

LVM(Logical Volume Manager)

1. LVM 이란?

논리 볼륨 관리자 라는 뜻으로, 여러 개의 물리적인 하드디스크 파티션을 Volume Group 으
로 묶은 다음, 논리적인 디스크로 할당하여 유연하게 관리 할 수 있게 해 주는 기술

2. LVM 특징

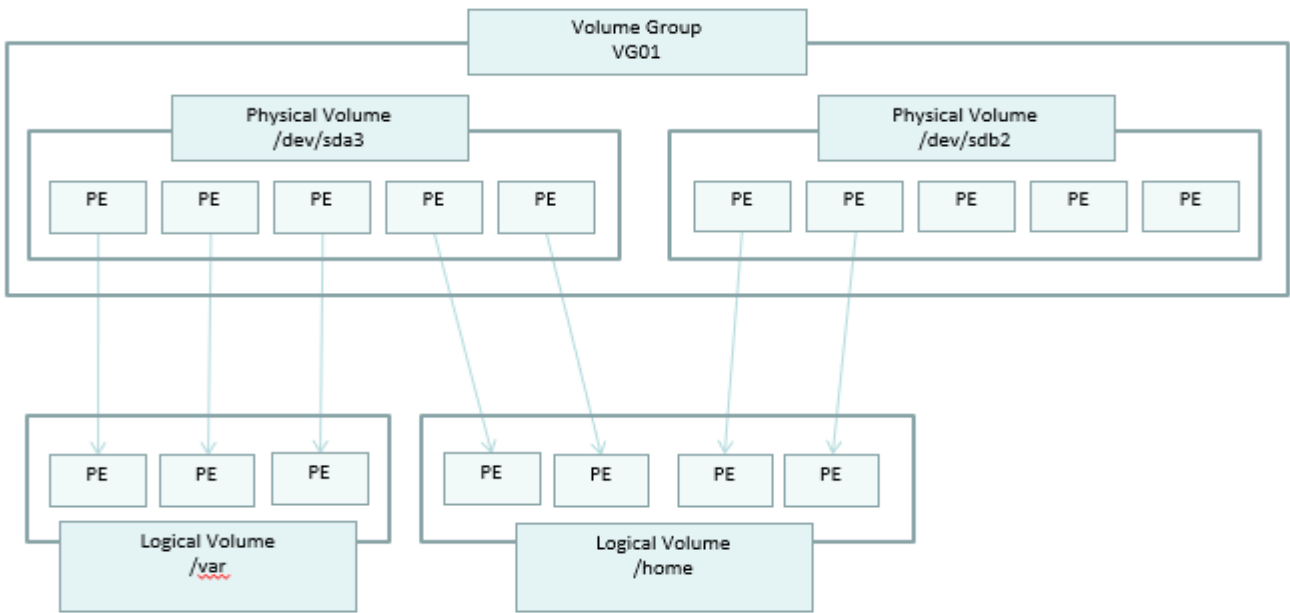
LVM은, 물리적으로 구분되어 있는 하드디스크 장치를 연결해서 하나로 사용이 가능하며, 확
장성, 안정성, 효율성 등이 뛰어납니다. 그리고, 스냅샷 기능을 이용해서 백업 및 복원도 편리
하게 할 수 있습니다

- 쉬운 관리
- 서로 다른 많은 디바이스 조합 지원
- 직관적인 저장 장치로의 접근
- 뛰어난 확장성
- 믿을 만한 안전성과 효율성
- 스냅샷 기능 지원

3. LVM 구조

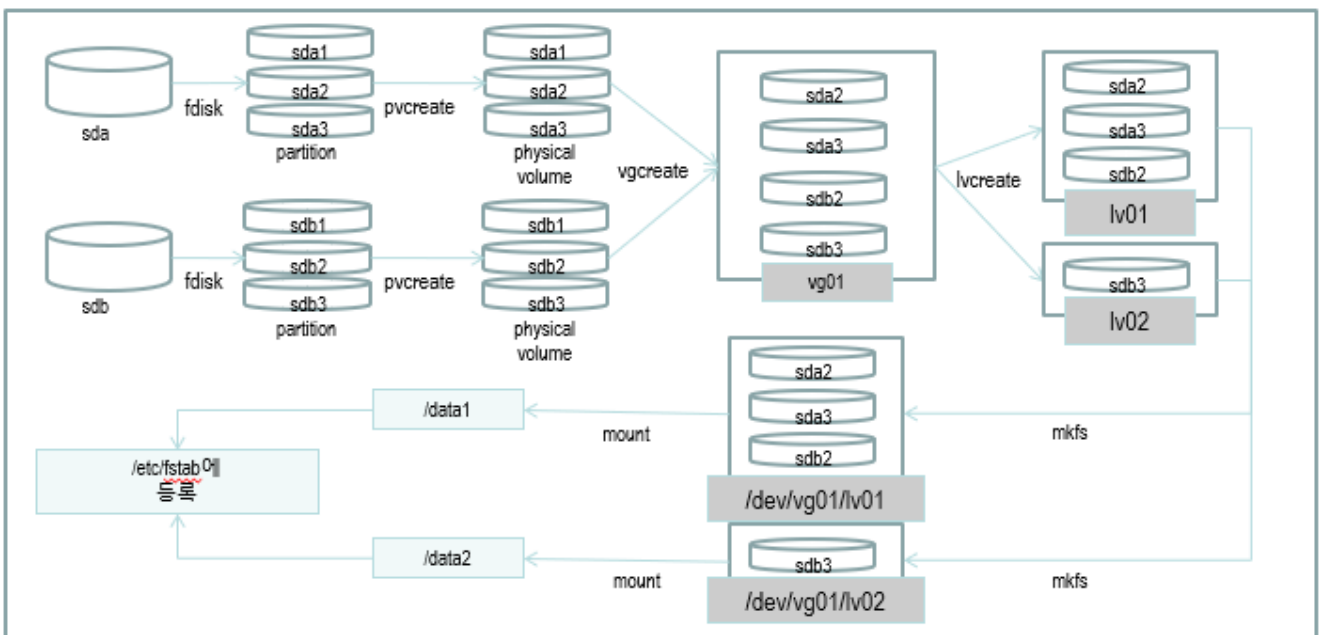
Volume Group은 여러 개의 Physical Volume으로 구성되어 있으며, 각각의 Physical Volume은
여러 개의 PE로 구성 됩니다. Logical Volume은 기존의 파티션과 같이 디렉토리에 mount해서
사용할 수 있는 Volume 입니다. Logical Volume은 각각의 Physical Volume에 포함돼 있는 여
러 PE들을 합쳐서, 원하는 Size로 구성합니다.

LVM 구조



4. LVM 구현 방법

- Fdisk를 통해서 생성한 파티션을 나누고, 파티션 타입을 Linux LVM(8e)로 선택합니다.
- 생성된 파티션을 pvcreate 명령으로 physical volume으로 생성합니다.
- vgcreate명령으로 각각의 PV(physical volume)들을 묶어 VG(volume group)를 만듭니다.
- lvcreate 명령으로 VG에서 원하는 size 만큼의 LV(logical volume)을 만듭니다.
- 생성된 LV에 filesystem을 생성하고, /etc/fstab에 등록하여 마운트 합니다.



5. LVM 구성 실습

- 20G 디스크를 5G 용량의 파티션 4개로 나눠서 파티션 타입을 8e(Linux LVM)으로 설정

```
[root@localhost ~]# fdisk -l /dev/xvdb
```

```
Disk /dev/xvdb: 21.5 GB, 21474836480 bytes, 41943040 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
[root@localhost ~]# fdisk /dev/xvdb
```

```
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
```

```
Be careful before using the write command.
```

```
Device does not contain a recognized partition table
```

```
Building a new DOS disklabel with disk identifier 0x4ecf7646.
```

```
Command (m for help): n
```

```
Partition type:
```

```
  p   primary (0 primary, 0 extended, 4 free)
```

```
  e   extended
```

```
Select (default p): p
```

```
Partition number (1-4, default 1):
```

```
First sector (2048-41943039, default 2048):
```

```
Using default value 2048
```

```
Last sector, +sectors or +size{K,M,G} (2048-41943039, default 41943039): +5G
```

```
Partition 1 of type Linux and of size 5 GiB is set
```

Command (m for help): **n**

Partition type:

p primary (1 primary, 0 extended, 3 free)

e extended

Select (default p): **p**

Partition number (2-4, default 2):

First sector (10487808-41943039, default 10487808):

Using default value 10487808

Last sector, +sectors or +size{K,M,G} (10487808-41943039, default 41943039): **+5G**

Partition 2 of type Linux and of size 5 GiB is set

Command (m for help): **n**

Partition type:

p primary (2 primary, 0 extended, 2 free)

e extended

Select (default p): **p**

Partition number (3,4, default 3):

First sector (20973568-41943039, default 20973568):

Using default value 20973568

Last sector, +sectors or +size{K,M,G} (20973568-41943039, default 41943039): **+5G**

Partition 3 of type Linux and of size 5 GiB is set

Command (m for help): **n**

Partition type:

p primary (3 primary, 0 extended, 1 free)

e extended

Select (default e): **p**

Selected partition 4

First sector (31459328-41943039, default 31459328):

Using default value 31459328

Last sector, +sectors or +size{K,M,G} (31459328-41943039, default 41943039):

Using default value 41943039

Partition 3 of type Linux and of size 5 GiB is set

Command (m for help): **t**

Partition number (1-4, default 4): **1**

Hex code (type L to list all codes): **8e**

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): **t**

Partition number (1-4, default 4): **2**

Hex code (type L to list all codes): **8e**

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): **t**

Partition number (1-4, default 4): **3**

Hex code (type L to list all codes): **8e**

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): **t**

Partition number (1-4, default 4): **4**

Hex code (type L to list all codes): **8e**

Changed type of partition 'Linux' to 'Linux LVM'

Command (m for help): **p**

Disk /dev/xvdb: 21.5 GB, 21474836480 bytes, 41943040 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x4ecf7646

Device	Boot	Start	End	Blocks	Id	System
/dev/xvdb1		2048	10487807	5242880	8e	Linux LVM
/dev/xvdb2		10487808	20973567	5242880	8e	Linux LVM
/dev/xvdb3		20973568	31459327	5242880	8e	Linux LVM
/dev/xvdb4		31459328	41943039	5241856	8e	Linux LVM

Partition table entries are not in disk order

Command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

[root@localhost ~]# **fdisk -l /dev/xvdb**

Disk /dev/xvdb: 21.5 GB, 21474836480 bytes, 41943040 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk label type: dos

Disk identifier: 0x4ecf7646

Device	Boot	Start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

/dev/xvdb1	2048	10487807	5242880	8e	Linux LVM
/dev/xvdb2	10487808	20973567	5242880	8e	Linux LVM
/dev/xvdb3	20973568	31459327	5242880	8e	Linux LVM
/dev/xvdb4	31459328	41943039	5241856	8e	Linux LVM

- PV(physical volume) 생성 / 확인

```
[root@localhost ~]# pvcreate /dev/xvdb1 /dev/xvdb2 /dev/xvdb3 /dev/xvdb4
```

Physical volume "/dev/xvdb1" successfully created.

Physical volume "/dev/xvdb2" successfully created.

Physical volume "/dev/xvdb3" successfully created.

Physical volume "/dev/xvdb4" successfully created.

```
[root@localhost ~]# pvs
```

```
PV          VG Fmt Attr PSize  PFree
/dev/xvdb1  lvm2 ---  5.00g  5.00g
/dev/xvdb2  lvm2 ---  5.00g  5.00g
/dev/xvdb3  lvm2 ---  5.00g  5.00g
  /dev/xvdb4  lvm2 ---  <5.00g <5.00g
```

- VG01 이라는 이름으로 VG(volume group) 생성 / 확인

```
[root@localhost ~]# vgcreate VG01 /dev/xvdb1 /dev/xvdb2 /dev/xvdb3 /dev/xvdb4
```

Volume group "VG01" successfully created

```
[root@localhost ~]# vgs
```

```
VG   #PV #LV #SN Attr   VSize  VFree
VG01  4   0   0 wz--n- 19.98g 19.98g
```

- data라는 이름의 8G size LV(logical volume) 생성 / 확인

```
[root@localhost ~]# lvcreate -n data -L 8G VG01
```

Logical volume "data" created.

```
[root@localhost ~]# lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
data VG01 -wi-a----- 8.00g
```

➤ 파일 시스템 생성 및 mount

```
[root@localhost ~]# mkfs.xfs /dev/VG01/data
```

```
meta-data=/dev/VG01/data      isize=512    agcount=4, agsize=524288 blks
      =                       sectsz=512   attr=2, projid32bit=1
      =                       crc=1        finobt=0, sparse=0
data      =                       bsize=4096  blocks=2097152, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096  ascii-ci=0 ftype=1
log       =internal log      bsize=4096  blocks=2560, version=2
      =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096  blocks=0, rtextents=0
```

```
[root@localhost ~]# mkdir /data
```

```
[root@localhost ~]# mount /dev/VG01/data /data/
```

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda3	5.8G	1.9G	4.0G	32%	/
devtmpfs	2.0G	0	2.0G	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	2.0G	8.4M	2.0G	1%	/run
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/xvda1	253M	105M	148M	42%	/boot
tmpfs	395M	0	395M	0%	/run/user/0
/dev/mapper/VG01-data	8.0G	33M	8.0G	1%	/data

6. LVM 운영

➤ Logical Volume 확장

lvextend 명령으로 기존의 LV 사이즈를 확장하고, xfs_growfs 명령으로 확장한 size를 적용해 준다.
ext2 나 ext3 파일시스템일 경우는 resize2fs -p /dev/VG01/data 명령을 사용

```
[root@localhost ~]# lvextend -L+2G /dev/VG01/data
```

Size of logical volume VG01/data changed from 8.00 GiB (2048 extents) to 10.00 GiB (2560 extents).

Logical volume VG01/data successfully resized.

```
[root@localhost ~]# xfs_growfs /data/
```

```
meta-data=/dev/mapper/VG01-data  isize=512    agcount=4, agsize=524288 blks
          =                       sectsz=512   attr=2, projid32bit=1
          =                       crc=1       finobt=0 spinodes=0
data      =                       bsize=4096 blocks=2097152, imaxpct=25
          =                       sunit=0     swidth=0 blks
naming    =version 2               bsize=4096  ascii-ci=0 ftype=1
log       =internal               bsize=4096  blocks=2560, version=2
          =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                   extsz=4096  blocks=0, rtextents=0
```

data blocks changed from 2097152 to 2621440

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda3	5.8G	1.9G	4.0G	32%	/
devtmpfs	2.0G	0	2.0G	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	2.0G	8.4M	2.0G	1%	/run
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/xvda1	253M	105M	148M	42%	/boot

```
tmpfs                395M    0 395M   0% /run/user/0
/dev/mapper/VG01-data 10G    33M  10G   1% /data
```

➤ Logical Volume Snapshot

lvcreate 명령에 `-s` 옵션을 추가하여 스냅샷 볼륨을 생성 한다.

생성된 snapshot volume을 다른 디렉토리에 mount하여 사용하면 된다.

```
[root@localhost /]# lvcreate -L8G -s -n snap_data /dev/VG01/data
```

```
Using default stripesize 64.00 KiB.
```

```
Logical volume "snap_data" created.
```

```
[root@localhost /]# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
data	VG01	owi-a-s---	8.00g									
snap_data	VG01	swi-a-s---	8.00g		data	0.00						

xfs 파일시스템일 경우 snapshot volume의 uuid까지 동일하기 때문에 mount가 불가능.

mount 시에 `-o nouuid` 옵션을 주고 mount 한다.

```
[root@localhost /]# blkid /dev/VG01/data
```

```
/dev/VG01/data: UUID="ee03c164-6061-47f8-b769-bd0030fe6113" TYPE="xfs"
```

```
[root@localhost /]# blkid /dev/VG01/snap_data
```

```
/dev/VG01/snap_data: UUID="ee03c164-6061-47f8-b769-bd0030fe6113" TYPE="xfs"
```

```
[root@localhost /]# mount /dev/VG01/snap_data /snap_data
```

```
mount: wrong fs type, bad option, bad superblock on /dev/mapper/VG01-snap_data,
missing codepage or helper program, or other error
```

In some cases useful info is found in syslog - try

`dmesg | tail` or so.

```
[root@localhost ~]# mount -o nouuid /dev/VG01/snap_data /snap_data
```

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda3	5.8G	1.9G	4.0G	32%	/
devtmpfs	2.0G	0	2.0G	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	2.0G	8.5M	2.0G	1%	/run
tmpfs	2.0G	0	2.0G	0%	/sys/fs/cgroup
/dev/xvda1	253M	105M	148M	42%	/boot
tmpfs	395M	0	395M	0%	/run/user/0
/dev/mapper/VG01-data	8.0G	33M	8.0G	1%	/data
/dev/mapper/VG01-snap_data	8.0G	33M	8.0G	1%	/snap_data