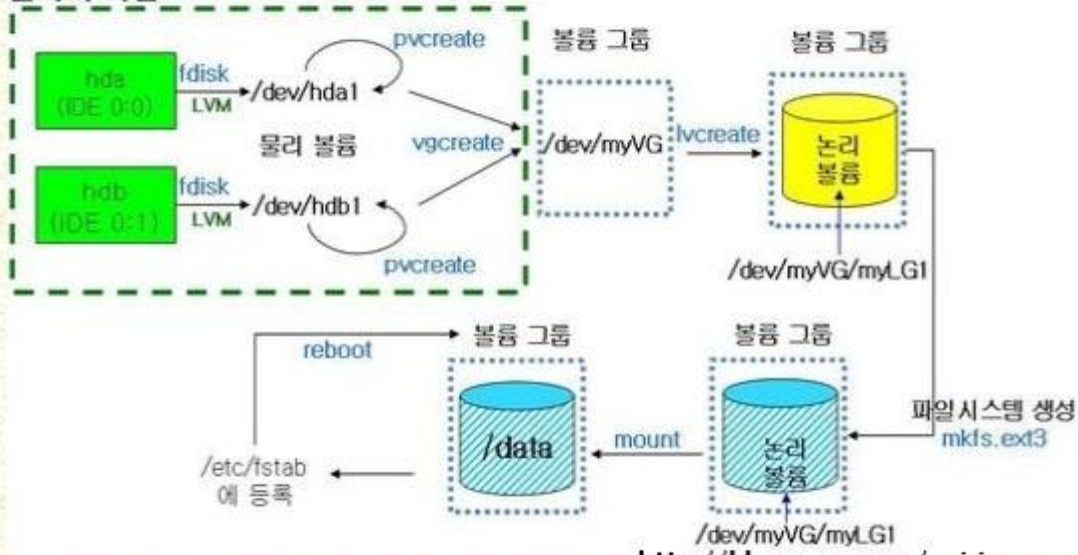


Logical Volume Manager - LVM

- o PCreate (Physical Volume)
- o VGCreate (Volume Group)
- o LVCreate (Logical Volume)

• LVM 흐름도

선처리 작업



o HDD 장착

1. fdisk 작업 => type | 8e 로 설정
2. reboot
3. pvcreate [device] //물리적인 하드 디스크 선언

```
#pvcreate /dev/hdb1
#pvcreate /dev/hdc1
```

4. vgcreate [그룹명][device 1][device 2]....
5. lvcreate -i [실제값] -n [파티션명][그룹명]

실제값 5555 입력하면 에러메시지출력됨.

본인의 프리사이즈를 에러메시지에서 찾아서 재입력함.

(에러 메시지가 불필요하면, HDD 의 용량을 실제 계산해서 사용해도 무방함).

6. reboot
7. mkfs.ext3 /dev/그룹명/파티션명
8. 마운트포인트생성

```
#mkdir /tmp/lvmtest
```

9. 마운트

```
#mount -t ext3 /dev/짱구/짱구1 /tmp/lvmtest
```

10. /etc/fstab 에 등록

□ Exercise

- HDD 를 두개 장착한 후, 이 두개의 HDD 를 LVM 으로 만들기 위해 fdisk 한다.

```
[root@localhost ~]# fdisk -l

Disk /dev/hda: 1073 MB, 1073741824 bytes
16 heads, 63 sectors/track, 2080 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1            1         2080     1048208+   8e  Linux LVM

Disk /dev/hdb: 1073 MB, 1073741824 bytes
16 heads, 63 sectors/track, 2080 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1         2080     1048208+   8e  Linux LVM

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *           1           13       104391    83  Linux
/dev/sda2             14          1958     15623212+   8e  Linux LVM
[root@localhost ~]# _
```

<http://blog.naver.com/goisimasyou>

```
#init 6
```

```
#pvcreate /dev/hda1
```

Physical volume "/dev/hda1" successfully created

```
#pvcreate /dev/hdb1
```

Physical volume "/dev/hdb1" successfully created

```
#vgcreate bbq /dev/hda1 /dev/hdb1
```

/dev/hdb: open failed : No medium found

Volume group "bbq" successfully created

```
#vgdisplay
```

```
#lvcreate -l 1024 -n bbq 2mb
```

```
Logical volume "bbq"create

#init 6
#mkfs.ext3 /dev/2mb/bbq
#mkdir /tmp/lvmtest
#mount -t ext3 /dev/2mb/bbq /tmp/lvmtest
#cd /tmp/lvmtest
#touch lvmt.txt

□ LVM 삭제방법

#vgremove bbq
#lvremove /dev/2mb/bbq

□ LVM 변경방법

#vgrename bbq bbq
#lvrename /dev/2mb/bbq /dev/2mb/bbq
```

LVM(Logical Volume Manager)

저장장치들을 좀더 효율적이고 유연하게 관리할 수 있는 커널의 부분과 프로그램을 말하며, 처음에는 IBM 에서 개발되었는데, 그후에 OSF(현재는 OpenGroup <http://www.opengroup.org>)에서 차용을 하여 OSF/1 operating system 에서 사용

■ LVM 와 RAID (Redundant Array of Inexpensive Drives)

● 비슷한 점

- 여러 물리적인 디스크들을 하나의 논리적인 디스크처럼 다룸
- 조합방법에 따라 고용량, 고속, 데이터의 무결성을 실현

● 차이점

- LVM 은 RAID 보다 관리 및 확장이 비교적 쉬움
- RAID 는 LVM 에는 없는 disk mirroring(RAID level 1), Parity Stripe (RAID level 4,5) 등의 방식이 있어 속도 또는 데이터의 무결성을 보장

■ LVM 의 특징

- 쉬운 관리.
- 서로 다른 많은 디바이스 조합 지원.
- 직관적인 저장 장치로의 접근.
- 뛰어난 확장성.
- 믿음만한 안전성과 효율.
- 스냅샷 기능 지원.

■ LVM 의 이점

- 간단하게 저장공간을 확장, 축소
- 중소형 서버에서는 비교적 적은 비용으로 대용량 저장 장치를 구현
- 백업 없이 기존의 환경을 유지한 채 확장이 가능(물론 백업 중요)

→ 유지보수면에서 상당한 이득

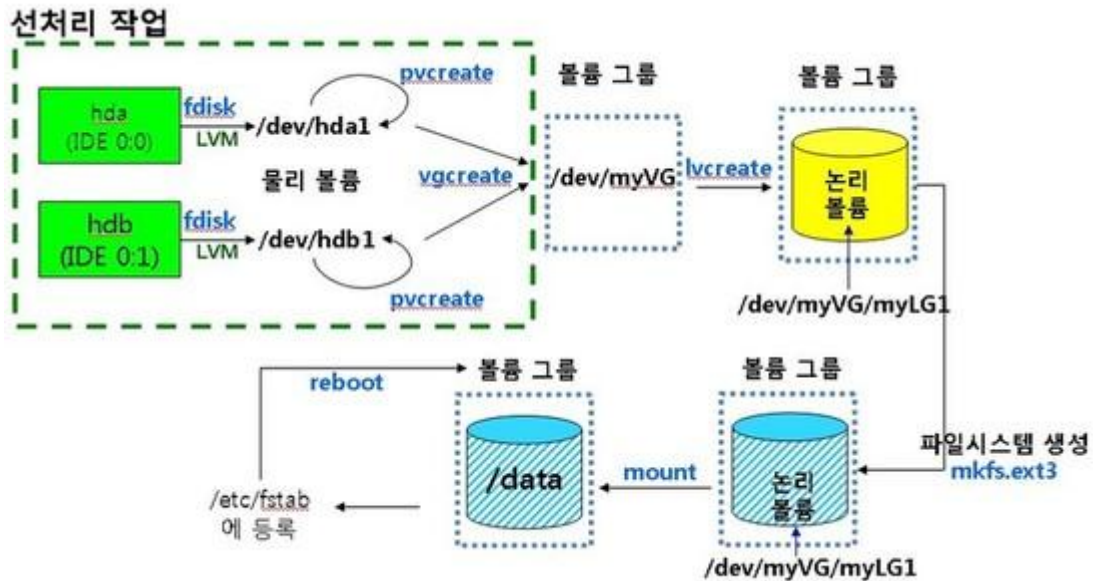
■ LVM 구성

● 개념

두 개(또는 그 이상) 하드 드라이브를 두 개의 포맷한 후, 두 드라이브를 하나의 논리 드라이브 그룹으로

만들어 원하는 단위로 구분하여 개별적인 논리 볼륨을 만듦

● LVM 흐름도



● 작업순서

○ 각각의 하드 드라이브를 fdisk를 통해 **lv 시스템으로 파티션 생성**

```
]#fdisk /dev/hda
:n(새로운 파티션생성)
=>primary partition으로 지정
:t(타입지정)
=>8e(lvm)
:w(저장)
```

※같은 방법으로 다른 하드 드라이브도 lvm 시스템으로 파티션 생성

```
]#reboot(파티션 작업후 재부팅)
```

```
]#pvcreate /dev/hda
]#pvcreate /dev/hdb
=> 물리적으로 하드디스크 볼륨 생성
※ pv:physical volume
```

```
]#vgcreate [그룹명] /dev/hda /dev/hdb
=> 물리디스크를 [그룹명]으로 하나로 묶어줌
※vg:volume group
```

```
]#lvcreate -l 실제값 -n [볼륨명] [그룹명]
=> 하나로 묶어진 그룹에 논리적 볼륨 생성
```

=> 실제값은 알기 어려므로 임의의 값(ex:5555)을 넣어주면 실제값을 보여주며 에러메세지가 뜨는데 이 실제값을 넣어 명령 실행
※lv:logical volume

]#reboot(논리 그룹 작업후 재부팅)

]#mkfs.ext3 /dev/[그룹명]/[볼륨명]

=>파일 시스템을 만들어줌

※mkfs:make file system

]#마운트전에 대상 디렉토리 생성

]#mount -t ext3 /dev/[그룹명]/[볼륨명] /마운트할 디렉토리

]vi /etc/fstab 를 통해 등록하면 마무리

*볼륨 확장시

Vgextend “vgroup name” “drive name”

]#vgextend VolGroup00 /dev/sdc1

[출처] [LVM\(Logical Volume Manager\)](#) |작성자 [jh0110love](#)

<http://www.turnkeylinux.org/blog/extending-lvm>

Extending a Logical Volume

Bottom line: if the underlying Volume Group doesn't have enough free space, to extend the Logical Volume you'll first have to extend the underlying Volume Group by adding another Physical Volume to it.

In VMWare you could either create a new virtual hard disk device to add to the volume group, or extend an existing virtual hard disk device, create a new partition with cfdisk, and add the new partition to the Volume Group:

example #1: you've added to VMWare a new virtual hard disk called /dev/sdb

```
pvccreate /dev/sdb
```

```
vgextend turnkey /dev/sdb
```

example #2: you've expanded the existing sda hard disk

```
cfdisk /dev/sda # creating /dev/sda3 (you may need to reboot before you can see this)
```

```
pvccreate /dev/sda3
```

```
vgextend turnkey /dev/sda3
```

After you've extended the Volume Group, you are free to extend the underlying Logical Volume:

```
# lvextend -L+10G /dev/turnkey/root
```

Extending logical volume root to 27.0 GiB

Logical volume root successfully resized

Finally, you'll have to resize the filesystem within /dev/turnkey/root so it can see that the underlying block device just got 10G bigger:

```
# resize2fs /dev/turnkey/root
```

```
resize2fs 1.41.11 (14-Mar-2010)
```

```
Filesystem at /dev/turnkey/root is mounted on /; on-line resizing required
```

```
old desc_blocks = 2, new_desc_blocks = 2
```

```
Performing an on-line resize of /dev/turnkey/root to 7077888 (4k) blocks.
```

```
The filesystem on /dev/turnkey/root is now 7077888 blocks long.
```