Національний Технічний Університет України

«Київський Політехнічний Інститут Імені Ігоря Сікорського»

Факультет Інформатики Та Обчислювальної Техніки

Кафедра Обчислювальної Техніки

Лабораторна робота №4

Мережна файлова система NFS

Виконала

Шпак Діана

Група: ІО-з21мп

Київ 2023

2. Завдання на роботу.

2.1. Встановити та налаштувати файловий сервер, який реалізує мережну файлову систему NFS та відповідає наступним вимогам:

- створені каталоги: /nfs/public, /nfs/private, /nfs/incoming;

- каталог /nfs/public експортується в режимі тільки читання;

- каталог /nfs/private експортується в режимі читання та запису і тільки для вузла, на якому працює NFS-клієнт, запити від всіх користувачів на клієнті відображаються на заданого варіантом користувача на сервері, який має повні права в цьому каталозі;

- каталог /nfs/incoming експортується в режимі читання та запису для вузлів, які знаходяться в тій же мережі, що і NFS-клієнт, запити від користувача root на клієнті відображаються на заданого варіантом користувача на сервері, запити інших користувачів клієнта залишаються без змін на сервері, права на каталог встановлені відповідно варіанту завдання.

2.2. Виконати аналіз протокольного обміну між клієнтом та сервером під час монтування файлової системи та пересилки файлів.

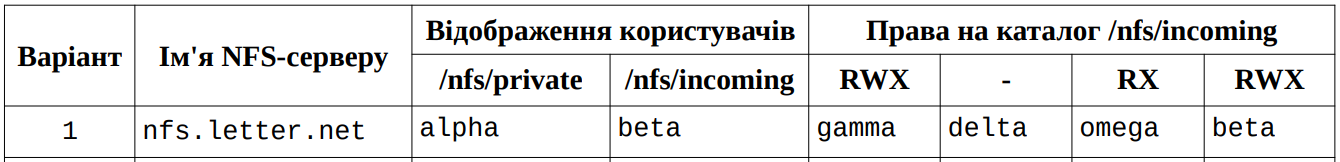
2.3. Рекомендується використовувати наступне програмне забезпечення:

- NFS-сервер: nfs-kernel-server;

- NFS-клієнт: nfs-common.

2.4. Для перевірки роботи файлового серверу та аналізу протокольного обміну рекомендується використовувати утиліти:

nfsstat, showmount, mount, tcpdump.



2.1. Встановити та налаштувати файловий сервер, який реалізує мережну файлову систему NFS та відповідає наступним вимогам:

- створені каталоги: /nfs/public, /nfs/private, /nfs/incoming;

- каталог /nfs/public експортується в режимі тільки читання;

- каталог /nfs/private експортується в режимі читання та запису і тільки для вузла, на якому працює NFS-клієнт, запити від всіх користувачів на клієнті відображаються на заданого варіантом користувача на сервері, який має повні права в цьому каталозі;

- каталог /nfs/incoming експортується в режимі читання та запису для вузлів, які знаходяться в тій же мережі, що і NFS-клієнт, запити від користувача root на клієнті відображаються на заданого варіантом користувача на сервері, запити інших користувачів клієнта залишаються без змін на сервері, права на каталог встановлені відповідно варіанту завдання.

Для виконання даної лабораторної роботи нам потрбно 4 віртуальних машини, 3 з яких лежать в одній мережі і 1 в інший, для перевірки доступу до експортованих каталогів з різних мереж, тому як завжди розпочинаємо з написання вагрант скриптів для запуску наших віртуальних машин:

Vagrant.configure("2") do |config|

config.vm.define "nfs" do |nfs|

nfs.vm.box\_download\_insecure = true

nfs.vm.box = "ubuntu/bionic64"

nfs.vm.network :private\_network, ip: "172.20.1.90"

nfs.vm.hostname = "nfs"

nfs.vm.post\_up\_message = "Run vm for NFS"

nfs.vm.provider :virtualbox do |vb|

vb.memory = "2048"

end

nfs.vm.provision "shell" do |s|

s.path = 'master.sh'

end

end

config.vm.define "clientnfs0" do |clientnfs0|

clientnfs0.vm.box\_download\_insecure = true

clientnfs0.vm.box = "ubuntu/bionic64"

clientnfs0.vm.network :private\_network, ip: "172.20.1.100"

clientnfs0.vm.hostname = "clientnfs0"

clientnfs0.vm.post\_up\_message = "Run vm for client nfs"

clientnfs0.vm.provider :virtualbox do |vb|

vb.memory = "2048"

end

clientnfs0.vm.provision "shell" do |s|

s.path = 'master.sh'

end

end

config.vm.define "clientnfs1" do |clientnfs1|

clientnfs1.vm.box\_download\_insecure = true

clientnfs1.vm.box = "ubuntu/bionic64"

clientnfs1.vm.network :private\_network, ip: "172.20.1.101"

clientnfs1.vm.hostname = "clientnfs1"

clientnfs1.vm.post\_up\_message = "Run vm for client nfs"

clientnfs1.vm.provider :virtualbox do |vb|

vb.memory = "2048"

end

clientnfs1.vm.provision "shell" do |s|

s.path = 'master.sh'

end

end

end

Після того як всі віртуальні машини запускаються з відповідними аймі адресами, можемо переходити до налаштування наших клієнтів і сервера:

На хост-сервері встановлюємо nfs-kernel-server пакет, який дозволить ділитися своїми каталогами. Оскільки це перша операція, яку ми виконуємо apt в цьому сеансі, оновлюємло локальний індекс пакета перед установкою:

sudo apt update

sudo apt install nfs-kernel-server

На клієнтськомусервері нам потрібно встановити пакет під назвою nfs-common, який надає функціональні можливості NFS без жодних серверних компонентів. Знову оновлюємо локальний індекс пакетів перед установкою, щоб переконатися, що у нас є актуальна інформація:

sudo apt update

sudo apt install nfs-common

Тепер, коли обидва сервери мають необхідні пакети, ми можемо почати їх налаштування.

Спочатку створюємо спільні каталоги:

vagrant@nfs:~$ sudo mkdir -p /nfs/private

vagrant@nfs:~$ ls -la /nfs/private

total 8

drwxr-xr-x 2 root root 4096 Jun 25 15:24 .

drwxr-xr-x 3 root root 4096 Jun 25 15:24 ..

vagrant@nfs:~$ sudo mkdir -p /nfs/incoming

vagrant@nfs:~$ ls -la /nfs/incoming

total 8

drwxr-xr-x 2 root root 4096 Jun 25 15:25 .

drwxr-xr-x 4 root root 4096 Jun 25 15:25 ..

vagrant@nfs:~$ sudo mkdir -p /nfs/public

vagrant@nfs:~$ ls -la /nfs/public

total 8

drwxr-xr-x 2 root root 4096 Jun 25 15:25 .

drwxr-xr-x 5 root root 4096 Jun 25 15:25 ..

Оскільки ми створюємо його за допомогою sudo, каталог належить користувачеві root хоста, далі нам потрібно змінити власника каталогу, права до каталогів і групу, щоб він відповідав обліковим даним.

sudo adduser alpha

sudo adduser beta

sudo adduser gamma

sudo adduser delta

sudo adduser omega

sudo chown alpha:alpha /nfs/private

sudo chown nobody:nogroup /nfs/incoming

sudo chown nobody:nogroup /nfs/public

sudo sudo groupadd mygroup1

sudo usermod -a -G mygroup1 beta

sudo usermod -a -G mygroup1 gamma

sudo chmod 775 /nfs/private

Для виконання завдання нам будуть потрібні айді юзерів і айді груп, яуі можна отримати наступним чином

sudo cat /etc/passwd

root:x:0:0:root:/root:/bin/bash

daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin

bin:x:2:2:bin:/bin:/usr/sbin/nologin

sys:x:3:3:sys:/dev:/usr/sbin/nologin

sync:x:4:65534:sync:/bin:/bin/sync

games:x:5:60:games:/usr/games:/usr/sbin/nologin

man:x:6:12:man:/var/cache/man:/usr/sbin/nologin

lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin

mail:x:8:8:mail:/var/mail:/usr/sbin/nologin

news:x:9:9:news:/var/spool/news:/usr/sbin/nologin

uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin

proxy:x:13:13:proxy:/bin:/usr/sbin/nologin

www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin

backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin

irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin

gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin

nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin

systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin

systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin

syslog:x:102:106::/home/syslog:/usr/sbin/nologin

messagebus:x:103:107::/nonexistent:/usr/sbin/nologin

\_apt:x:104:65534::/nonexistent:/usr/sbin/nologin

lxd:x:105:65534::/var/lib/lxd/:/bin/false

uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin

dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin

landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin

sshd:x:109:65534::/run/sshd:/usr/sbin/nologin

pollinate:x:110:1::/var/cache/pollinate:/bin/false

vagrant:x:1000:1000:,,,:/home/vagrant:/bin/bash

ubuntu:x:1001:1001:Ubuntu:/home/ubuntu:/bin/bash

statd:x:111:65534::/var/lib/nfs:/usr/sbin/nologin

alpha:x:1002:1003:,,,:/home/alpha:/bin/bash

beta:x:1003:1004:,,,:/home/beta:/bin/bash

gamma:x:1004:1005:,,,:/home/gamma:/bin/bash

delta:x:1005:1006:,,,:/home/delta:/bin/bash

omega:x:1006:1007:,,,:/home/omega:/bin/bash

Далі ми заглибимося в конфігураційний файл NFS, щоб налаштувати спільне використання цих ресурсів.

На головній машині відкриваємо /etc/exports файл у текстовому редакторі з правами root :

sudo nano /etc/exports

Початковий вигляд файлу буде ось такий, з прикладами для експорту каталогів:

# /etc/exports: the access control list for filesystems which may be exported

# to NFS clients. See exports(5).

#

# Example for NFSv2 and NFSv3:

# /srv/homes hostname1(rw,sync,no\_subtree\_check) hostname2(ro,sync,no\_subtree\_check)

#

# Example for NFSv4:

# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no\_subtree\_check)

# /srv/nfs4/homes gss/krb5i(rw,sync,no\_subtree\_check)

#

Нам потрібно буде створити рядки для кожного з каталогів, якими ми плануємо поділитися. Обов’язково вказуємо мережу для якої будемо експортувати потрібний нам каталог:

/nfs/public \*(ro,sync,no\_subtree\_check)

/nfs/private 172.20.1.100(rw,sync,root\_squash, anonuid=1002,anongid=1002)

/nfs/incoming 172.20.1.0/24 (rw,sync, no\_root\_squash, no\_subtree\_check, anonuid=1003, anongid=1003)

Після збереження змін робимо рестарт сервера і переходимо до налаштування клієнтів:

vagrant@nfs:~$ sudo systemctl restart nfs-kernel-server

vagrant@nfs:~$ sudo systemctl status nfs-kernel-server

● nfs-server.service - NFS server and services

Loaded: loaded (/lib/systemd/system/nfs-server.service; enabled; vendor preset: enabled)

Active: active (exited) since Sun 2023-06-25 16:08:57 UTC; 14s ago

Process: 16103 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCESS)

Process: 16097 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCESS)

Process: 16092 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)

Process: 16114 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)

Process: 16113 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)

Main PID: 16114 (code=exited, status=0/SUCCESS)

Jun 25 16:08:57 nfs systemd[1]: Starting NFS server and services...

Jun 25 16:08:57 nfs exportfs[16113]: exportfs: /etc/exports [2]: Neither 'subtree\_check' or 'no\_subtree\_check'

Jun 25 16:08:57 nfs exportfs[16113]: Assuming default behaviour ('no\_subtree\_check').

Jun 25 16:08:57 nfs exportfs[16113]: NOTE: this default has changed since nfs-utils version 1.0.x

Jun 25 16:08:57 nfs systemd[1]: Started NFS server and services.

Тепер, коли хост- сервер налаштовано та обслуговує свої спільні ресурси, ми підготуємо наш клієнт.

Щоб зробити віддалені спільні ресурси доступними на клієнті, нам потрібно змонтувати каталоги на хості, яким ми хочемо надати спільний доступ, до порожніх каталогів на клієнті.

Створимо користувачів на клієнті:

починаємо з 1 клієнта, даному клієнті мають бути доступні всі каталоги для монтування

sudo adduser alpha

sudo adduser beta

sudo adduser gamma

sudo adduser delta

sudo adduser omega

Ми створимо два каталоги для наших монтувань:

vagrant@clientnfs0:~$ sudo mkdir -p /nfs/private

vagrant@clientnfs0:~$ ls -la /nfs/private

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:09 .

drwxr-xr-x 3 root root 4096 Jun 25 17:09 ..

vagrant@clientnfs0:~$ sudo mkdir -p /nfs/incoming

vagrant@clientnfs0:~$ ls -la /nfs/incoming

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:10 .

drwxr-xr-x 4 root root 4096 Jun 25 17:10 ..

vagrant@clientnfs0:~$ sudo mkdir -p /nfs/public

vagrant@clientnfs0:~$ ls -la /nfs/public

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:10 .

drwxr-xr-x 5 root root 4096 Jun 25 17:10 ..

Тепер, коли у нас є місце для розміщення віддалених спільних ресурсів і ми відкрили брандмауер, ми можемо підключити спільні ресурси за допомогою IP-адреси нашого хост- сервера:

/nfs/public \*(ro,sync,no\_subtree\_check)

/nfs/private 172.20.1.100(rw,sync,root\_squash, anonuid=1002,anongid=1002)

/nfs/incoming 172.20.1.0/24 (rw,sync, no\_root\_squash, no\_subtree\_check, anonuid=1003, anongid=1003)

sudo mount 172.20.1.90:/nfs/private /nfs/private

sudo mount 172.20.1.90:/nfs/incoming /nfs/incoming

sudo mount 172.20.1.90:/nfs/public /nfs/public

Ці команди монтуватимуть спільні ресурси з головного комп’ютера на клієнтську машину. Перевірити, чи успішно вони встановлені, можна кількома способами. Ви можете перевірити це за допомогою команди mount або findmnt, але df -h надає більш зрозумілий результат:

vagrant@clientnfs0:~$ df -h

Filesystem Size Used Avail Use% Mounted on

udev 984M 0 984M 0% /dev

tmpfs 200M 632K 199M 1% /run

/dev/sda1 39G 1.4G 38G 4% /

tmpfs 997M 0 997M 0% /dev/shm

tmpfs 5.0M 0 5.0M 0% /run/lock

tmpfs 997M 0 997M 0% /sys/fs/cgroup

vagrant 457G 173G 285G 38% /vagrant

tmpfs 200M 0 200M 0% /run/user/1000

172.20.1.90:/nfs/private 39G 1.4G 38G 4% /nfs/private

172.20.1.90:/nfs/incoming 39G 1.4G 38G 4% /nfs/incoming

172.20.1.90:/nfs/public 39G 1.4G 38G 4% /nfs/public

Всі спільні ресурси, які ми змонтували, відображаються внизу. Оскільки їх було змонтовано з однієї файлової системи, вони показують однакове використання диска. Щоб побачити, скільки місця фактично використовується під кожною точкою монтування, скористайтеся командою використання диска duта шляхом монтування. Прапор -s надає підсумок використання, а не відображає використання для кожного файлу. Друкує -h зрозумілий для людини вихід.

vagrant@clientnfs0:~$ du -sh /nfs/private

4.0K /nfs/private

vagrant@clientnfs0:~$ du -sh /nfs/incoming

4.0K /nfs/incoming

vagrant@clientnfs0:~$ du -sh /nfs/public

4.0K /nfs/public

Переходимо до налаштування 2 клієнта, виконуємо всі попередні операції але нам тут буде недоступна /nfs/private бо ми неї експортували для конкретної айпі адреси

Створюємо каталоги для монтування:

vagrant@clientnfs1:~$ sudo mkdir -p /nfs/private

vagrant@clientnfs1:~$ ls -la /nfs/private

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:25 .

drwxr-xr-x 3 root root 4096 Jun 25 17:25 ..

vagrant@clientnfs1:~$ sudo mkdir -p /nfs/incoming

vagrant@clientnfs1:~$ ls -la /nfs/incoming

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:25 .

drwxr-xr-x 4 root root 4096 Jun 25 17:25 ..

vagrant@clientnfs1:~$ sudo mkdir -p /nfs/public

vagrant@clientnfs1:~$ ls -la /nfs/public

total 8

drwxr-xr-x 2 root root 4096 Jun 25 17:26 .

drwxr-xr-x 5 root root 4096 Jun 25 17:26 ..

Монтуємо і бачимо що каталог /nfs/private не монтується все працює вірно

/nfs/public \*(ro,sync,no\_subtree\_check)

/nfs/private 172.20.1.100(rw,sync,root\_squash, anonuid=1002,anongid=1002)

/nfs/incoming 172.20.1.0/24 (rw,sync, no\_root\_squash, no\_subtree\_check, anonuid=1003, anongid=1003)

vagrant@clientnfs1:~$ sudo mount 172.20.1.90:/nfs/private /nfs/private

mount.nfs: access denied by server while mounting 172.20.1.90:/nfs/private

vagrant@clientnfs1:~$ sudo mount 172.20.1.90:/nfs/incoming /nfs/incoming

vagrant@clientnfs1:~$ sudo mount 172.20.1.90:/nfs/public /nfs/public

Перевіряємо що /nfs/incoming і /nfs/public примонтувалися до нашого клієнта

vagrant@clientnfs1:~$ df -h

Filesystem Size Used Avail Use% Mounted on

udev 984M 0 984M 0% /dev

tmpfs 200M 644K 199M 1% /run

/dev/sda1 39G 1.4G 38G 4% /

tmpfs 997M 0 997M 0% /dev/shm

tmpfs 5.0M 0 5.0M 0% /run/lock

tmpfs 997M 0 997M 0% /sys/fs/cgroup

vagrant 457G 173G 285G 38% /vagrant

tmpfs 200M 0 200M 0% /run/user/1000

172.20.1.90:/nfs/incoming 39G 1.4G 38G 4% /nfs/incoming

172.20.1.90:/nfs/public 39G 1.4G 38G 4% /nfs/public

2.2. Виконати аналіз протокольного обміну між клієнтом та сервером під час монтування файлової системи та пересилки файлів.

vagrant@clientnfs0:~$ df -h

Filesystem Size Used Avail Use% Mounted on

udev 984M 0 984M 0% /dev

tmpfs 200M 632K 199M 1% /run

/dev/sda1 39G 1.4G 38G 4% /

tmpfs 997M 0 997M 0% /dev/shm

tmpfs 5.0M 0 5.0M 0% /run/lock

tmpfs 997M 0 997M 0% /sys/fs/cgroup

vagrant 457G 171G 287G 38% /vagrant

tmpfs 200M 0 200M 0% /run/user/1000

172.20.1.90:/nfs/private 39G 1.4G 38G 4% /nfs/private

172.20.1.90:/nfs/incoming 39G 1.4G 38G 4% /nfs/incoming

172.20.1.90:/nfs/public 39G 1.4G 38G 4% /nfs/public

vagrant@clientnfs1:~$ df -h

Filesystem Size Used Avail Use% Mounted on

udev 984M 0 984M 0% /dev

tmpfs 200M 644K 199M 1% /run

/dev/sda1 39G 1.4G 38G 4% /

tmpfs 997M 0 997M 0% /dev/shm

tmpfs 5.0M 0 5.0M 0% /run/lock

tmpfs 997M 0 997M 0% /sys/fs/cgroup

vagrant 457G 172G 285G 38% /vagrant

172.20.1.90:/nfs/incoming 39G 1.4G 38G 4% /nfs/incoming

172.20.1.90:/nfs/public 39G 1.4G 38G 4% /nfs/public

tmpfs 200M 0 200M 0% /run/user/1000

як бачимо пермішини вірні

vagrant@clientnfs0:/nfs/public$ touch file\_client1

touch: cannot touch 'file\_client1': Read-only file system

public тільки для читання, в каталозі incoming є можливість створення файлів під рутом

vagrant@clientnfs1:/nfs/incoming$ touch test\_client2

touch: cannot touch 'test\_client2': Permission denied

vagrant@clientnfs1:/nfs/incoming$ sudo touch test\_client2

vagrant@clientnfs1:/nfs/incoming$ ls

file\_incoming.txt test test2 test\_client2

nfsstat -c

Client rpc stats:

calls retrans authrefrsh

198 0 198

Client nfs v4:

null read write commit open

2 1% 0 0% 0 0% 0 0% 4 2%

open\_conf open\_noat open\_dgrd close setattr

0 0% 0 0% 0 0% 1 0% 1 0%

fsinfo renew setclntid confirm lock

9 4% 0 0% 0 0% 0 0% 0 0%

lockt locku access getattr lookup

0 0% 0 0% 9 4% 12 6% 12 6%

lookup\_root remove rename link symlink

3 1% 0 0% 0 0% 0 0% 0 0%

create pathconf statfs readlink readdir

0 0% 6 3% 4 2% 0 0% 1 0%

server\_caps delegreturn getacl setacl fs\_locations

15 7% 0 0% 0 0% 0 0% 0 0%

rel\_lkowner secinfo fsid\_present exchange\_id create\_session

0 0% 0 0% 0 0% 4 2% 2 1%

destroy\_session sequence get\_lease\_time reclaim\_comp layoutget

1 0% 106 53% 0 0% 2 1% 0 0%

getdevinfo layoutcommit layoutreturn secinfo\_no test\_stateid

0 0% 0 0% 0 0% 3 1% 0 0%

free\_stateid getdevicelist bind\_conn\_to\_ses destroy\_clientid seek

0 0% 0 0% 0 0% 1 0% 0 0%

allocate deallocate layoutstats clone

0 0% 0 0% 0 0% 0 0%