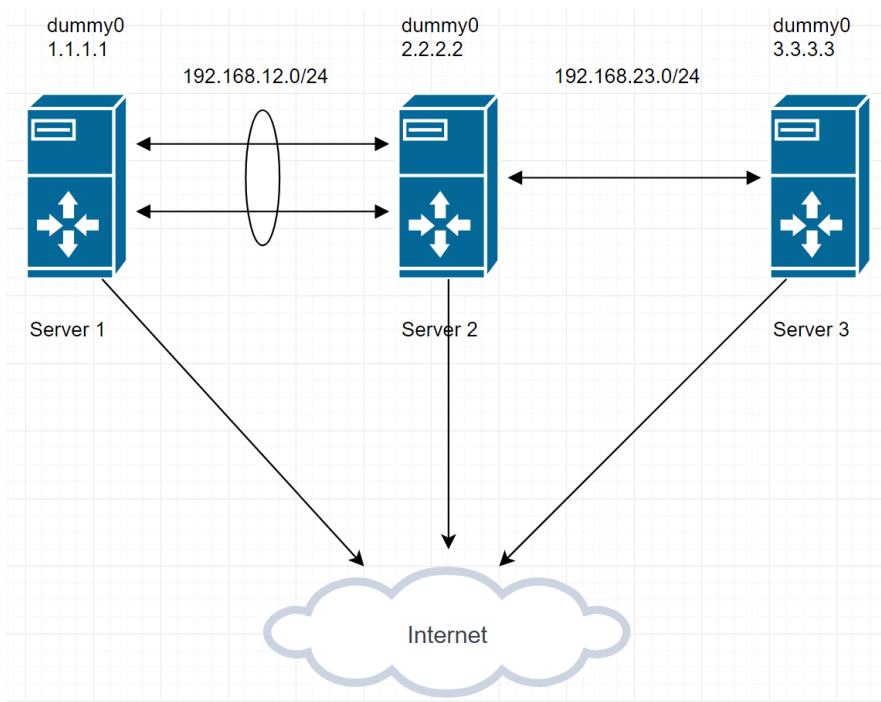


Для создания топологии, необходимой для выполнения ДЗ:



Созданы три виртуальные машины в VirtualBox с CentOS7 minimal:  
R1 (Server 1): 3 сетевых адаптера, первый через мост, два оставшихся - внутренняя сеть.

R1 (Server 1)  
Выключена

R2 (Server 2)  
Выключена

R3 (Server 3)  
Выключена

Запись:Выключена

Носители

Контроллер: IDE  
Вторичный мастер IDE: [Оптический привод] Пусто  
Контроллер: SATA  
SATA порт 0: R1 (Server 1).vdi (Обычный, 5,00 ГБ)

Аудио

Аудиодрайвер: Windows DirectSound  
Аудиоконтроллер: ICH AC97

Сеть

Адаптер 1: Intel PRO/1000 MT Desktop (Сетевой мост, 'Intel(R) Ethernet Connection (2) I219-V')  
Адаптер 2: Intel PRO/1000 MT Desktop (Внутренняя сеть, 'intnet')  
Адаптер 3: Intel PRO/1000 MT Desktop (Внутренняя сеть, 'intnet')

R2 (Server 2): 3 сетевых адаптера, первый через мост, два оставшихся - внутренняя сеть.

R1 (Server 1)  
Выключена

R2 (Server 2)  
Выключена

R3 (Server 3)  
Выключена

Запись:Выключена

Носители

Контроллер: IDE  
Вторичный мастер IDE: [Оптический привод] CentOS-7-x86\_64-Minimal-2009.iso (973,00 МБ)  
Контроллер: SATA  
SATA порт 0: R2 (Server 2).vdi (Обычный, 5,00 ГБ)

Аудио

Аудиодрайвер: Windows DirectSound  
Аудиоконтроллер: ICH AC97




Сеть

Адаптер 1: Intel PRO/1000 MT Desktop (Сетевой мост, 'Intel(R) Ethernet Connection (2) I219-V')  
Адаптер 2: Intel PRO/1000 MT Desktop (Внутренняя сеть, 'intnet')  
Адаптер 3: Intel PRO/1000 MT Desktop (Внутренняя сеть, 'intnet')

На момент выполнения этой части домашнего задания машины R2 & R3 - переустанавливались после выполнения "тренировочного задания" по окончанию вебинара №1 и начала вебинара №2. Дополнительные логические диски и прочее будут прикручиваться последовательно при выполнении заданий.

На все сервера на домашнем роутере были выставлены статические IP для дальнейшего упрощения работы, наверное.

R3 (Server 3): 2 сетевых адаптера, первый через мост, оставшийся - внутренняя сеть.

 R1 (Server 1) Выключена	Сервер удалённого дисплея: Выключен Запись: Выключена
 R2 (Server 2) Выключена	Носители Контроллер: IDE Вторичный мастер IDE: [Оптический привод] CentOS-7-x86_64-Minimal-2009.iso (973,00 МБ) Контроллер: SATA SATA порт 0: R3 (Server 3).vdi (Обычный, 5,00 ГБ)
 R3 (Server 3) Выключена	Аудио Аудиодрайвер: Windows DirectSound Аудиоконтроллер: ICH AC97
	Сеть Адаптер 1: Intel PRO/1000 MT Desktop (Сетевой мост, 'Intel(R) Ethernet Connection (2) I219-V') Адаптер 2: Intel PRO/1000 MT Desktop (Внутренняя сеть, 'intnet')

После установки и первого запуска всех трех машин, были присвоены имена всем трём

```
#hostnamectl set-hostname Server(1..3)
```

Так же установлена на все три - target и initiator-utils

```
#yum install targetcli
```

```
#yum install iscsi-initiator-utils
```

Статистика перед выполнением задания:

R1: enp0s3 - 192.168.1.191/24 - для подключения по ssh  
enp0s8 - остаётся пустым  
enp0s9 - будет назначен по заданию 192.168.12.1/24

R2: enp0s3 - 192.168.1.192/24 - для подключения по ssh  
enp0s8 - будет назначен по заданию 192.168.23.2/24  
enp0s9 - будет назначен по заданию 192.168.12.2/24

R3: enp0s3 - 192.168.1.193/24 - для подключения по ssh  
enp0s8 - будет назначен по заданию 192.168.12.3/24

Настройка enp0s9 для сервера R1, файл конфига храниться /etc/sysconfig/network-scripts/ для данного интерфейса ifcfg-enp0s9

TYPE=Ethernet

PROXY\_METHOD=none

BROWSER\_ONLY=no

BOOTPROTO=static /смена на static

DEFROUTE=yes

IPV4\_FAILURE\_FATAL=no

IPV6INIT=yes

IPV6\_AUTOCONF=yes

IPV6\_DEFROUTE=yes

IPV6\_FAILURE\_FATAL=no

IPV6\_ADDR\_GEN\_MODE=stable-privacy

NAME=enp0s9

UUID=6d60bdc2-3cab-4f73-996b-4fda6cb65792

DEVICE=enp0s9

IPADDR=192.168.12.1 /добавление строчки IPADDR

NETMASK=255.255.255.0 /добавление маски сети

ONBOOT=yes /включение при загрузке

```
#systemctl restart network /ребут нетворка для применения изменений
```

```
#ip a /проверить изменения, получилось.
```

Аналогичные изменения применить для остальных серверов и адаптеров.

Далее установка FRR для CentOS 7. <https://rpm.frrouting.org/> - инструкция по установке.

Устанавливается на все 3 сервера. После установки - настройка сетей.

```
#vtysh
```

Для сервера R1

```
server1# conf terminal
server1(config)# router ospf
server1(config-router)# network 192.168.12.0/24 area 0 /после не забыть сохранить.
#sh running-config
!
frr version 8.3
frr defaults traditional
hostname server1
log syslog informational
no ip forwarding
no ipv6 forwarding
!
router ospf
 network 192.168.12.0/24 area 0
exit
!
end /анонсирование подсети 192.168.12.0/24
```

Для сервера R2

```
server2# sh ip ro
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, A - Babel, F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup
       t - trapped, o - offload failure
```

```
K>* 0.0.0.0/0 [0/100] via 192.168.1.1, enp0s3, 00:13:01
C>* 192.168.1.0/24 is directly connected, enp0s3, 00:13:01
C>* 192.168.12.0/24 is directly connected, enp0s9, 00:13:01
C>* 192.168.23.0/24 is directly connected, enp0s8, 00:13:01
```

```
server2# conf t
server2(config)# router ospf
server2(config-router)# network 192.168.12.0/24 area 0
server2(config-router)# network 192.168.23.0/24 area 0 /не забыть сохранить настройки
```

```
server2# sh running-config
Building configuration...
```

Current configuration:

```
!
frr version 8.3
frr defaults traditional
hostname server2
log syslog informational
no ip forwarding
no ipv6 forwarding
!
router ospf
 network 192.168.12.0/24 area 0
 network 192.168.23.0/24 area 0
```

```
exit
!  
end
```

Для сервера R3:

```
server3# sh ip ro  
Codes: K - kernel route, C - connected, S - static, R - RIP,  
O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,  
T - Table, A - Babel, F - PBR, f - OpenFabric,  
> - selected route, * - FIB route, q - queued, r - rejected, b - backup  
t - trapped, o - offload failure
```

```
K>* 0.0.0.0/0 [0/100] via 192.168.1.1, enp0s3, 00:00:07  
C>* 192.168.1.0/24 is directly connected, enp0s3, 00:00:07  
C>* 192.168.23.0/24 is directly connected, enp0s8, 00:00:07
```

```
server3# conf t  
server3(config)# router ospf  
server3(config-router)# network 192.168.23.0/24 area 0  
server3(config-router)# ex  
server3(config)# ex  
server3# w /не забываем сохранять F5 не работает :(  
Note: this version of vtysh never writes vtysh.conf  
Building Configuration...  
Integrated configuration saved to /etc/frr/frr.conf  
[OK]
```

```
server3# sh ru  
Building configuration...
```

Current configuration:

```
!  
frr version 8.3  
frr defaults traditional  
hostname server3  
log syslog informational  
no ip forwarding  
no ipv6 forwarding  
!  
router ospf  
 network 192.168.23.0/24 area 0  
exit  
!  
end
```

Далее проверка пингом с 192.168.12.1 до адресов 192.168.23.(2..3). Аналогично второй половине вебинара №2 для прохождения пинга мешают firewalld и ipforward.

Для первого теста прохождения пакетов были отключены на всех трех машинах firewalld.

На "среднем" сервере R2 в файле **/etc/sysctl.d/99-sysctl.conf** добавляется строка

**net.ipv4.ip\_forward=1**, что позволит при перезагрузке автоматически включать проброс.

Далее необходимо настроить firewalld, чтобы при перезагрузке автоматически был доступен обмен tcp пакетов.

При перезапуске серверов R1 R2 R3 недоступность пинга с R1 на R3 и наоборот недоступно при перезапуске по следующим причинам:

- нет запуска при перезагрузке `frr`:

**`systemctl enable frr`** /создает правило загрузки `frr` при запуске системы, делается на всех серверах.

- загрузка `firewalld` при запуске системы:

Необходимо настроить правила `firewalld` в которых разрешено прохождение пакетов, но с `firewalld` сложнее разбираться в меру незнакомого дистрибутива и незнакомого синтаксиса, поэтому отключу `systemctl disable firewalld` и буду использовать `iptables` как более знакомая для меня утилита.

Для перманентного применения правил `iptables` в конце настройки необходимо будет настроить автоматическую загрузку правил.

```
iptables -A INPUT -p tcp --dport=22 -j ACCEPT /команда разрешающая работу по SSH
```

```
iptables -A INPUT -i lo -j ACCEPT
```

```
iptables -A OUTPUT -o lo -j ACCEPT /две команды разрешающие межпроцессорный обмен и обмен данных через loopback для работы внутренних сервисов сервера.
```

```
iptables -A INPUT -p icmp -j ACCEPT
```

```
iptables -A OUTPUT -p icmp -j ACCEPT /две команды разрешающие обмен данными ICMP.
```

```
iptables -A OUTPUT -p tcp --sport 32768:61000 -j ACCEPT
```

```
iptables -A OUTPUT -p udp --sport 32768:61000 -j ACCEPT /разрешение входящих соединений с локальных динамических портов.
```

```
iptables -A INPUT -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -A INPUT -p UDP -m state --state ESTABLISHED,RELATED -j ACCEPT /правила разрешающие возвращающиеся пакета с установленными соединениями.
```

```
iptables -A INPUT -p tcp --dport=80 -j ACCEPT /для подключения через порт 80 протоколами tcp
```

```
iptables -A INPUT -p tcp --dport=443 -j ACCEPT /для подключения через порт 443 протоколами tcp
```

```
iptables -A INPUT -p udp --dport=80 -j ACCEPT /для подключения через порт 80 протоколами udp
```

```
iptables -A INPUT -p udp --dport=443 -j ACCEPT /для подключения через порт 443 протоколами udp
```

```
iptables -P INPUT DROP /назначение основной политики входящих цепочек на DROP
```

Всю эту политику повторить для всех трёх серверов.

Создание `dummy0` интерфейсов на серверах R1 R2 R3 с адресами 1.1.1.1/32 2.2.2.2/32 3.3.3.3/32 соответственно на примере сервера R1:

```
#ip link add dummy0 type dummy
```

```
#ip addr add 2.2.2.2/32 dev dummy0
```

```
#ip link set up dev dummy0
```

Теперь необходимо создать правила для автоматического создания `dummy0` при старте машины:

```
cat > /etc/modules-load.d/dummy.conf
```

```
# Load dummy.ko at boot
```

```
dummy
```

```
cat > /etc/modprobe.d/dummy.conf
```

```
install dummy /sbin/modprobe --ignore-install dummy; /sbin/ip link set name dummy0 dev dummy
```

```
cat > /etc/sysconfig/network-scripts/ifcfg-dummy0
```

```
NAME=dummy0
```

```
DEVICE=dummy0
```

```
MACADDR=00:22:22:ff:ff:ff
```

```
IPADDR=1.1.1.1
```

```
PREFIX=32
```

```
ONBOOT=yes
```

```
TYPE=dummy
```

```
NM_CONTROLLED=no
```

Аналогичную работу проделал на серверах R2 R3, всё заработало, автоматически при перезапуске сервера создает `dummy0` с назначенными IP.

Далее на роутерах необходимо настроить `ospf` для сетей 1.1.1.1/32 2.2.2.2/32 3.3.3.3/32. Подобная задача выполнялась ранее, выполняется аналогично.

Создание dummy0 интерфейсов на серверах R1 R2 R3 с адресами 1.1.1.1/32 2.2.2.2/32 3.3.3.3/32 соответственно на примере сервера R1:

```
#ip link add dummy0 type dummy
#ip addr add 2.2.2.2/32 dev dummy0
#ip link set up dev dummy0
```

Теперь необходимо создать правила для автоматического создания dummy0 при старте машины:

```
cat > /etc/modules-load.d/dummy.conf
# Load dummy.ko at boot
dummy
cat > /etc/modprobe.d/dummy.conf
install dummy /sbin/modprobe --ignore-install dummy; /sbin/ip link set name dummy0 dev dummy
cat > /etc/sysconfig/network-scripts/ifcfg-dummy0
NAME=dummy0
DEVICE=dummy0
MACADDR=00:22:22:ff:ff:ff
IPADDR=1.1.1.1
PREFIX=32
ONBOOT=yes
TYPE=Ethernet
NM_CONTROLLED=no
```

Аналогичную работу проделал на серверах R2 R3, всё заработало, автоматически при перезапуске сервера создает dummy0 с назначенными IP.

Далее на роутерах необходимо настроить ospf для сетей 1.1.1.1/32 2.2.2.2/32 3.3.3.3/32. Подобная задача выполнялась ранее, выполняется аналогично.

Для экспорта созданных папок nfs\_1 и nfs\_2 на сервере R3:

Воспользуюсь помощью найденного в сети "гайда" по NFS:

На сервере-источнике:

yum install nfs-utils nfs-utils-lib rpcbind /установка утилит NFS и RPCBind, в установленных дистрибутивах у меня они были предустановлены.

chkconfig nfs on

chkconfig rpcbind on

service rpcbind start

service nfs start /запуск и добавление в автозагрузку NFS и RPCBind

/192.168.1.193 - сервер источник

/192.168.1.191 - сервер клиент

mkdir -p /nfs\_(1..2)

chmod -R 777 /nfs\_(1..2) /создание директорий и полное разрешение для взаимодействия с ними

Далее необходимо создать разрешения на экспорт для директорий в **etc/exports**

/nfs\_1 192.168.1.191/24(rw,sync,no\_root\_squash,no\_all\_squash)

/nfs\_2 192.168.1.191/24(rw,sync,no\_root\_squash,no\_all\_squash) /доступ к директориям и разрешенным доступам к ним из IP-подсетей, плюс дополнительные правила

# firewall-cmd --permanent --zone=public --add-service=nfs

# firewall-cmd --permanent --zone=public --add-service=mountd

# firewall-cmd --permanent --zone=public --add-service=rpc-bind

# firewall-cmd --reload /разрешающие команды для firewalld и перезапуск firewalld

exportfs -a /сохранение и выполнение файла exports

На сервере-клиенте:

```
yum install nfs-utils nfs-utils-lib  
chkconfig nfs on  
service nfs start  
service nfs status
```

Необходимо создать директории для импорта:

```
mkdir -p /mnt/export/nfs_1  
mkdir -p /mnt/export/nfs_2
```

и монтирую:

```
mount 192.168.1.193:/nfs_1 /mnt/export/nfs_1  
mount 192.168.1.193:/nfs_2 /mnt/export/nfs_2
```

df -h /проверка смонтированных директорий

Для постоянного монтирования после перезагрузки в **/etc/fstab**:

```
192.168.1.193:/nfs_1 /mnt/export/nfs_1 nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0  
192.168.1.193:/nfs_2 /mnt/export/nfs_2 nfs auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0  
mount -fav /проверка примонтированных директорий
```

На сервере-источнике:

В файле **/etc/sysconfig/nfs** раскомментировать:

```
LOCKD_TCPSPORT=32803  
LOCKD_UDPPORT=32769  
MOUNTD_PORT=892  
STATD_PORT=662  
STATD_OUTGOING_PORT=2020
```

Перезапустить:

```
service nfs restart  
service rpcbind restart
```

Следующим делом эти поры потребуется открыть в iptables :

Создаю отдельную цепочку и в нее занесем IP адрес сервера-клиента

```
iptables -N NFS  
iptables -A NFS -s 192.168.1.30 -j ACCEPT  
iptables -A NFS -j RETURN
```

Открываю порты 111 и 2049 (TCP and UDP) и других сервисов для монтирования:

```
iptables -A INPUT -p tcp -m multiport --dport 111,2049,32803,892,662 -j NFS  
iptables -A INPUT -p udp -m multiport --dport 111,2049,32769,892,662 -j NFS
```

Сохраняю

```
service iptables save
```

Далее перезагрузка серверов R1 и R3, проверка, что директории автоматически прилепляются и доступны - пройдена. Всё функционирует, директории доступны, сохраняется доступность записи, удаления на сервере R1, так же как и синхронизация с сервером R3.

Для выполнения задания 12\* на сервере R3 в VB был создан дополнительный накопитель. Проверяется командой lsblk, новый носитель sdb смонтирован на систему.

На сервер R3 устанавливается targetcli.

Первым делом необходимо создать блочное хранилище:

```
[root@server3 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   5G  0 disk
├─sda1       8:1    0    1G  0 part /boot
├─sda2       8:2    0    4G  0 part
│   └─centos-root 253:0    0  3.5G  0 lvm  /
│       └─centos-swap 253:1    0  512M  0 lvm  [SWAP]
sdb          8:16    0   5G  0 disk
sr0         11:0    1 1024M  0 rom
[root@server3 ~]# targetcli
targetcli shell version 2.1.53
Copyright 2011-2013 by Datera, Inc and others.
For help on commands, type 'help'.

/> ls
o- / ..... [Targets: 0]
  o- backstores ..... [Targets: 0]
    | 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]
```

#targetcli

>cd /backstores/block

/backstores/block> create name=TASK12-DISK-5G dev=/dev/sdb

Created block storage object TASK12-DISK-5G using /dev/sdb.

/backstores/block> cd ..

/backstores> ls /создание блочного хранилища, с размером диска промахнулся и создал на 5гигабайт.

```
o- backstores ..... [Targets: 0]
  o- block ..... [Storage Objects: 1]
    | o- TASK12-DISK-5G ..... [/dev/sdb (5.0GiB) 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]
```

Далее необходимо создать iscsi target

> cd /iscsi

/iscsi > create wwn=iqn.2022-08.ru.test.gb:TASK12-DISK-5G

Created target iqn.2022-08.ru.test.gb:task12-disk-5g.

Created TPG 1.

Global pref auto\_add\_default\_portal=true

Created default portal listening on all IPs (0.0.0.0), port 3260.

```
o- / ..... [Targets: 0]
  o- backstores ..... [Targets: 0]
    | o- block ..... [Storage Objects: 1]
    | | o- TASK12-DISK-5G ..... [/dev/sdb (5.0GiB) 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: 1]
      o- iqn.2022-08.ru.test.gb:task12-disk-5g ..... [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]
    o- loopback ..... [Targets: 0]
```

Далее необходимо настроить iqn сервера-клиента



На сервере R1 устанавливается пакет iscsi-initiator-utils  
# cat /etc/iscsi/initiatorname.iscsi  
InitiatorName=iqn.1994-05.com.redhat:7b8630c1619

Далее на сервере R3

```
/> cd iscsi/iqn.2022-08.ru.test.gb:task12-disk-5g/tpg1/luns  
/iscsi/iqn.20...-5g/tpg1/luns> create /backstores/block/TASK12-DISK-5G  
Created LUN 0.
```

```
o- tpg1 ..... [no-gen-acls, no-auth]  
o- acls ..... [ACLs: 0]  
o- luns ..... [LUNs: 1]  
| o- lun0 ..... [block/TASK12-DISK-5G (/dev/sdb) (default_tg_pt_gp)]  
o- portals ..... [Portals: 1]  
o- 0.0.0.0:3260 ..... [OK]
```

```
/iscsi/iqn.20...-disk-5g/tpg1> cd acls  
/iscsi/iqn.20...-5g/tpg1/acls> create wwn=iqn.1994-05.com.redhat:7b8630c1619  
Created Node ACL for iqn.1994-05.com.redhat:7b8630c1619  
Created mapped LUN 0.  
iscsi target настроен.
```

```
o- / .....  
o- backstores .....  
| o- block ..... [Storage Objects: 1]  
| | o- TASK12-DISK-5G ..... [/dev/sdb (5.0GiB) 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.2022-08.ru.test.gb:task12-disk-5g ..... [TPGs: 1]  
| | o- tpg1 ..... [no-gen-acls, no-auth]  
| | o- acls ..... [ACLs: 1]  
| | | o- iqn.1994-05.com.redhat:7b8630c1619 ..... [Mapped LUNs: 1]  
| | | o- mapped_lun0 ..... [lun0 block/TASK12-DISK-5G (rw)]  
| | o- luns ..... [LUNs: 1]  
| | | o- lun0 ..... [block/TASK12-DISK-5G (/dev/sdb) (default_tg_pt_gp)]  
| | o- portals ..... [Portals: 1]  
| | o- 0.0.0.0:3260 ..... [OK]  
o- loopback ..... [Targets: 0]
```

Так же можно ограничить подсеть используемого портала, так как обычно подсеть для iscsi не маршрутизируемая.

Для этого нужно будет удалить созданный автоматически 0.0.0.0, а затем назначить на портал адрес, который находится на интерфейсе enp0s3 — 192.168.1.193.

Далее для подключения к iscsi-target из **man iscsiadm** вытащил команды-примеры:

```
iscsiadm --mode discoverydb --type sendtargets --portal 192.168.1.10 --discover  
iscsiadm --mode node --targetname iqn.2001-05.com.doe:test --portal 192.168.1.1:3260 --login  
и по фактическим данным меняю на необходимые данные для подключения:  
iscsiadm --mode discoverydb --type sendtargets --portal 192.168.1.193 --discover  
192.168.1.193:3260,1 iqn.2022-08.ru.test.gb:task12-disk-5g
```

```
iscsiadm --mode node --targetname iqn.2022-08.ru.test.gb:task12-disk-5g --portal  
192.168.1.193:3260 --login
```

```
Logging in to [iface: default, target: iqn.2022-08.ru.test.gb:task12-disk-5g, portal:  
192.168.1.193,3260] (multiple)
```

```
Login to [iface: default, target: iqn.2022-08.ru.test.gb:task12-disk-5g, portal: 192.168.1.193,3260]  
successful.
```

```
lsblk
```

```
[root@server1 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0   7G  0 part
│   └─centos-root 253:0    0  6.2G  0 lvm  /
│       └─centos-swap 253:1    0  820M  0 lvm  [SWAP]
sdb          8:16   0   5G  0 disk
sr0         11:0    1 1024M  0 rom
```

Диск виден и смонтирован.

```
[root@server1 ~]# dmesg | tail -n 15
[ 5159.203567] iscsi: registered transport (tcp)
[ 5317.923607] scsi host3: iSCSI Initiator over TCP/IP
[ 5317.932500] scsi 3:0:0:0: Direct-Access      LIO-ORG  TASK12-DISK-5G   4.0  PQ: 0 ANSI: 5
[ 5317.933684] scsi 3:0:0:0: alua: supports implicit and explicit TPGS
[ 5317.933687] scsi 3:0:0:0: alua: device naa.6001405a0334954fd7649f3b977d75b6 port group 0 rel port 1
[ 5317.933689] scsi 3:0:0:0: alua: Attached
[ 5317.935725] sd 3:0:0:0: Attached scsi generic sg2 type 0
[ 5317.939209] sd 3:0:0:0: alua: transition timeout set to 60 seconds
[ 5317.939212] sd 3:0:0:0: alua: port group 00 state A non-preferred supports TOLUSNA
[ 5317.939217] sd 3:0:0:0: [sdb] 10485760 512-byte logical blocks: (5.36 GB/5.00 GiB)
[ 5317.940706] sd 3:0:0:0: [sdb] Write Protect is off
[ 5317.940709] sd 3:0:0:0: [sdb] Mode Sense: 43 00 10 08
[ 5317.941125] sd 3:0:0:0: [sdb] Write cache: enabled, read cache: enabled, supports DPO and FUA
[ 5317.941720] sd 3:0:0:0: [sdb] Optimal transfer size 33550336 bytes
[ 5317.947128] sd 3:0:0:0: [sdb] Attached SCSI disk
```

Осталось смонтировать на диск файловую систему xfs. И после перезагрузить сервер, убедиться что диск будет доступен после ребута.

```
[root@server1 ~]# man mkfs
[root@server1 ~]# mkfs.xfs /dev/sdb
meta-data=/dev/sdb          isize=512    agcount=4, agsize=327680 blks
                =           sectsz=512    attr=2, projid32bit=1
                =           crc=1        finobt=0, sparse=0
data        =              bsize=4096    blocks=1310720, imaxpct=25
                =              sunit=0    swidth=0 blks
naming      =version 2          bsize=4096    ascii-ci=0 ftype=1
log         =internal log      bsize=4096    blocks=2560, version=2
                =              sectsz=512    sunit=0 blks, lazy-count=1
realtime    =none             extsz=4096    blocks=0, rtextents=0
[root@server1 ~]# mount /dev/sdb /tmp
[root@server1 ~]# mount tail -n 5
mount: mount point 5 does not exist
[root@server1 ~]# mount | tail -n 5
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
192.168.1.193:/nfs_1 on /mnt/export/nfs_1 type nfs (rw,noatime,vers=3,rsz=131072,wsz=131072,namlen=255,
=sys,mountaddr=192.168.1.193,mountvers=3,mountport=892,mountproto=tcp,local_lock=all,addr=192.168.1.193)
192.168.1.193:/nfs_2 on /mnt/export/nfs_2 type nfs (rw,noatime,vers=3,rsz=131072,wsz=131072,namlen=255,
=sys,mountaddr=192.168.1.193,mountvers=3,mountport=892,mountproto=tcp,local_lock=all,addr=192.168.1.193)
tmpfs on /run/user/0 type tmpfs (rw,nosuid,nodev,relatime,seclabel,size=101476k,mode=700)
/dev/sdb on /tmp type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Видно примонтированный диск sdb подключенный по iscsi, так же видны примонтированные папки с R3 уже после ребута сервера. Диск доступен после ребута сервера.

```
Using username "root".
Last login: Fri Aug 19 13:02:28 2022 from win-8rdq53jl7tf
[root@server1 ~]# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   8G  0 disk
├─sda1       8:1    0   1G  0 part /boot
├─sda2       8:2    0   7G  0 part
│   └─centos-root 253:0    0  6.2G  0 lvm  /
│       └─centos-swap 253:1    0  820M  0 lvm  [SWAP]
sdb          8:16   0   5G  0 disk
sr0         11:0    1 1024M  0 rom
[root@server1 ~]#
```