# Preparations:

# Prepare 3 machines (VM1 for server and VM2 and VM3 for clients)

# rewrite the vagrant file to name them homework-vmX, where x is number from 1 to 3.

# also change the network from 99 to 2, because of a problem with existing network adapters

A screenshot of a computer program

Description automatically generated

# So let’s begin:

# TASK1

(samba group share)

**# VM1**

sudo zypper install samba samba-client

**# after that (as in practice file) enable and start smb,nmb services,**

**# also add firewall exception and reload the configuration**

sudo systemctl enable smb nmb

sudo systemctl start smb nmb

sudo firewall-cmd --add-service samba --permanent

sudo firewall-cmd --reload

**# add group that will be used for samba shares**

sudo groupadd homework-task1

**# add folder to share**

sudo mkdir homework-task1

**# add the corresponding permissions**

sudo chgrp homework-task1 homework-task1

sudo chmod 770 homework-task1

**# add two users to the group**

sudo useradd vm2

sudo useradd vm3

sudo usermod -a -G homework-task1 vm2

sudo usermod -a -G homework-task1 vm3

**# create passwords and enable the users**

sudo smbpasswd -a vm2

sudo smbpasswd -a vm3

sudo smbpasswd -e vm2

sudo smbpasswd -e vm3

**# create backup of the original /etc/samba/smb.conf**

sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.bak

sudo vi /etc/samba/smb.conf

# restart and test config

sudo systemctl restart smb

testparm

A screenshot of a computer program

Description automatically generated

The force group option must have “k” in the group name to work, so I’ve edited it

## # test from users that are homework-task1 members:

**# VM2 and VM3**

sudo zypper install samba-client

A screenshot of a computer screen

Description automatically generated

## # test with user that is not part of homework-task1 group:

A computer screen shot of a computer program

Description automatically generated

# add the user to the group:

A screenshot of a computer

Description automatically generated

# TASK 2

(nfs share with ro and rw for two stations)

**# VM1 (again as in practice)**

sudo zypper install nfs-kernel-server

sudo systemctl enable --now nfsserver

**# check for exports**

A black screen with white text

Description automatically generated

**# create folder to share**

sudo mkdir homework\_task2

**# ip addresses of vm2 and vm3 must be 2.132 and 2.133**

**sudo vi /etc/exports**

A black background with white text

Description automatically generated

vagrant@homework-vm1:~/homework\_task2> sudo exportfs -rav

exporting 192.168.2.132:/home/vagrant/homework\_task2

exporting 192.168.2.133:/home/vagrant/homework\_task2**# allow service communication**

sudo firewall-cmd --add-service nfs --permanent

**# reload the firewall rules**

sudo firewall-cmd –reload

# VM2 and VM3

**# the nfs client is already installed, so we proceed with the next step -> mount the share**

sudo mkdir -p /mnt/homework\_task2

sudo mount -t nfs4 homework-vm1:/home/vagrant/homework\_task2 /mnt/homework\_task2

# TEST FROM READ ONLY IP (192.168.2.133) VM3:

A computer screen shot of a program

Description automatically generated

-----------------

# TEST FROM R/W IP (192.168.2.132) VM2:

A computer screen with white text

Description automatically generated

# The share can be mounted for autostart on both machines as in the practice file with adding this row in /etc/fstab:

homework-vm1:/home/vagrant/homework\_task2 /mnt/homework\_task2 nfs4 defaults 0 0

# TASK 3

(Create an iSCSI disk-based target)

(the machines are now under hyper-v so 2.131 is no longer the IP of homework-vm1)

# again as in practice

# VM1:

sudo zypper install targetcli-fb

# the fisk files will be stored here:

sudo mkdir homework-task3

# start the administration tool

sudo targetcli

# Switch to the fileio backend

cd backstores/fileio

# create 10GB fileio storage object

create homework\_disk /home/vagrant/homework-task3/homework\_disk.img 10G

# switch to the iscsi functions

cd /iscsi

# Define a new target with default name

A screen shot of a computer

AI-generated content may be incorrect.

# Enter the target

cd iqn.2003-01.org.linux-iscsi.homework-vm1.x8664:sn.5b76311274cb/tpg1/luns

# Create a new LUN

create /backstores/fileio/homework\_disk

# Created LUN 0.

# before registering the initator I am switching to vm2 to get its name

# VM2:

sudo zypper install open-iscsi

A computer screen with white text

AI-generated content may be incorrect.

# the name is iqn.1996-04.de.suse:01:43c4c54ab6a5 so we can use it in VM1 now

# but first lets reboot vm2

sudo reboot

# VM1:

# Adjust access to the resource

cd ../acls

# register the initiator

create iqn.1996-04.de.suse:01:43c4c54ab6a5

# Created Node ACL for iqn.1996-04.de.suse:01:43c4c54ab6a5

# Created mapped LUN 0.

# enter the record and set user and password

# (the defaults in vm2 /etc/iscsi/iscisd.conf are username/password)

cd iqn.1996-04.de.suse:01:43c4c54ab6a5/

set auth userid=username

set auth password=password

# now check the authentication flag

# as expected it is [no-gen-acls, no-auth]

# so we are going to /iscsi/iqn.2003-01.org.linux-iscsi.homework-vm1.x8664:sn.5b76311274cb/tpg1

# and execute:

set attribute authentication=1

exit

# adjust the firewall

sudo firewall-cmd --add-service iscsi-target --permanent

sudo firewall-cmd --reload

sudo systemctl enable --now targetcli.service

# aand back to VM2

# VM2:

# edit the /etc/iscsi/iscsi.conf

sudo vi /etc/iscsi/iscsi.conf

A screenshot of a computer program

AI-generated content may be incorrect.

# initiate target discovery

sudo iscsiadm -m discovery -t sendtargets -p homework-vm1

# iscsiadm: connect to 192.168.2.131 timed out

# iscsiadm: connect to 192.168.2.131 timed out

# iscsiadm: connect to 192.168.2.131 timed out

# so I am changing the hostname with the corresponding ip

sudo iscsiadm -m discovery -t sendtargets -p 192.168.99.160

# 192.168.99.160:3260,1 iqn.2003-01.org.linux-iscsi.homework-vm1.x8664:sn.5b76311274cb

# confirm:

sudo iscsiadm -m node -o show

# BEGIN RECORD 2.1.10

node.name = iqn.2003-01.org.linux-iscsi.homework-vm1.x8664:sn.5b76311274cb

node.tpgt = 1

node.startup = automatic

node.leading\_login = No

iface.iscsi\_ifacename = default

iface.net\_ifacename = <empty>

iface.ipaddress = <empty>

….

node.conn[0].address = 192.168.99.160

node.conn[0].port = 3260

….

node.conn[0].iscsi.MaxXmitDataSegmentLength = 0

node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144

node.conn[0].iscsi.HeaderDigest = None

node.conn[0].iscsi.DataDigest = None

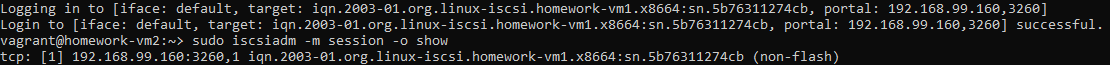
node.conn[0].iscsi.IFMarker = No

node.conn[0].iscsi.OFMarker = No

# END RECORD

# login to the target and confirm the established session

sudo iscsiadm -m node --login



# now we have sdb as new partion

sudo fdisk -l

A screenshot of a computer program

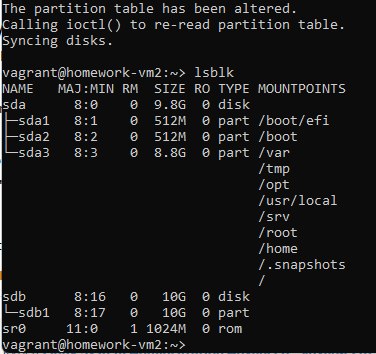
AI-generated content may be incorrect.

# and now (exactly as in the practice file) we can create partition, filesystem and mount it

# but instead of parted we can use fdisk, защото може

sudo fdisk /dev/sdb

n, p, 1, Enter, Enter, w



sudo mkfs.ext4 /dev/sdb1

sudo mkdir -p /mnt/homework\_task3

sudo mount /dev/sdb1 /mnt/homework\_task3

# It's there: /dev/sdb1      ext4      9.8G   24K  9.3G   1% /mnt/homework-task3  
# for “permanent” mounting we have to add in /etc/fstab with the following:

UUID="a359e31d-f6c9-43db-a3bc-8d9a612fe782" /mnt/homework\_task3 ext4 \_netdev 0 0

# the id was obtained from

sudo blkid /dev/sdb1

# /dev/sdb1: UUID="a359e31d-f6c9-43db-a3bc-8d9a612fe782" BLOCK\_SIZE="4096" TYPE="ext4" PARTUUID="592f466f-01"

# TASK 4

(GlusterFS dispersed volume)

# The machines are for now. homework-vm{1..3} will be the bricks and 4 will be the client.

# So VM1, VM2, VM3 as in practice file:

# install the missing repo, install glusterfs, enable the service (add the xml file in /etc/firewalld/services),

# add it to the firewall exceptions:

sudo zypper ar https://download.opensuse.org/repositories/home:/glusterfs:/SLES15SP5-10/15.5/home:glusterfs:SLES15SP5-10.repo

sudo zypper install glusterfs

sudo systemctl enable --now glusterd

sudo vi /etc/firewalld/services/glusterfs.xml

sudo systemctl reload firewalld

sudo firewall-cmd --add-service=glusterfs --permanent

sudo firewall-cmd --reload

mkdir homework\_task4

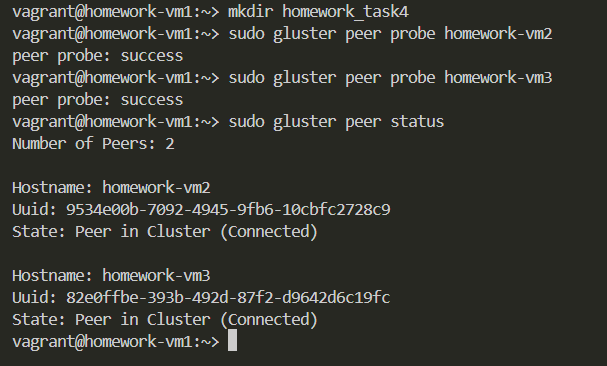
sudo gluster volume create vol01 disperse 3 redundancy 1 transport tcp \

homework-vm1:/home/vagrant/homework\_task4 \

homework-vm2:/home/vagrant/homework\_task4 \

homework-vm3:/home/vagrant/homework\_task4 force

# volume create: vol01: success: please start the volume to access data



sudo gluster volume info vol01

A computer screen shot of a black screen

AI-generated content may be incorrect.

# start the volume

sudo gluster volume start vol01

# verify status

sudo gluster volume status vol01

A screenshot of a computer

AI-generated content may be incorrect.

# VM4 also needs the repo and glusterfs installed

sudo zypper ar https://download.opensuse.org/repositories/home:/glusterfs:/SLES15SP5-10/15.5/home:glusterfs:SLES15SP5-10.repo

sudo zypper install glusterfs

# also need mounting point

sudo mkdir -p /mnt/homework\_task4

sudo mount -t glusterfs homework-vm1:/vol01 /mnt/homework\_task4

A screen shot of a computer

AI-generated content may be incorrect.

# test:



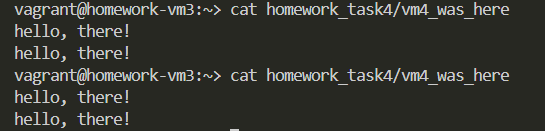




# the file exists on both 3 machines, but the behavior is odd, because of the calculated parity information.:





  
# Yep, I’ve checked twice   
# So I can simulate failure in order to see if the file is still accessible on the client.  
sudo systemctl stop glusterd  
# the file is still accessible on the client

sudo systemctl start glusterd

# The missing data should automatically recover from the parity.  
# And now if we want to make it permanent we have to add this to /etc/fstab

homework-vm1:/vol01 /mnt/homework\_task4 glusterfs defaults,\_netdev 0 0