

<LVM 설정 매뉴얼>

이 매뉴얼은 시스템에 연결된 하드디스크 중 sdb와 sdc를 VG(Volume Group)으로 설정하고 LV(Logical Volume)를 생성하는 과정을 기준으로 LVM(Logical Volume Manager)에 대해 설명한다. LVM의 가장 큰 매력은 기존의 데이터를 보존하면서 가용 용량을 확장할 수 있는 것이라고 할 수 있다. 예를 들면, 200GB 용량의 LV를 사용하다가 필요에 의해 50GB를 기존의 볼륨에 더해 250GB 만큼의 용량을 사용할 수 있다는 것이다.

사용된 하드디스크는 180GB이며, IEEE 1394 인터페이스를 통해 연결하였다.

저널링 파일시스템이면서, 향후 기존 데이터를 유지하면서 볼륨 사이즈를 확장할 수 있는 reiserfs 파일시스템을 사용하였는데, ext2 파일시스템에 비해 속도가 빠르다는 장점이 있다.

Kernel 2.4.27, LVM ver. 1.0.8

<필수설정>

1. fdisk를 통해 /dev/sdb1 와 /dev/sdc1 파티션 타입을 '0x8e' 변경.

먼저 기존의 파티션을 모두 삭제한 후 새로운 파티션을 생성해야 오류가 적다.

2. PV(Physical Volume) 생성.

```
# pvcreate /dev/sd[bc]1"
```

```
pvcreate -- physical volume "/dev/sdb1" successfully created
```

```
pvcreate -- physical volume "/dev/sdc1" successfully created
```

* 한 디스크 상에 여러개의 파티션을 나눠 테스트를 해 볼 수는 있지만, 실제 사용할 때는 이렇게 하면 안 된다. 성능이 떨어지기 때문이다.

* LVM을 처음 설정할 경우 /etc/lvmtab 이 없으므로 'vgscan'을 먼저 실행한다.

```
# vgscan
```

```
vgscan -- reading all physical volumes (this may take a while...)
```

```
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
```

```
vgscan -- WARNING: This program does not do a VGDA backup of your volume group
```

3. VG 생성.

'vgcreate'는 새로운 VG을 활성화시켜 그 위에 LV를 생성할 수 있도록 한다.

앞에서 PV로 설정한 두 파티션의 전체 용량을 갖고, 이름은 'test_vg'.

```
# vgcreate test_vg /dev/sd[bc]1
```

```
vgcreate -- INFO: using default physical extent size 4 MB
```

```
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
```

```
vgcreate -- doing automatic backup of volume group "test_vg"
```

```
vgcreate -- volume group "test_vg" successfully created and activated
```

* 위와 같이 pv를 생성하면 최대 LV 크기가 256GB로 제한된다. (default extents value = 4MB) 이보다 큰 용량의 LV를 생성하고자 하면 extents 값을 늘려주어야 한다. extents 값을 32로 하면 최대 LV 크기가 2TB가 된다. (16으로 하면 1TB)

```
# vgcreate -s 32 test_vg /dev/sd[bc]1
vgcreate -- INFO: maximum logical volume size is 2 Terabyte
vgcreate -- doing automatic backup of volume group "test_vg"
vgcreate -- volume group "test_vg" successfully created and activated
```

4. LV 생성

(1) 200GB 용량의 linear 볼륨, 이름은 'test_lv'

이 볼륨의 block device 명은 '/dev/test_vg/test_lv'

```
# lvcreate -L 200G -n test_lv test_vg
lvcreate -- doing automatic backup of "test_vg"
lvcreate -- logical volume "/dev/test_vg/test_lv" successfully created
```

(2) 200GB 용량의 stripe 볼륨, 이름은 'test_lv'

2 stripes and stripesize 32KB

이 때 stripes의 수는 VG를 생성할 때 사용한 PV의 갯수와 같아야 한다.

```
# lvcreate -i 2 -I 32 -L 200G -n test_lv test_vg
lvcreate -- doing automatic backup of "test_vg"
lvcreate -- logical volume "/dev/test_vg/test_lv" successfully created
```

5. 파일시스템 생성 & 마운트. reiserfs 사용, /lv1 에 마운트.

```
# mkreiserfs /dev/test_vg/test_lv
# mount /dev/test_vg/test_lv /lv1
```

<볼륨 확장 및 축소>

기존 볼륨의 크기를 변경하기 전에 반드시 볼륨을 unmount 해야 한다.

6. 상대적인 크기 표현법으로 /dev/test_vg/test_lv 크기를 250GB로 확장

```
# lvextend -L +50G /dev/test_vg/test_lv
lvextend -- extending logical volume "/dev/test_vg/test_lv" to 250 GB
lvextend -- doing automatic backup of volume group "test_vg"
lvextend -- logical volume "/dev/test_vg/test_lv" successfully extended
```

혹은 절대적인 크기로도 변경 가능

```
# lvextend -L 250G /dev/test_vg/test_lv
```

7. 상대적인 크기 표현법으로 /dev/test_vg/test_lv 크기를 150GB로 축소

```
# lvreduce -L -100G /dev/test_vg/test_lv
```

혹은 절대적인 크기로도 변경 가능

```
# lvreduce -L 150G /dev/test_vg/test_lv
```

8. 파일시스템 크기 변경. 'resize_reiserfs' 사용

```
# resize_reiserfs /dev/test_vg/test_lv
```

<볼륨 이름 변경>

9. VG 이름 변경

```
# vgrename test_vg whatever
```

10. LV 이름 변경

```
# lvrename /dev/whatever/test_lv /dev/whatever/whatvolume
```

혹은

```
# lvrename whatever test_lv whatvolume
```

<특이사항>

- IEEE 1394로 연결한 하드디스크는 부팅시 LVM 보다 늦게 설정되므로 VG이 자동으로 활성화되지 않는다. 따라서 /etc/rc.local 에 다음의 내용을 삽입하여 VG을 활성화시키고 원하는 위치에 마운트 하도록 한다.

```
/sbin/vgscan
```

```
/sbin/vgchange -a y
```

```
mount /dev/test_vg/test_lv /lv1
```

- VG을 삭제하려면 먼저 해당 VG을 비활성화(deactivation)시켜야 한다.

```
# vgchange -a n VG_name
```

```
# vgremove VG_name
```

- ext3 파일시스템에서도 LV 사이즈를 확장 또는 축소 할 수 있다. 우선 6, 7과 같이 LV 사이즈를 확장 또는 축소한 후, 'resize2fs' 명령어를 사용한다.

```
# resize2fs /dev/test_vg/test_lv2
```