

목차

환경	3
NFS-Server 기본 구성	3
NFS-Client 기본 구성	3
NFS-Server && Client (rw 옵션) 마운트 테스트	4
NFS 구성시 주의점	5

환경

- Red Hat Enterprise Linux 7.3
- NFS-Server
 - Hostname : system1.example.com
- NFS-Client
 - Hostname : system2.example.com

NFS-Server 기본 구성

1. 패키지 설치 및 구동

```
[root@system1 ~]# yum install nfs-server nfs-secure
[root@system1 ~]# systemctl status nfs-secure nfs-server
[root@system1 ~]# systemctl enable nfs-server nfs-secure
[root@system1 ~]# systemctl start nfs-server nfs-secure
```

1. NFS 마운트 폴더 생성

```
// NFS 에서 사용할 mount 폴더 생성
[root@system1 ~]# mkdir /public
[root@system1 ~]# mkdir /private
```

1. NFS Export 설정

```
[root@system1 ~]# vi /etc/exports
/public      *.example.com(ro)      // ro 옵션은 Read only
/private     *.example.com(rw,no_root_squash,sec=krb5p) // rw 옵션은
ReadWrite,

[root@system1 ~]# exportfs -avr      // exports 파일에서 설정한 내용 시스템에 적용
exporting *.example.com:/private
exporting *.example.com:/public
```

1. 방화벽 설정

```
[root@system1 ~]# firewall-cmd --permanent --add-service=nfs
success
[root@system1 ~]# firewall-cmd --reload
success
```

NFS-Client 기본 구성

1. NFS-Utills 패키지 설치

```
[root@system2 ~]# yum install nfs-utils
```

1. NFS Client 에서 마운트할 디렉토리 생성

```
[root@system2 ~]# mkdir /mnt/nfsmount  
[root@system2 ~]# mkdir /mnt/nfssecure
```

1. NFS 마운트 테스트

```
[root@system2 ~]# mount -t nfs 192.168.70.46:/public /mnt/nfsmount/  
[root@system2 ~]# cd /mnt/nfsmount/  
[root@system2 ~]# ls
```

1. /etc/fstab 설정

```
[root@system2 ~]# vim /etc/fstab  
system1.example.com:/public /mnt/nfsmount nfs defaults,_netdev 0 0  
system1.example.com:/private /mnt/nfssecure nfs defaults,_netdev,sec=krb5p 0  
0
```

NFS-Server & Client (rw 옵션) 마운트 테스트

- rw 옵션은 Read & Write 가 가능하며, 허용된 User 에 대해서만 접근 권한이 있음
- 해당 테스트는 User “ Koov ” 유저만 rw 만 가능하게 설정하여 테스트 진행 함

System1.example.com

```
[root@system1 ~]# useradd koov  
[root@system1 ~]# id koov  
uid=1001(koov) gid=1001(koov) groups=1001(koov)  
[root@system1 ~]# chown koov -R /private/  
[root@system1 ~]# ls -ld /private/  
drwxr-xr-x. 3 koov root 30 Jan 24 16:56 /private/
```

system2.example.com

```
[root@system2 secret]# id koov  
uid=1001(koov) gid=1001(koov) groups=1001(koov)  
[root@system2 ~]# mount -t nfs system1.example.com:/private /mnt/nfssecure/  
[root@system2 ~]# cd /mnt/nfssecure  
[root@system2 nfssecure]# whoami  
root  
[root@system2 nfssecure]# touch TEST.txt  
touch: cannot touch 'TEST.txt': Permission denied
```

```
[root@system2 ~]# su - koov
Last login: Tue Jan 24 17:09:03 KST 2017 on pts/0
[koov@system2 ~]$ whoami
koov
[koov@system2 ~]$ cd /mnt/nfssecure/
[koov@system2 nfssecure]$ touch TEST.txt
[koov@system2 nfssecure]$ ls
TEST.txt
```

NFS 구성시 주의점

동일 NFS서버에 마운트가 필요한 경우 마운트 포인트중 몇개는 마운트가 안되는 경우가 생긴다. 예를 들어

```
192.168.0.5:/volume2/PUBLIC /pub nfs _netdev,defaults 0 0
192.168.0.5:/volume2/ISO /iso nfs _netdev,defaults 0 0
```

위와같이 192.168.0.5 서버에 2개의 exports가 존재하고 둘다 마운트해야할때 클라이언트에 위와같이 fstab에 선언하여도 둘다 마운트 되는것이 아니라 하나만 마운트 되는 경우가 있다.

이것은 NFS서버쪽에서 짧은시간에 여러개의 마운트 요청이 들어오면 하나만 처리되고 나머지는 무시되는 현상이 발생해서 그렇다.

이럴때는 마운트 하는 클라이언트 쪽에서 해당 경로 접근시 자동 마운트 하도록 automount기능을 이용하는것이 좋다.

```
192.168.0.5:/volume2/PUBLIC /pub nfs _netdev,defaults,x-
systemd.automount,x-systemd.idle-timeout=600 0 0
192.168.0.5:/volume2/ISO /iso nfs _netdev,defaults,x-
systemd.automount,x-systemd.idle-timeout=600 0 0
```

위와같이 하면 해당경로를 시스템이 접근하는 순간 자동으로 마운트 해주기 때문에 부팅시 딜레이도 줄일수 있고 안정적이므로 권장한다.

From:

<https://atl.kr/dokuwiki/> - AllThatLinux!

Permanent link:

https://atl.kr/dokuwiki/doku.php/nfs_%EC%84%A4%EC%A0%95

Last update: 2026/01/20 01:04

