

Linux 3.2

[Linux 3.1](#)



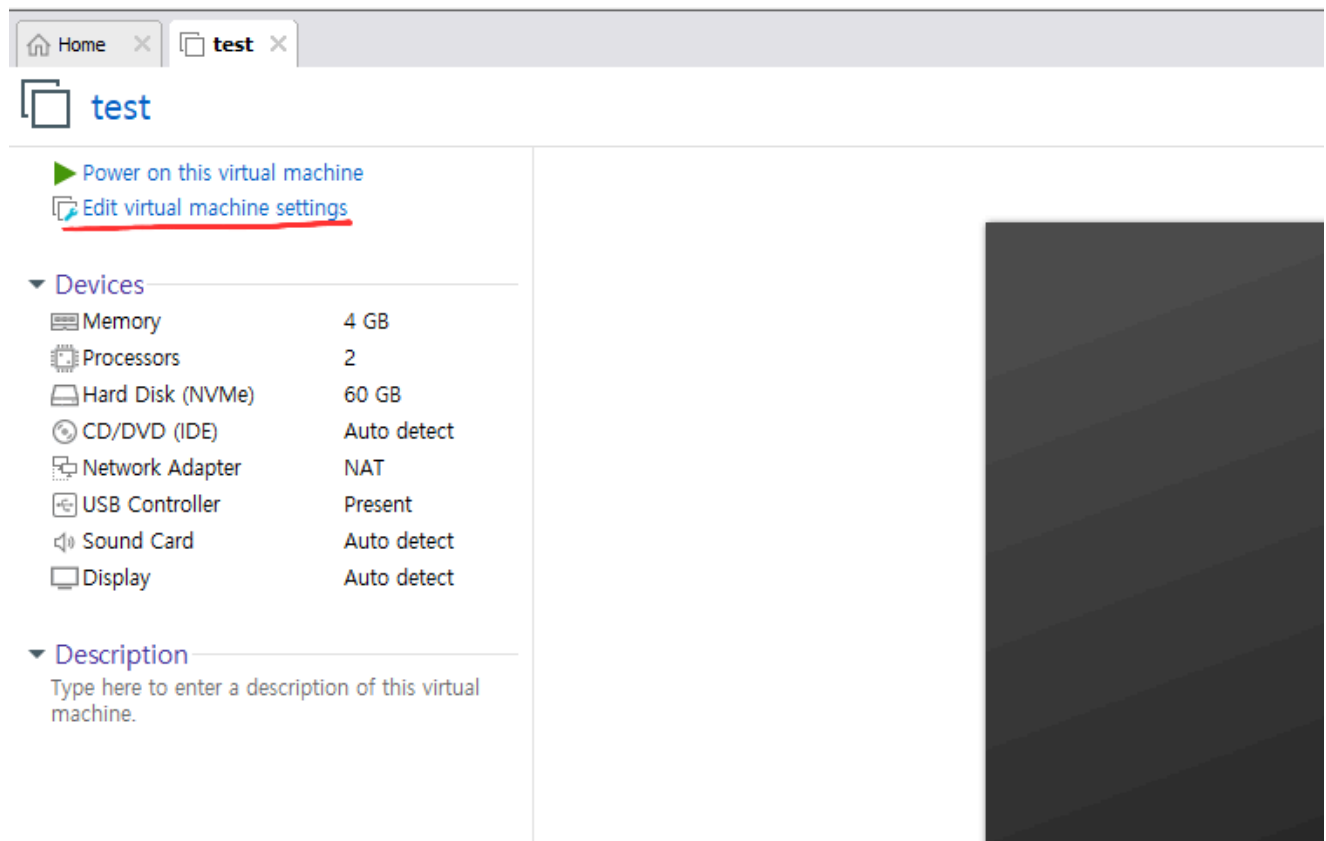
Linux 디스크 관리 및 LVM 완벽 가이드

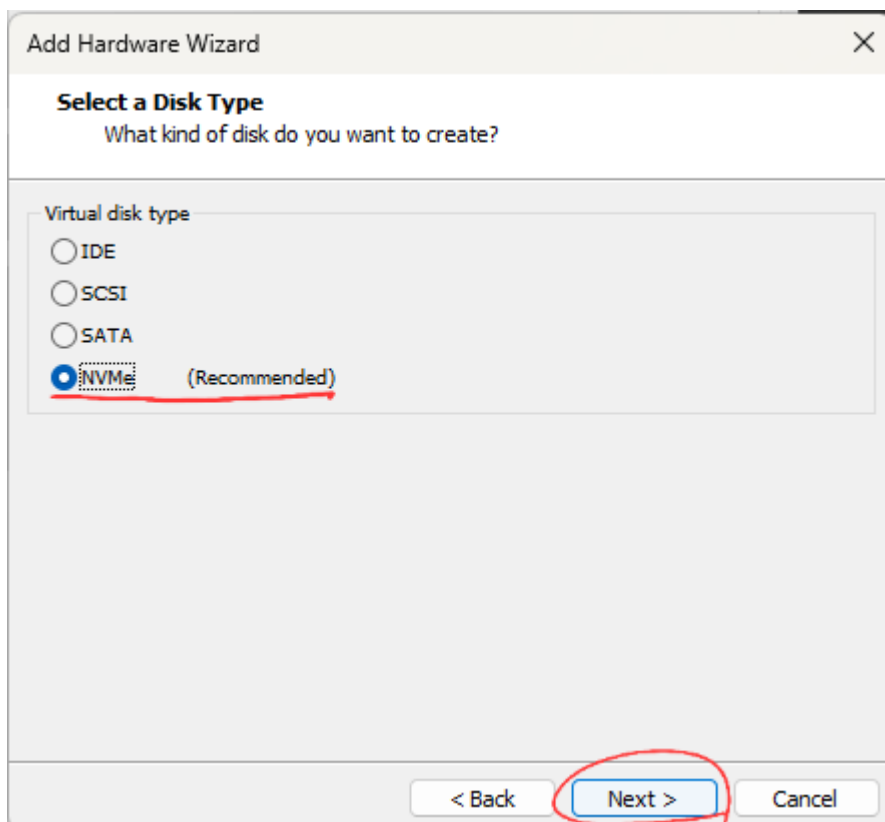
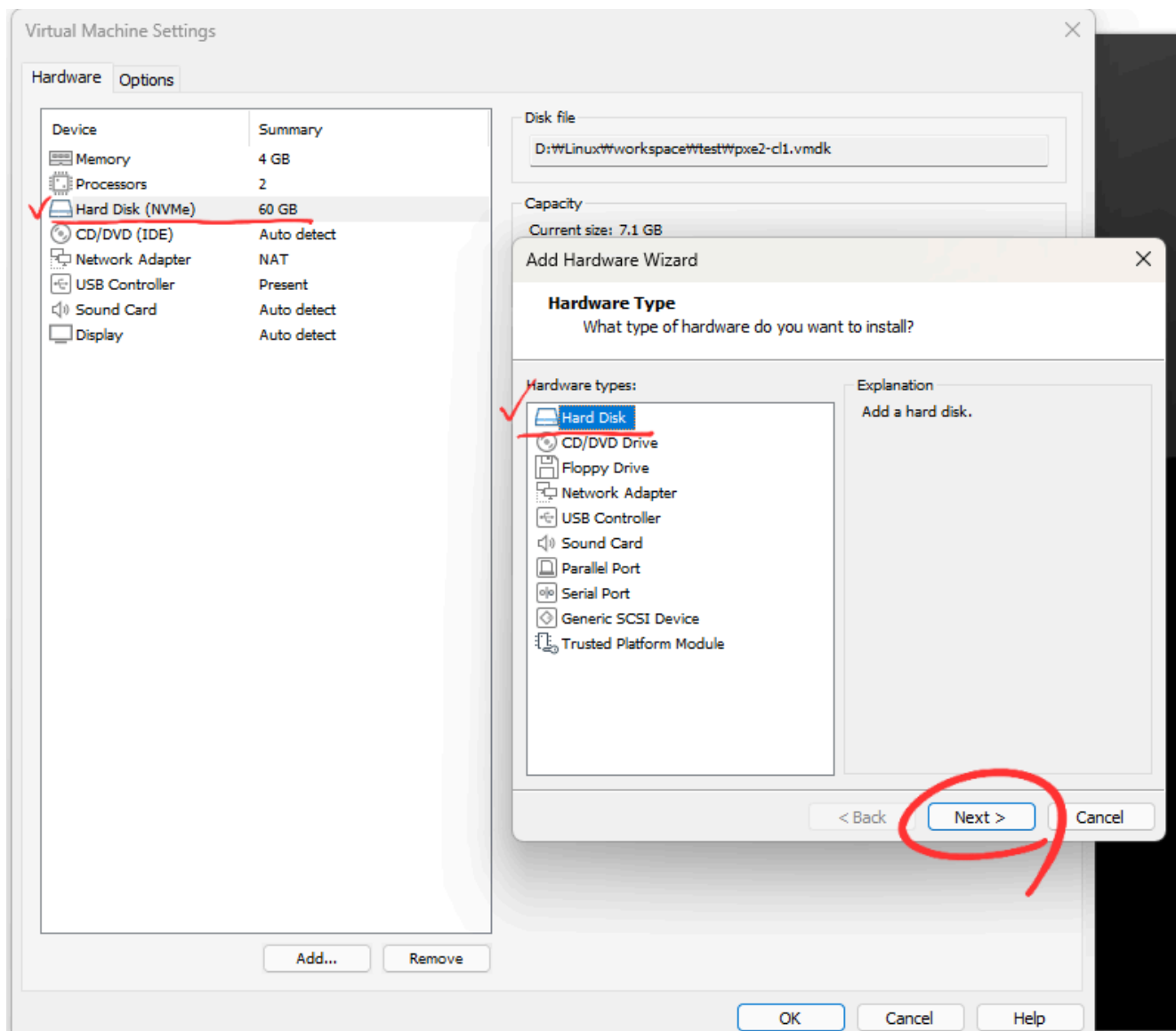


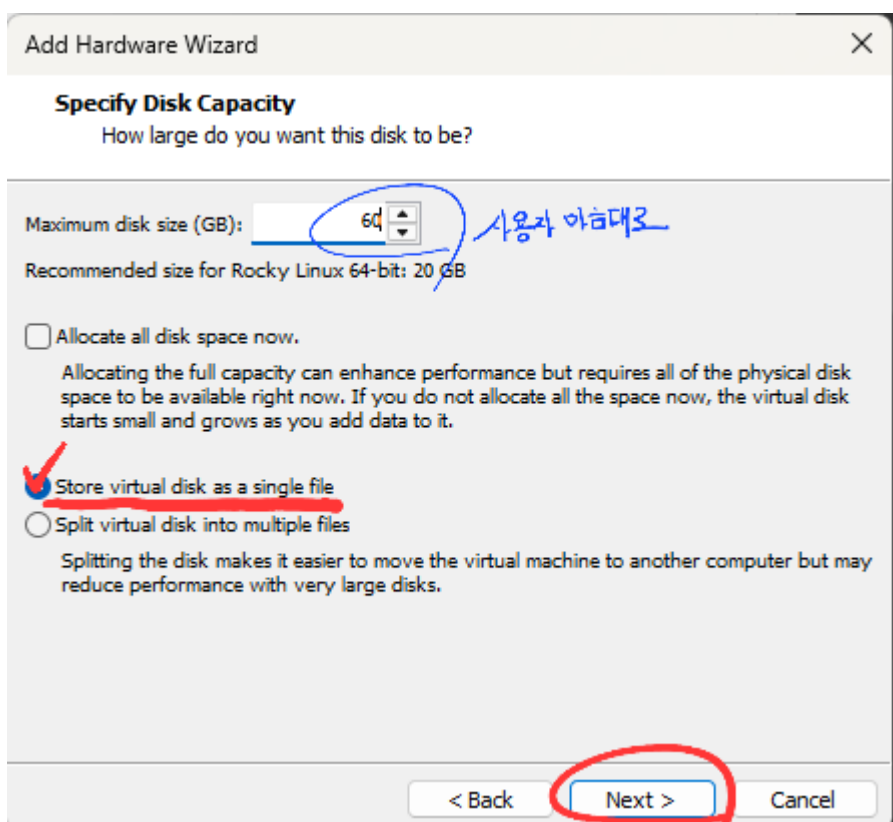
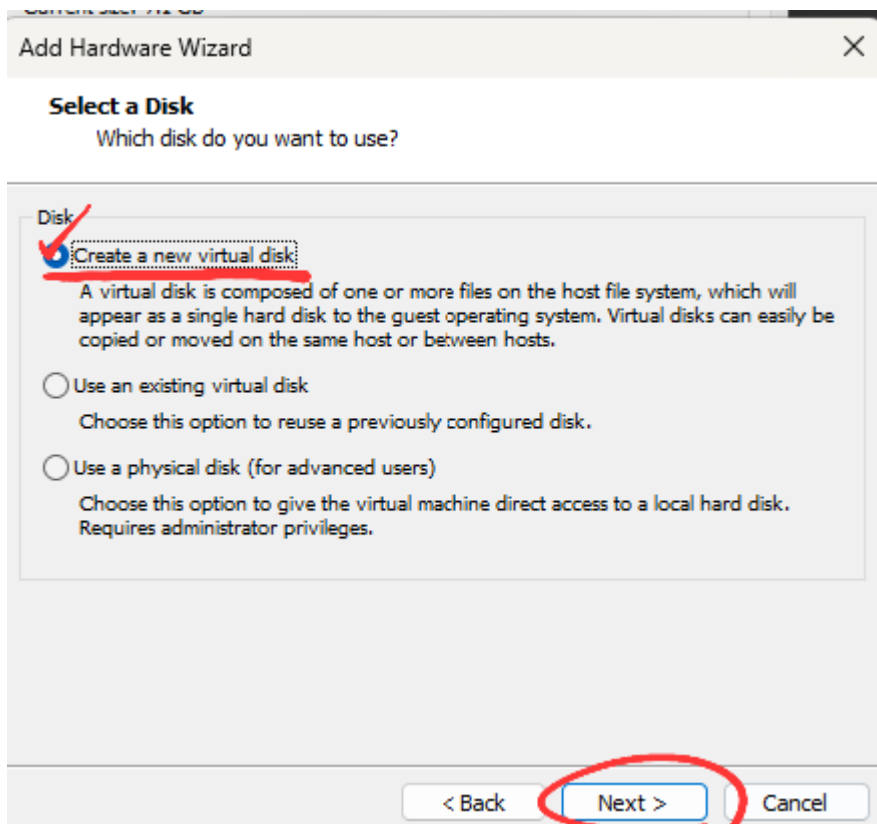
1. 디스크 장착 및 마운트

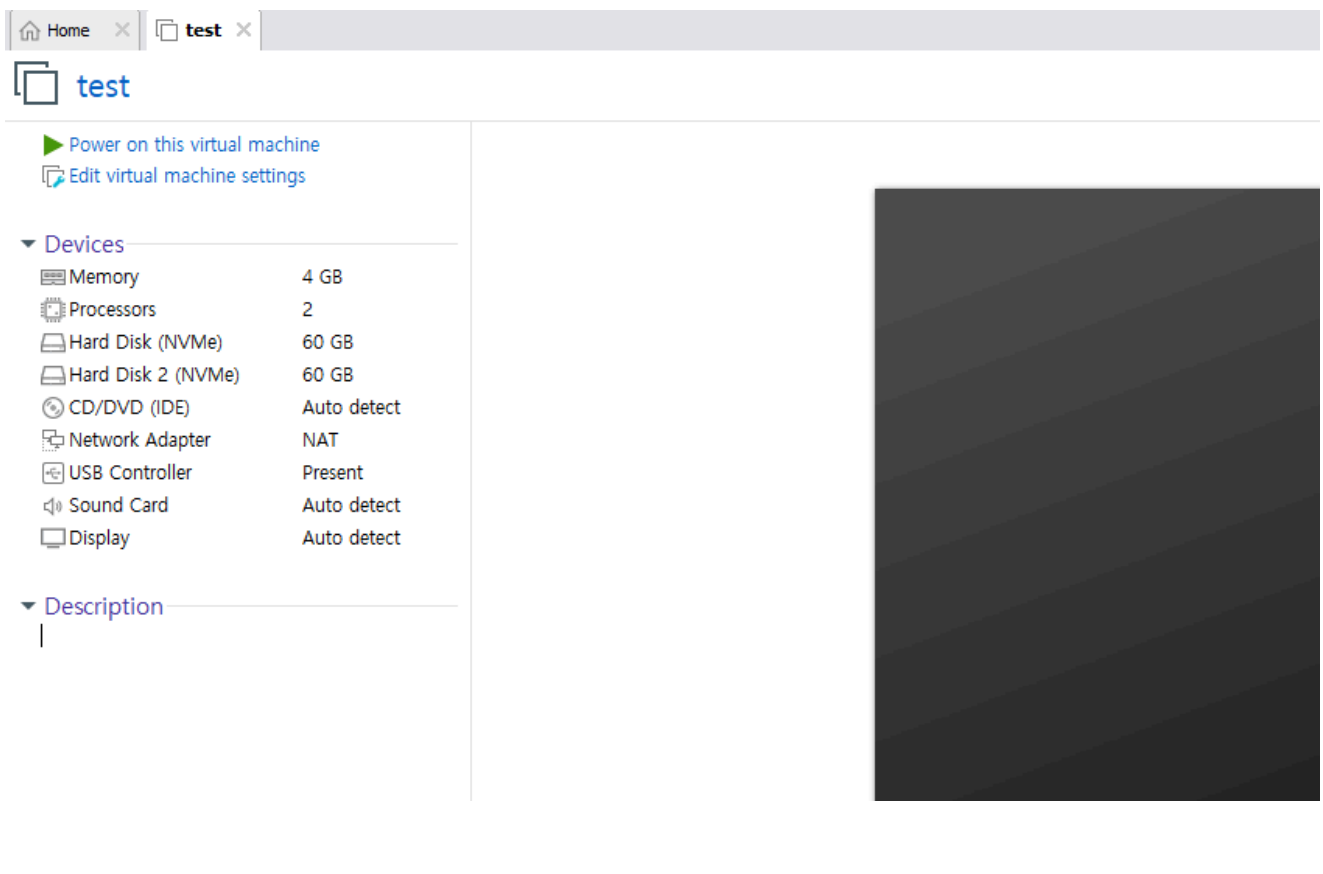
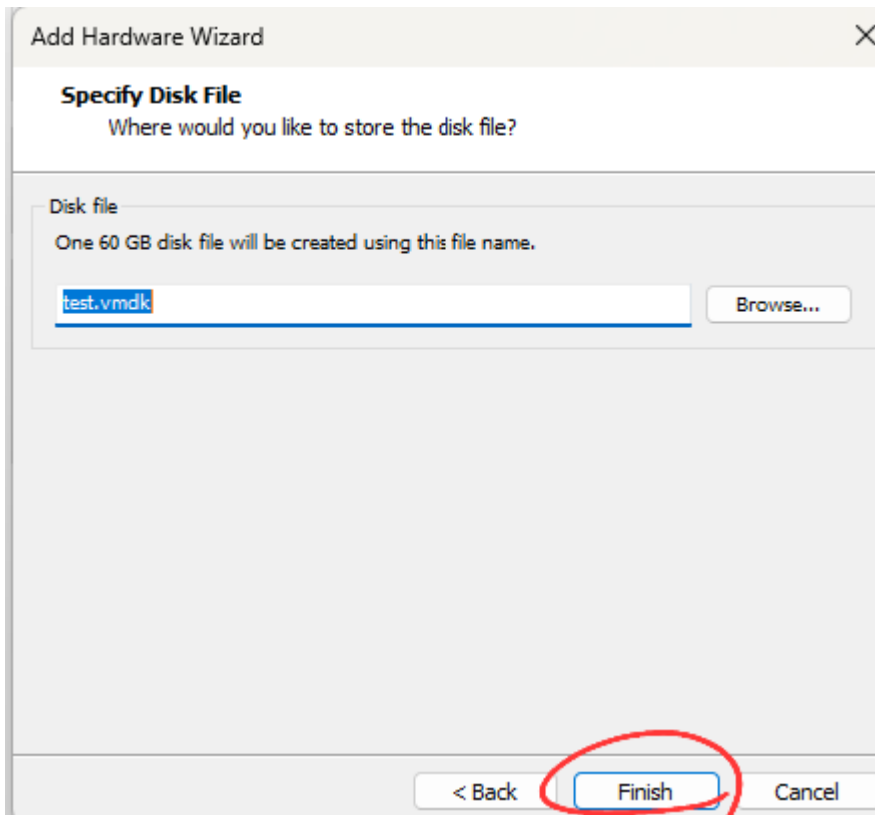
1-1. 디스크 장착 (VMware)

VMware에서 새로운 디스크를 추가하는 과정입니다.









1-2. 🔧 파티션 나누기

첫 번째로, 추가한 장치명을 알아야 합니다.

🔍 fdisk 명령어

HDD, SSD, USB 드라이브 같은 저장 장치의 **파티션 생성, 삭제, 수정** 등을 할 수 있습니다.

기본 구문:

fdisk [옵션] [명령어]

장치명과 파티션 유무 등을 확인

fdisk -l

```
root@localhost:~  
[root@localhost ~]# fdisk -l  
Disk /dev/nvme0n1: 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: 0xe462a11e  
  
Device            Boot    Start        End    Sectors  Size Id Type  
/dev/nvme0n1p1    *          2048     2099199    2097152    1G 83 Linux  
/dev/nvme0n1p2          2099200 125829119 123729920    59G 8e Linux LVM  
  
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  
  
Disk /dev/mapper/rl-root: 37.01 GiB, 39741030400 bytes, 77619200 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes
```

장치 정보 분석

```
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
```

💡 **분석 결과:** 장치명이 `nvme0n2` 이며, 60GB이고 현재 파티션이 없는 상태

파티션 생성 과정

fdisk /dev/nvme0n2

```
root@localhost:~ — fdisk /dev/nvme0n2

[root@localhost ~]# fdisk /dev/nvme0n2

Welcome to fdisk (util-linux 2.37.4).
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.
Created a new DOS disklabel with disk identifier 0x1bacee86.

Command (m for help): m

Help:

DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
```

명령어 실행 후 `m` 입력

주요 fdisk 명령어

명령어	설명
<code>l</code>	list known partition types
<code>n</code>	add a new partition
<code>p</code>	print the partition table
<code>t</code>	change a partition type

```

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p):

Using default response p.
Partition number (1-4, default 1):
First sector (2048-125829119, default 2048):
Last sector, +/-sectors or +/-size[K,M,G,T,P] (2048-125829119, default 125829119):

Created a new partition 1 of type 'Linux' and of size 60 GiB.

Command (m for help):

```

⚡ **진행 방법:** 이 단계가 올 때까지 계속 Enter를 입력합니다.

```

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: 0x1bacee86

Device            Boot Start      End  Sectors  Size Id Type
/dev/nvme0n2p1    2048 125829119 125827072   60G 83 Linux

Command (m for help):

```

p를 눌러 파티션 테이블을 출력합니다.

```

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

[root@localhost ~]#

```

w로 변경사항을 저장하고 fdisk를 종료합니다.

1-2-1. 📁 리눅스의 파티션 정보

◆ Primary Partition (기본 파티션)

- ✓ 최대 4개까지 생성 가능
- ✓ 운영체제가 직접 접근 가능한 기본적인 파티션

◆ Extended Partition (확장 파티션)

- ☒ 파티션 중 하나를 Extended로 설정하면, 그 안에서 여러개의 논리 파티션 생성 가능
- ☒ Primary Partition을 초과하는 공간을 확보할 때 필요

◆ Logical Partition (논리 파티션)

- ☒ Extended Partition 내부에서 생성되는 파티션
- ☒ 5번째 파티션부터는 Logical로 할당

1-3. 🗄 파일 시스템 생성 (포맷)

기본 구문:

```
mkfs.xfs /dev/nvme0n2p1
```

```
[root@localhost ~]# mkfs.xfs /dev/nvme0n2p1
meta-data=/dev/nvme0n2p1          isize=512    agcount=4, agsize=3932096 blks
        =                       sectsz=512    attr=2, projid32bit=1
        =                       crc=1        finobt=1, sparse=1, rmapbt=0
        =                       reflink=1     bigtime=1 inobtcount=1 nrext64=0
data      =                       bsize=4096   blocks=15728384, 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
[root@localhost ~]#
```

1-4. 🔗 마운트

기본 구문:

```
mount [장치명] [마운트포인트]
mount /dev/nvme0n2p1 /mnt
```



```
[root@localhost ~]# mount /dev/nvme0n2p1 /mnt
[root@localhost ~]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs                4096         0      4096    0% /dev
tmpfs                  1856544       0    1856544    0% /dev/shm
tmpfs                   742620     9872     732748    2% /run
/dev/mapper/rl-root    38744064 4758192   33985872   13% /
/dev/mapper/rl-home    18878464  164776   18713688    1% /home
/dev/nvme0n1p1         983040   355452     627588   37% /boot
tmpfs                   371308      104     371204    1% /run/user/0
/dev/nvme0n2p1        62848000 471224   62376776    1% /mnt
[root@localhost ~]#
```

디스크 용량 확인

df 명령어 - 파일 시스템 단위 용량 확인

기본 구문: `df [옵션]`

주요 옵션:

- `-T`: 파일 시스템 유형 표시
- `-h`: 사용자가 읽기 쉽게 용량 출력

```
root@localhost:~
[root@localhost ~]# df -T
Filesystem            Type      1K-blocks    Used Available Use% Mounted on
devtmpfs              devtmpfs     4096         0      4096    0% /dev
tmpfs                 tmpfs       1856544       0    1856544    0% /dev/shm
tmpfs                 tmpfs       742620     9868     732752    2% /run
/dev/mapper/rl-root   xfs       38744064 4758172   33985892   13% /
/dev/mapper/rl-home   xfs       18878464  164776   18713688    1% /home
/dev/nvme0n1p1        xfs       983040   355452     627588   37% /boot
tmpfs                 tmpfs       371308      104     371204    1% /run/user/0
/dev/nvme0n2p1        xfs      62848000 471224   62376776    1% /mnt
[root@localhost ~]# df -h
Filesystem            Size  Used Avail Use% Mounted on
devtmpfs              4.0M   0  4.0M   0% /dev
tmpfs                 1.8G   0  1.8G   0% /dev/shm
tmpfs                 726M  9.7M  716M   2% /run
/dev/mapper/rl-root   37G   4.6G   33G  13% /
/dev/mapper/rl-home   19G  161M   18G   1% /home
/dev/nvme0n1p1        960M  348M  613M  37% /boot
tmpfs                 363M  104K  363M   1% /run/user/0
/dev/nvme0n2p1        60G  461M   60G   1% /mnt
[root@localhost ~]#
```

du 명령어 - 디렉토리 단위 용량 확인

기본 구문: `du [옵션]`

주요 옵션:

- `-h`: 사용자가 읽기 쉽게 용량 출력

```
root@localhost:~  
[root@localhost ~]# du -h  
0      ./ssh  
0      ./cache/ibus  
0      ./cache/evolution/addressbook/trash  
0      ./cache/evolution/addressbook  
0      ./cache/evolution/calendar/trash  
0      ./cache/evolution/calendar  
0      ./cache/evolution/mail/trash  
0      ./cache/evolution/mail  
0      ./cache/evolution/memos/trash  
0      ./cache/evolution/memos  
0      ./cache/evolution/sources/trash  
0      ./cache/evolution/sources  
0      ./cache/evolution/tasks/trash  
0      ./cache/evolution/tasks  
0      ./cache/evolution  
540K   ./cache/gstreamer-1.0  
3.3M   ./cache/gnome-software/appstream  
4.0K   ./cache/gnome-software/flatpak-system-default  
1.7M   ./cache/gnome-software/odrs  
4.0K   ./cache/gnome-software/flatpak-user-user  
5.0M   ./cache/gnome-software  
504K   ./cache/thumbnails/large  
32K    ./cache/thumbnails/fail/gnome-thumbnail-factory
```

1-5. 🔒 언마운트

기본 구문:

```
# 마운트포인트로 언마운트  
umount /mnt  
  
# 장치명으로 언마운트  
umount /dev/nvme0n2p1
```

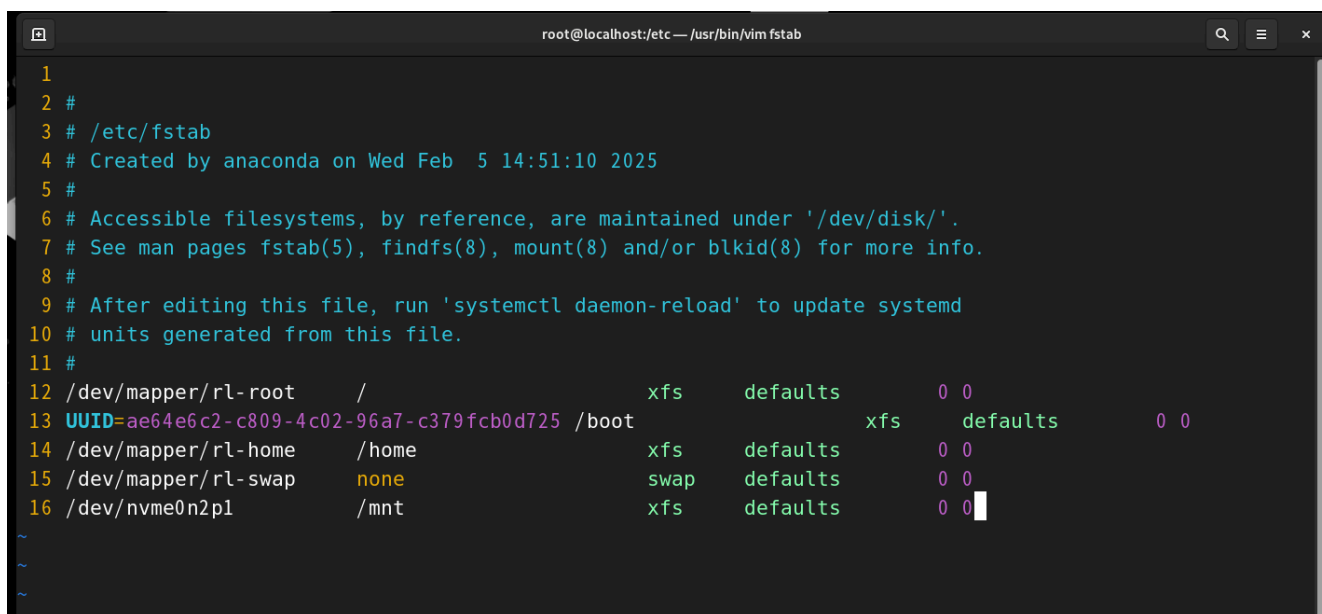
```

root@localhost ~]# umount /dev/nvme0n2p1
root@localhost ~]# df
Filesystem            1K-blocks      Used Available Use% Mounted on
devtmpfs                4096          0      4096   0% /dev
tmpfs                  1856536          0    1856536   0% /dev/shm
tmpfs                   742616     9876     732740   2% /run
/dev/mapper/rl-root    38744064 4758548   33985516  13% /
/dev/mapper/rl-home    18878464  164776   18713688   1% /home
/dev/nvme0n1p1         983040   355452     627588  37% /boot
tmpfs                   371304         92     371212   1% /run/user/0
root@localhost ~]#

```

1-6. 자동마운트

`mount` 명령어로 마운트하고 재부팅을 하게 되면, 다시 마운트해야 하는 번거로움이 있습니다. 이를 해결하기 위해서 **/etc/fstab** 파일을 편집하면 재부팅하더라도 마운트가 유지됩니다.



```

1
2 #
3 # /etc/fstab
4 # Created by anaconda on Wed Feb  5 14:51:10 2025
5 #
6 # Accessible filesystems, by reference, are maintained under '/dev/disk/'.
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
8 #
9 # After editing this file, run 'systemctl daemon-reload' to update systemd
10 # units generated from this file.
11 #
12 /dev/mapper/rl-root    /                    xfs     defaults        0 0
13 UUID=ae64e6c2-c809-4c02-96a7-c379fcb0d725 /boot                xfs     defaults        0 0
14 /dev/mapper/rl-home    /home               xfs     defaults        0 0
15 /dev/mapper/rl-swap     none                swap    defaults        0 0
16 /dev/nvme0n2p1         /mnt                 xfs     defaults        0 0
~
~
~

```

/etc/fstab 설정 형식

[장치명] [마운트포인트] [파일시스템유형] [defaults] [덤프여부] [파일시스템검사여부]

```

# 예제
/dev/nvme0n2p1 /mnt xfs defaults 0 0

```

2. 파티션 통합 기술

물리적으로 100TB, 100TB가 있고, 논리적으로 200TB로 합치는 기술 2개가 존재합니다.

통합 기술 종류

- 기술1: LVM (Logical Volume Manager) - 리눅스만 지원
- 기술2: RAID (Redundant Array of Independent Disks)

2-1. 🌀 LVM (Logical Volume Manager)

HDD 60GB 2개를 논리적으로 묶는 과정:

1. 각각을 **Physical Volume**로 변환
2. 물리적으로 묶기 (**Volume Grouping**)
3. **Logical Volume**로 120GB 생성
4. **File System** 생성
5. **마운트** 진행

🔧 사전 준비 - HDD 60GB 2개 장착

▼ Devices	
Memory	4 GB
Processors	2
Hard Disk (NVMe)	60 GB
Hard Disk 2 (NVMe)	60 GB
Hard Disk 4 (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
▼ Description	

```
root@localhost:~  
  
Device          Boot Start      End    Sectors  Size Id Type  
/dev/nvme0n2p1   2048 125829119 125827072  60G 83 Linux  
  
Disk /dev/nvme0n3: 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  
  
Disk /dev/nvme0n4: 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
```

💡 확인된 장치명: /dev/nvme0n3 , /dev/nvme0n4

📄 파티션 작업 및 LVM 타입 설정

```
fdisk /dev/nvme0n3  
fdisk /dev/nvme0n4
```

⚠️ **중요:** ID를 **8E (LVM)**으로 변경해야 합니다.

- t 를 누르고 8e 입력 (대소문자 구분 없음)
- ID가 83번에서 8E로 변경
- 타입도 "Linux LVM"으로 변경됨

```

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 8e
Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): p
Disk /dev/nvme0n3: 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: 0x9dd5a273

Device                Boot Start      End  Sectors  Size Id Type
/dev/nvme0n3p1         2048 125829119 125827072   60G 8e Linux LVM

Command (m for help): █

```

2-2. ♦ PV (Physical Volume) 생성

기본 구문:

```

# 개별 생성
pvcreate /dev/nvme0n3p1
pvcreate /dev/nvme0n4p1

# 한번에 2개 생성
pvcreate /dev/nvme0n3p1 /dev/nvme0n4p1

```

✓ **성공 확인:** "successfully created" 문구가 나타나야 합니다.

```

[root@localhost ~]# pvcreate /dev/nvme0n3p1 /dev/nvme0n4p1
Physical volume "/dev/nvme0n3p1" successfully created.
Physical volume "/dev/nvme0n4p1" successfully created.
[root@localhost ~]# █

```

2-3. ♦ VG (Volume Group) 생성

기본 구문:

```

vgcreate [VG그룹명] [장치명1] [장치명2]...

```

```
# 예제
vgcreate myvg /dev/nvme0n3p1 /dev/nvme0n4p1
```

```
[root@localhost ~]# vgcreate myvg /dev/nvme0n3p1 /dev/nvme0n4p1
Volume group "myvg" successfully created
[root@localhost ~]#
```

💡 논리적으로 바꿔주는 작업을 해야 사용할 수 있습니다.

2-4. LV (Logical Volume) 생성

기본 구문:

```
lvcreate -n [LV그룹명] -l 100%FREE [VG그룹명]
```

```
# 예제
lvcreate -n mylv -l 100%FREE myvg
```

```
Volume group "myvg" successfully created
[root@localhost ~]# lvcreate -n mylv -l 100%FREE myvg
Logical volume "mylv" created.
[root@localhost ~]#
```

2-5. 파일 시스템 생성

기본 구문:

```
mkfs.xfs /dev/[VG그룹명]/[LV그룹명]
```

```
# 예제
mkfs.xfs /dev/myvg/mylv
```

```
[root@localhost ~]# mkfs.xfs /dev/myvg/mylv
meta-data=/dev/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
```

2-6. 🔗 마운트

기본 구문:

```
mount /dev/[VG그룹명]/[LV그룹명] [마운트포인트]
```

예제

```
mount /dev/myvg/mylv /mnt
```

```
[root@localhost ~]# mount /dev/myvg/mylv /root/mnt2
[root@localhost ~]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
devtmpfs                4096         0      4096   0% /dev
tmpfs                  1856548       0    1856548   0% /dev/shm
tmpfs                   742620     9900     732720   2% /run
/dev/mapper/rl-root    38744064 5129176 33614888 14% /
/dev/nvme0n1p1         983040    355452   627588 37% /boot
/dev/mapper/rl-home    18878464 164776 18713688  1% /home
/dev/nvme0n2p1         62848000 471224 62376776  1% /mnt
tmpfs                   371308      100     371208   1% /run/user/0
/dev/mapper/myvg-mylv 125755392 909816 124845576  1% /root/mnt2
```



```
/dev/nvme0n2p1          /mnt                xfs      defaults    0 0
/dev/myvg/mylv          /root/mnt2          xfs      defaults    0 0
```

💡 재부팅해도 자동 마운트가 되도록 /etc/fstab 에 추가합니다.

2-7. 📋 LVM 구성 순서 정리

Error parsing Mermaid diagram!

Parse error on line 3:




```
... B --> C[🔄 ID를 8E(LVM)으로 변경] C --
```


-----^

Expecting 'SQE', 'DOUBLECIRCLEEND', 'PE', '(-)', 'STADIUMEND', 'SUBROUTINEEND',
'PIPE', 'CYLINDEREND', 'DIAMOND_STOP', 'TAGEND', 'TRAPEND', 'INVTRAPEND',
'UNICODE_TEXT', 'TEXT', 'TAGSTART', got 'PS'

✨ 단계별 명령어 요약

- 🔧 하드 디스크 추가
- 📁 파티션 작업 진행
- 🔄 작업 완료 후, ID를 8E(LVM)으로 변경
- ✅ p 를 누르고, 변경되었는지 확인
- ♦ PV 생성 → pvcreate
- ♦ VG 생성 → vgcreate

7.  **LV 생성** → `lvcreate`
 8.  **파일 시스템 생성** → `mkfs.xfs /dev/[VG그룹명]/[LV그룹명]`
 9.  **마운트** → `mount`
-

 **LVM 마스터하기!** 이 가이드를 통해 Linux에서 디스크를 효율적으로 관리하고 논리 볼륨을 구성해 보세요!