

Linux 4.1

[Linux 3.2](#)



LVM 확장 및 RAID 구성 완벽 가이드



1. 디스크 볼륨 확장

기존 볼륨 그룹(VG), 논리볼륨(LV)을 확장하는 과정입니다.



확장 과정 개요

1. ♦ PV 생성
2. ♦ VG에 PV 추가
3. ♦ LV 확장
4. 🔗 마운트

💡 사전 조건:

- 기존: /dev/nvme0n3p1 , /dev/nvme0n4p1 사용
- 추가: 60GB 1개 추가하여 /dev/nvme0n5 작업
- ID를 8e로 변경 완료
- VG명: myvg , LV명: mylv

1-1. ♦ PV 생성

기본 구문:

```
pvcreate /dev/nvme0n5p1
```

```
[root@localhost ~]# pvcreate /dev/nvme0n7p1
Physical volume "/dev/nvme0n7p1" successfully created.
[root@localhost ~]#
```

1-2. ♦ VG에 PV 추가

기본 구문:

```
vgextend myvg /dev/nvme0n5p1
```

```
[root@localhost 바 탕 화 면 ]# vgextend myvg /dev/nvme0n5p1
Volume group "myvg" successfully extended
[root@localhost 바 탕 화 면 ]#
```

💡 설명:

- 기존 볼륨 그룹 myvg 에 새로 생성한 물리볼륨 /dev/nvme0n5p1 을 추가
- 볼륨 그룹을 확장하면 더 많은 논리 볼륨을 만들거나 기존 논리 볼륨을 확장할 수 있음
- vgcreate myvg /dev/nvme0n3p1 /dev/nvme0n4p1 명령어에서 사용한 그룹명 myvg 를 입력

1-3. ♦ LV 확장

기본 구문:

```
lvextend -l +5119 /dev/myvg/mylv
```

```
[root@localhost ~]# lvextend -l +5119 /dev/myvg/mylv
Size of logical volume myvg/mylv changed from 119.99 GiB (30718 extents) to <139.99 GiB (35837 extents).
Logical volume myvg/mylv successfully resized.
```

💡 설명:

- mylv 논리볼륨을 5119개의 익스텐트만큼 확장
- -l 옵션은 익스텐트 단위로 크기를 지정하는 옵션
- 볼륨 그룹을 확장한 후, 논리 볼륨을 확장해야 해당 공간을 활용할 수 있음

1-4. 🔗 마운트

기본 구문:

```
mount /dev/myvg/mylv /root/mnt3
```

```
root@localhost:~  
[root@localhost ~]# mount /dev/myvg/mylv /root/mnt3  
mount: /root/mnt3: /dev/mapper/myvg-mylv already mounted on /root/mnt2.  
[root@localhost ~]# mount /dev/myvg/mylv /root/mnt3  
mount: /root/mnt3: /dev/mapper/myvg-mylv already mounted on /root/mnt3.  
[root@localhost ~]# df  
Filesystem            1K-blocks    Used Available Use% Mounted on  
devtmpfs                4096         0      4096    0% /dev  
tmpfs                  1856532      0    1856532    0% /dev/shm  
tmpfs                   742616     9924    732692    2% /run  
/dev/mapper/rl-root    38744064 5553140 33190924  15% /  
/dev/nvme0n1p1         983040    355452   627588   37% /boot  
/dev/mapper/rl-home    18878464 164776  18713688    1% /home  
/dev/nvme0n2p1         62848000 471224  62376776    1% /mnt  
tmpfs                   371304         96    371208    1% /run/user/0  
/dev/mapper/myvg-mylv 125755392 909816 124845576    1% /root/mnt3  
[root@localhost ~]#
```

1-5. XFS 파일 시스템 확장

기본 구문:

```
xfs_growfs /dev/myvg/mylv
```

```
[root@localhost ~]# xfs_growfs /dev/myvg/mylv  
meta-data=/dev/mapper/myvg-mylv isize=512    agcount=4, agsize=7863808 blks  
        =                               sectsz=512   attr=2, projid32bit=1  
        =                               crc=1        finobt=1, sparse=1, rmapbt=0  
        =                               reflink=1     bigtime=1 inobtcount=1 nnext64=0  
/data    =                               bsize=4096   blocks=31455232, imaxpct=25  
        =                               sunit=0      swidth=0 blks  
naming   =version 2                       bsize=4096   ascii-ci=0, ftype=1  
log      =internal log                    bsize=4096   blocks=16384, version=2  
        =                               sectsz=512   sunit=0 blks, lazy-count=1  
realtime =none                            extsz=4096    blocks=0, rtextents=0  
data blocks changed from 31455232 to 36697088
```

 설명:

- XFS 파일 시스템에서 확장된 논리 볼륨의 크기를 반영하는 명령어
- 논리 볼륨을 확장하더라도 파일 시스템 크기는 자동으로 증가하지 않으므로, 명시적으로 확장해야 함

1-6. LVM 상태 확인 명령어

pvdisplay - Physical Volume 정보 확인

```
[root@localhost ~]# pvdisk
--- Physical volume ---
PV Name           /dev/nvme0n3p1
VG Name           myvg
PV Size           <60.00 GiB / not usable 3.00 MiB
Allocatable       yes (but full)
PE Size           4.00 MiB
Total PE          15359
Free PE           0
Allocated PE      15359
PV UUID           zo8xnD-JWZD-D4Bg-IpAD-ZxFp-kNWP-N032vb
```

항목	설명
PV Name	물리적 볼륨명
VG Name	PV가 속한 볼륨 그룹명
PV Size	PV 크기
Allocatable	PV 할당 가능 여부
PE Size	물리적인 PE(Physical Extent) 크기
Total PE	전체 PE 개수
Free PE	사용 가능한 PE 개수
Allocated PE	할당된 PE 개수
PV UUID	PV 식별자

vgdisplay - Volume Group 정보 확인

--- Volume group ---

VG Name	rl
System ID	
Format	lvm2
Metadata Areas	1
Metadata Sequence No	4
VG Access	read/write
VG Status	resizable
MAX LV	0
Cur LV	3
Open LV	3
Max PV	0
Cur PV	1
Act PV	1
VG Size	<59.00 GiB
PE Size	4.00 MiB
Total PE	15103
Alloc PE / Size	15103 / <59.00 GiB
Free PE / Size	0 / 0
VG UUID	Y4Z8Bi-HzIS-LQca-TY5S-nIxK-SXKc-v06onT

항목	설명
VG Name	VG명
System ID	시스템 ID
Format	LVM 포맷 (보통 <code>lvm2</code>)
Metadata Areas	메타 데이터 영역 개수
Metadata Sequence No	메타 데이터 순서
VG Access	읽기/쓰기 가능 여부
VG Status	크기 재설정 상태
MAX LV	생성 가능한 최대 LV 개수
Cur LV	현재 LV 개수
Open LV	현재 열린 LV 개수
Max PV	포함 가능한 최대 PV 개수
Cur PV	현재 포함된 PV 개수
Act PV	활성화된 PV 개수
VG Size	VG 전체 크기
PE Size	PE 전체 크기
Total PE	전체 PE 수
Alloc PE / Size	할당된 PE 개수
Free PE / Size	할당 가능한 PE 개수
VG UUID	VG 식별자

lvdisplay - Logical Volume 정보 확인

```

--- Logical volume ---
LV Path                /dev/rl/root
LV Name                 root
VG Name                 rl
LV UUID                 VFHdQK-jGmX-NQKJ-cZr2-vGBt-G5lJ-Glm1wS
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2025-02-05 23:51:08 +0900
LV Status               available
# open                  1
LV Size                 37.01 GiB
Current LE              9475
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     256
Block device            253:0

```

항목	설명
LV Path	LV 경로
LV Name	LV명
VG Name	LV가 속한 VG명
LV UUID	LV 식별자
LV Write Access	읽기/쓰기 가능 여부
LV Creation host, time	생성된 호스트명 및 생성 시간
LV Status	LV 상태
# open	열린 LV 개수
LV Size	LV 크기
Current LE	LV Extent 개수
Segments	세그먼트 개수
Allocation	할당 여부
Read ahead sectors	자동 설정 (auto)
- currently set to	현재 설정된 값 (기본: 256)
Block device	블록 디바이스 번호











2. RAID (Redundant Array of Independent Disks)

RAID 종류

- **Linear RAID:** 순차적 저장
- **RAID 0:** Striping (성능 향상)
- **RAID 1:** Mirroring (안정성 향상)
- **RAID 5:** 패리티를 이용한 분산 저장

사전 준비

▼ Devices

 Memory	4 GB
 Processors	2
 Hard Disk (NVMe)	60 GB
 Hard Disk 2 (NVMe)	60 GB
 Hard Disk 3 (NVMe)	60 GB
 CD/DVD (IDE)	Auto detect
 Network Adapter	NAT
 USB Controller	Present
 Sound Card	Auto detect
 Display	Auto detect

💡 이 상태에서 스냅샷을 찍어 단계별로 이 상태로 시작합니다.

파티션 ID 변경

```
fdisk /dev/nvme0n2
fdisk /dev/nvme0n3
```

Enter를 계속 누르고, t → fd 입력 후 p로 확인

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): fd
Changed type of partition 'Linux' to 'Linux raid autodetect'.





Command (m for help): p
Disk /dev/nvme0n2: 60 GiB, 64424509440 bytes, 125829120 sectors
Disk model: VMware Virtual NVMe Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x62d4c216

Device           Boot Start      End  Sectors  Size Id Type
/dev/nvme0n2p1    2048 125829119 125827072   60G fd Linux raid autodetect

Command (m for help):
```

2-1. 📄 Linear RAID

🎯 특징

-  여러개의 디스크를 하나의 큰 논리적 볼륨으로 결합
-  데이터를 연속적으로 저장 → 한 디스크가 다 차면 다음 디스크에 저장
-  스토리지 용량 증가 목적으로 사용
-  데이터 보호 기능 없음 → 한 개의 디스크라도 고장나면 전체 데이터 손실 가능

Linear RAID 구성

```
# RAID 생성
mdadm --create /dev/md0 --level=linear --raid-devices=2 /dev/nvme0n2p1
/dev/nvme0n3p1

# 파일 시스템 생성
mkfs.xfs /dev/md0

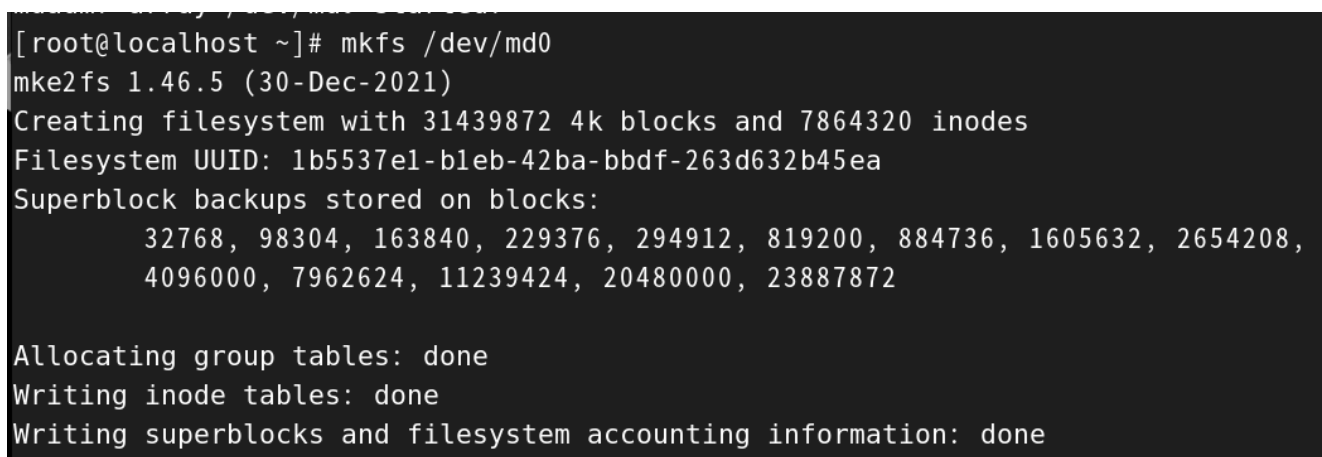
# 마운트
mount /dev/md0 /mnt
```

일반 구문:

```
mdadm --create /dev/[이름] --level=linear --raid-devices=2 /dev/[장치명1] /dev/[장치명2]
```



```
root@localhost:~
[root@localhost ~]# mdadm --create /dev/md0 --level=linear --raid-devices=2 /dev
/dev/nvme0n2p1 /dev/nvme0n3p1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@localhost ~]#
```



```
[root@localhost ~]# mkfs /dev/md0
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 31439872 4k blocks and 7864320 inodes
Filesystem UUID: 1b5537e1-b1eb-42ba-bbdf-263d632b45ea
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

```
[root@localhost ~]# mount /dev/md0 /mnt
[root@localhost ~]# df
Filesystem            1K-blocks      Used Available Use% Mounted on
devtmpfs                4096          0       4096   0% /dev
tmpfs                  1856528        0    1856528   0% /dev/shm
tmpfs                   742612      9888     732724   2% /run
/dev/mapper/rl-root    38744064 5549772 33194292  15% /
/dev/mapper/rl-home    18878464 164776 18713688   1% /home
/dev/nvme0n1p1         983040    355452     627588  37% /boot
tmpfs                   371304        100     371204   1% /run/user/0
/dev/md0               123724228      24 117436232   1% /mnt
[root@localhost ~]#
```

```
[root@localhost ~]# mdadm --detail --scan
ARRAY /dev/md0 metadata=1.2 UUID=f606f4e3:bd1a68d2:86a9cb42:c412b023
[root@localhost ~]# mdadm --detail --scan -v
ARRAY /dev/md0 level=linear num-devices=2 metadata=1.2 UUID=f606f4e3:bd1a68d2:86a9cb42:c412b023
    devices=/dev/nvme0n2p1,/dev/nvme0n3p1
[root@localhost ~]# mdadm --detail --scan /dev/md0
ARRAY /dev/md0 metadata=1.2 UUID=f606f4e3:bd1a68d2:86a9cb42:c412b023
[root@localhost ~]#
```

RAID 상태 확인 명령어




```
mdadm --detail --scan
mdadm --detail --scan -v
mdadm --detail --scan /dev/md127
```

```
Disk /dev/md127: 119.93 GiB, 128777715712 bytes, 251518976 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 524288 bytes / 1048576 bytes
```

⚠️ 주의사항: md0 으로 생성해도 재부팅하면 Rocky Linux에서 장치명을 변경할 수 있습니다. 따라서 fdisk -l 에서 장치명을 확인해야 합니다.

2-2. ⚡ RAID 0 (Striping)

특징

-  최소 2개 이상의 디스크가 필요
-  데이터를 블록 단위로 분할(Striping)하여 여러 디스크에 동시에 저장
-  데이터 보호 기능이 없음

🔧 RAID 0 구성

```
# RAID 0 생성
mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/nvme0n2p1 /dev/nvme0n3p1

# 파일 시스템 생성 및 마운트
mkfs.xfs /dev/md0
mount /dev/md0 /mnt
```

일반 구문:

```
mdadm --create /dev/[이름] --level=0 --raid-devices=2 /dev/[장치명p1] /dev/[장치명p2]
```

💡 RAID 0 동작 원리

2개의 같은 용량의 디스크가 있다고 가정할 때:

- "대한민국" 저장 시 → 디스크1[대한], 디스크2[민국]으로 분할 저장
- 만약 1개의 디스크가 손상된다면 복구할 수 없음 (복구 능력 없음)

```
[root@localhost ~]# mdadm /dev/md127 --fail /dev/nvme0n3p1
mdadm: set device faulty failed for /dev/nvme0n3p1: Device or resource busy
```

2-3. 🛡️ RAID 1 (Mirroring)

🎯 특징

- ✅ 같은 크기의 두 개 이상의 디스크에 동일한 데이터를 복제(Mirroring)하여 저장
- ✅ 데이터의 안전성이 높음 (한 개의 디스크가 고장나도 정상작동 가능)
- ❌ 저장공간 활용도가 50%

📊 예시: 60GB + 60GB = 60GB (나머지 60GB는 미러링에 사용)

🔧 RAID 1 구성

```
# RAID 1 생성
mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/nvme0n2p1 /dev/nvme0n3p1

# 파일 시스템 생성 및 마운트
mkfs.xfs /dev/md1
mount /dev/md1 /mnt
```

2-3-1. 디스크 용량이 다를 때

? 2개의 디스크 용량이 다르다면?

RAID 1은 완벽한 미러링이 필요하므로 **작은 디스크에 기준**을 둡니다.

예시: 20GB + 40GB 디스크

- 40GB 디스크는 20GB밖에 사용하지 못함
- 40GB 디스크의 나머지 20GB는 낭비됨

Hard Disk 3 (NVMe)	20 GB
Hard Disk 5 (NVMe)	40 GB
Hard Disk 4 (NVMe)	20 GB
CD/DVD (IDE)	Auto detect

```
[root@localhost ~]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs                4096          0      4096   0% /dev
tmpfs                 1856552          0  1856552   0% /dev/shm
tmpfs                  742624     9928   732696   2% /run
/dev/mapper/rl-root    38744064 5550760 33193304  15% /
/dev/mapper/rl-home    18878464 164776 18713688   1% /home
/dev/nvme0n1p1         983040    355460   627580  37% /boot
/dev/md127             125689856 910108 124779748   1% /mnt
tmpfs                  371308        100   371208   1% /run/user/0
/dev/md1               20887552 178692 20708860   1% /mnt3
```

 **결과:** 20GB로 인식합니다.

2-3-2. Spare (예비 디스크)

예시 구성:

- 20GB인 `nvme0n4`
- 40GB인 `nvme0n5`
- 20GB인 `nvme0n6` (Spare)

우선 `nvme0n4`, `nvme0n5` 를 RAID 1으로 구성 완료 후 진행합니다.

Spare 관련 명령어

```
# 장치를 고장내기
mdadm /dev/[이름] --fail /dev/[장치명p1]

# 자세한 정보 보기
mdadm --detail /dev/[이름]
```

```
# 장치 제거
mdadm /dev/[이름] --remove /dev/[장치명p1]

# 장치 추가
mdadm /dev/[이름] --add /dev/[장치명p1]
```

💡 **Spare 동작:** 디스크 1개가 고장나면 spare에 있던 디스크가 자동으로 활성화됩니다.

```
[root@localhost ~]# mdadm /dev/md1 --add /dev/nvme0n6p1
mdadm: added /dev/nvme0n6p1
[root@localhost ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Sun Feb 16 01:57:02 2025
    Raid Level : raid1
    Array Size : 20953088 (19.98 GiB 21.46 GB)
  Used Dev Size : 20953088 (19.98 GiB 21.46 GB)
    Raid Devices : 2
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Sun Feb 16 01:58:20 2025
      State : clean, resyncing
 Active Devices : 2
Working Devices : 3
```

```
Consistency Policy : resync

    Resync Status : 76% complete

    Name : localhost.localdomain:1 (local to host localhost.localdomain)
    UUID : 6a1962ca:32fcd1f0:9b058058:6b1f78f9
    Events : 17

    Number   Major   Minor   RaidDevice State   /dev/nvme0n4p1
      0       259     11       0      active sync
      1       259     12       1      active sync
      2       259     13       -      spare   /dev/nvme0n6p1

[root@localhost ~]#
```

🚨 디스크 고장 시뮬레이션

nvme0n5를 고장내면:

Consistency Policy : resync

Rebuild Status : 15% complete

복구 15% 완료

Name : localhost.localdomain:1 (local to host localhost.localdomain)

UUID : 6a1962ca:32fcd1f0:9b058058:6b1f78f9

Events : 26

Number	Major	Minor	RaidDevice	State	
0	259	11	0	active sync	/dev/nvme0n4p1
2	259	13	1	spare rebuilding	/dev/nvme0n6p1

1	259	12	-	faulty	/dev/nvme0n5p1
---	-----	----	---	--------	----------------

spare이 늘었음

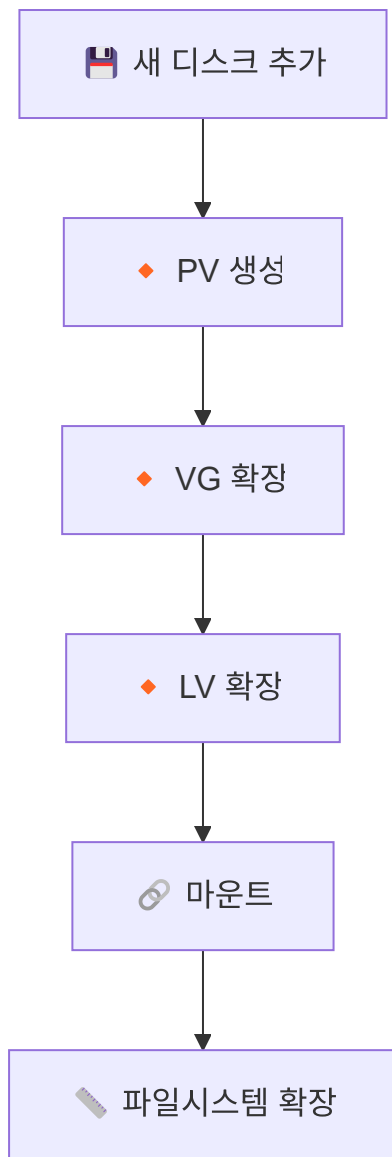
[root@localhost ~]#

```
root@localhost:~  
Raid Devices : 2  
Total Devices : 3  
Persistence : Superblock is persistent  
  
Update Time : Sun Feb 16 02:06:12 2025  
State : clean  
Active Devices : 2  
Working Devices : 2  
Failed Devices : 1  
Spare Devices : 0  
  
Consistency Policy : resync  
  
Name : localhost.localdomain:1 (local to host localhost.localdomain)  
UUID : 6a1962ca:32fcd1f0:9b058058:6b1f78f9  
Events : 41  
  
Number Major Minor RaidDevice State  
0 259 11 0 active sync /dev/nvme0n4p1  
2 259 13 1 active sync /dev/nvme0n6p1  
1 259 12 - faulty /dev/nvme0n5p1  
[root@localhost ~]#
```

복구완료

요약 정리

LVM 확장 순서



RAID 레벨 비교

RAID 레벨	최소 디스크	용량 효율성	성능	안정성	용도
Linear	2개	100%	보통	낮음	용량 확장
RAID 0	2개	100%	높음	낮음	성능 중심
RAID 1	2개	50%	보통	높음	안정성 중심
RAID 5	3개	67%~94%	높음	중간	균형잡힌 구성

 **LVM & RAID 마스터하기!** 이 가이드를 통해 Linux에서 효율적인 스토리지 관리를 구현해보세요!