: 3260/tcp
  protocols: ospf
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
[root@r2 ~]#
```

Настройка клиента (Инициатора) (r1)

iscsi-initiator-utils - это набор утилит и библиотек для работы с iSCSI в Linux. Они предоставляют средства для инициации (инициатор) iSCSI-соединений, которые позволяют Linux-системе подключаться к удаленным iSCSI-устройствам (iSCSI target), таким как хранилища данных.

Устанавливаем нужную утилиту:

```
[root@r1 ~]# yum install -y iscsi-initiator-utils
Загружены модули: fastestmirror
Loading mirror speeds from cached hostfile
```

Редактируем файл initiatorname.iscsi, указываем имя, которое добавили в acl при настройке таргета:

```
[root@r1 ~]# nano /etc/iscsi/initiatorname.iscsi
```

```
GNU nano 2.3.1      файл: /etc/iscsi/initiatorname.iscsi

InitiatorName=iqn.2024-03.com.example:client1
```

Запускаем службу, добавляем ее в автозапуск и проверяем статус:

```
[user@localhost ~]# sudo systemctl start iscsi
[user@localhost ~]# sudo systemctl enable iscsi
[user@localhost ~]# sudo systemctl status iscsi
● iscsi.service - Login and scanning of iSCSI devices
   Loaded: loaded (/usr/lib/systemd/systemd/iscsi.service; indirect; preset: enabled)
   Active: active (exited) since Sun 2024-02-25 15:36:26 +06; 19s ago
     Docs: man:iscsiadm(8)
           man:iscsid(8)
  Main PID: 1587 (code=exited, status=21)
    CPU: 9ms

Feb 25 15:36:26 localhost.localdomain systemd[1]: Starting Login and scanning of iSCSI devices...
Feb 25 15:36:26 localhost.localdomain iscsiadm[1587]: iscsiadm: No records found
Feb 25 15:36:26 localhost.localdomain systemd[1]: Finished Login and scanning of iSCSI devices.
[user@localhost ~]#
```

iscsiadm - это утилита командной строки в Linux, предназначенная для управления iSCSI-инициатором и настройки iSCSI-соединений. Она позволяет администраторам настраивать и управлять iSCSI-инициатором, управлять сеансами и соединениями iSCSI, а также выполнять другие административные задачи.

Вот некоторые основные команды и параметры **iscsiadm**:

1. **iscsiadm -m discovery -t st -p <target_IP>**: Выполняет процедуру обнаружения iSCSI-целей (targets) на указанном IP-адресе.
2. **iscsiadm -m node -T <target_name> -p <target_IP> --login**: Подключается к указанной iSCSI-цели, предоставляя ее имя и IP-адрес, и выполняет процедуру входа в систему.
3. **iscsiadm -m node -T <target_name> -p <target_IP> --logout**: Отключается от указанной iSCSI-цели.
4. **iscsiadm -m session**: Показывает список текущих iSCSI-сеансов.
5. **iscsiadm -m session -P 3**: Показывает подробную информацию о текущих iSCSI-сеансах, включая параметры конфигурации и статус подключения.
6. **iscsiadm -m node**: Показывает список iSCSI-устройств (инициаторов), к которым система может подключаться.
7. **iscsiadm -m iface**: Показывает список доступных сетевых интерфейсов, которые могут использоваться для iSCSI.

Нужно обнаружить доступные таргеты, смотрим примеры в мануале:

```
[user@localhost ~]$ man iscsiadm
```

EXAMPLES

Discover targets at a given IP address:

```
sh# iscsiadm --mode discoverydb --type sendtargets --portal 192.168.1.10 --discovery
```

Login, must use a node record id found by the discovery:

```
sh# iscsiadm --mode node --targetname iqn.2001-05.com.doe:test --portal 192.168.1.1:3260 --login
```

Проводим поиск по образцу. Таргет найден:

```
[root@r1 ~]# iscsiadm -m discoverydb -t st -p 2.2.2.2 -D 2.2.2.2:3260,1 iqn.2024-03.com.example:server1-disk1
[root@r1 ~]#
```

Логинимся так же по образцу, успешно:

```
[root@r1 ~]# iscsiadm -m node -T iqn.2024-03.com.example:server1-disk1 -l
Logging in to [iface: default, target: iqn.2024-03.com.example:server1-disk1, portal: 2.2.2.2,3260] (multiple)
Login to [iface: default, target: iqn.2024-03.com.example:server1-disk1, portal: 2.2.2.2,3260] successful.
[root@r1 ~]#
```

Проверяем список доступных блочных устройств. Добавился новый диск sdb:

```
[root@r1 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   40G  0 disk
├─sda1       8:1    0    1G  0 part /boot
└─sda2       8:2    0   39G  0 part
   ├─centos-root 253:0    0   37G  0 lvm  /
   └─centos-swap 253:1    0    2G  0 lvm  [SWAP]
sdb          8:16    0    8G  0 disk
sr0         11:0    1 1024M  0 rom
[root@r1 ~]#
```

Нужно его подготовить, устанавливаем dosfstools (пакет программного обеспечения для работы с файловыми системами в Linux.)

```
[root@r1 ~]# yum install dosfstools -y
```

Форматируем диск в ext4:

```
[root@r1 ~]# mkfs.ext4 /dev/sdb
mke2fs 1.42.9 (28-Dec-2013)
/dev/sdb is entire device, not just one partition!
Proceed anyway? (y,n) y
Filesystem label=
```

Создадим подкаталог newdisk в каталоге /mnt и смонтируем в него диск:

```
[root@r1 ~]# mkdir /mnt/newdisk
[root@r1 ~]# mount /dev/sdb /mnt/newdisk
```

Перейдем в диск, создадим файл. Проверяем, что с файлом можно работать как обычно:

```
[root@r1 newdisk]# nano test-iscsi.txt
```

```
GNU nano 2.3.1          файл: test-iscsi.txt
```

```
hello!
test iscsi!!!
```

```
[root@r1 newdisk]# ls -l
итого 20
drwx-----. 2 root root 16384 мар 17 01:11 lost+found
-rw-r--r--. 1 root root    21 мар 17 01:14 test-iscsi.txt
[root@r1 newdisk]# cat test-iscsi.txt
hello!
test iscsi!!!
[root@r1 newdisk]#
```

Настройка iscsi завершена.