

Standard Partition:

☒ Offline Method (Normal Partitioning):

In traditional partitioning, extending a partition is an offline activity, meaning the partition must be unmounted first.

Here's how it works step-by-step:

☒ Normal Partition Extension (Offline):

1. Add a New Disk in VMware:

Go to VM Settings

Click Add → Select Hard Disk

Configure the disk (size, type, etc.)

Click Finish to add the disk

```
root@suria:~  
[root@suria ~]# lsblk | grep sdb  
sdb      8:16    0    20G    0 disk  
[root@suria ~]# fdisk -l /dev/sdb  
Disk /dev/sdb: 20 GiB, 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@suria ~]# █
```

2. Access the disk using fdisk:

fdisk /dev/sdb

```
root@suria:~  
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  
t change a partition type  
v verify the partition table  
i print information about a partition  
  
Misc  
m print this menu  
u change display/entry units  
x extra functionality (experts only)  
  
Script  
I load disk layout from sfdisk script file  
O dump disk layout to sfdisk script file  
  
Save & Exit  
w write table to disk and exit  
q quit without saving changes  
  
Create a new label  
g create a new empty GPT partition table  
G create a new empty SGI (IRIX) partition table  
o create a new empty DOS partition table  
s create a new empty Sun partition table
```

3. Create a new partition:

Inside fdisk, follow these prompts:

Press n (new partition)

Press p (primary)

Enter partition number (default is fine)

Press Enter for starting sector (default)

Enter size (e.g., +1G, +2G, etc.)

Press w (write and exit)

root@suria:~

```
[root@suria ~]# fdisk /dev/sdb
```

Welcome to fdisk (util-linux 2.32.1).

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 0xc8c32930.

Command (m for help): n

Partition type

p primary (0 primary, 0 extended, 4 free)
e extended (container for logical partitions)

Select (default p): p

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

First sector (2048-41943039, default 2048):

Last sector, +sectors or +size{K,M,G,T,P} (2048-41943039, default 41943039): +1G

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

Command (m for help): w

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

```
[root@suria ~]# q
```

root@suria:/

```
[root@suria /]#
```

```
[root@suria /]# fdisk /dev/sdb
```

Welcome to fdisk (util-linux 2.32.1).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n

Partition type

p primary (1 primary, 0 extended, 3 free)
e extended (container for logical partitions)

Select (default p): p

Partition number (2-4, default 2):

First sector (2099200-41943039, default 2099200):

Last sector, +sectors or +size{K,M,G,T,P} (2099200-41943039, default 41943039): +1G

Created a new partition 2 of type 'Linux' and of size 1 GiB.

Command (m for help): t

Partition number (1,2, default 2):

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

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

Command (m for help): w

The partition table has been altered.
Syncing disks.

```
[root@suria /]#
```

4. Inform the kernel about partition changes:

`partprobe /dev/sdb`

5. Format the new partition:

`mkfs.xfs /dev/sdb1` (XFS filesystem)

`mkfs.ext4 /dev/sdb2` (ext4 filesystem)

```
root@suria:~  
[root@suria ~]# lsblk | grep sdb  
sdb            8:16    0    20G  0 disk  
└─sdb1         8:17    0     1G  0 part  
[root@suria ~]#  
[root@suria ~]# partprobe /dev/sdb  
[root@suria ~]#  
[root@suria ~]# mkfs.xfs /dev/sdb1  
meta-data=/dev/sdb1            isize=512    agcount=4, agsize=65536 blks  
      =                       sectsz=512    attr=2, projid32bit=1  
      =                       crc=1        finobt=1, sparse=1, rmapbt=0  
      =                       reflink=1    bigtime=0 inobtcount=0  
data      =                   bsize=4096    blocks=262144, 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@suria ~]#  
[root@suria ~]# blkid -o list | grep sdb  
/dev/sdb1 xfs                 (not mounted)  75394572-62fd-4509-a9c1-6e61f63655d1  
[root@suria ~]#
```

6. Create a mount point directory:

`mkdir /data`

`mkdir /data2`

root@suria:~

```
[root@suria ~]# mkdir data
[root@suria ~]# mount /dev/sdb1 /data
[root@suria ~]#
[root@suria ~]# df -h | grep sdb
/dev/sdb1          1014M    40M   975M    4% /data
[root@suria ~]#
```

root@suria:~

```
[root@suria ~]#
[root@suria ~]# partprobe /dev/sdb
[root@suria ~]#
[root@suria ~]# lsblk | grep sdb2
└─sdb2            8:18    0    1G  0 part
[root@suria ~]#
[root@suria ~]# mkfs.ext4 /dev/sdb2
mke2fs 1.45.6 (20-Mar-2020)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: b210e054-61ad-4e8f-9c38-4b50fa032977
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@suria ~]#
[root@suria ~]# blkid -o list | grep sdb2
/dev/sdb2 ext4          (not mounted)  b210e054-61ad-4e8f-9c38-4b50fa032977
[root@suria ~]#
[root@suria ~]#
```

7. Mount the partition:

```
mount /dev/sdb1 /data
```

```
mount /dev/sdb2 /data2
```

8. Verify the partition is mounted:

```
df -h
```

lsblk

9. Add Entry to /etc/fstab (Persistent Mount):

blkid /dev/sdb1 /dev/sdb2

10. Open the fstab file for editing:

vi nano /etc/fstab

UUID=abcd-1234-efgh-5678 /data xfs defaults 0 0

UUID=ijkl-8989-mnop-3434 /data2 ext4 defaults 0 1

```
root@suria:/  
[root@suria /]# vi /etc/fstab  
[root@suria /]#  
[root@suria /]# cat /etc/fstab | grep data1  
75394572-62fd-4509-a9c1-6e61f63655d1    /data1  xfs      defaults    0 0  
[root@suria /]#
```

```

root@suria:/
[root@suria /]#
[root@suria /]# mkdir data2
[root@suria /]#
[root@suria /]# mount /dev/sdb2 /data2
[root@suria /]#
[root@suria /]# blkid -o list | grep sdb2
/dev/sdb2 ext4 /data2 b210e054-61ad-4e8f-9c38-4b50fa032977
[root@suria /]#
[root@suria /]# df -h | grep sdb2
/dev/sdb2 974M 24K 907M 1% /data2
[root@suria /]#
[root@suria /]# vi /etc/fstab
[root@suria /]#
[root@suria /]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Mon Apr 21 21:12:18 2025
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel-root / xfs defaults 0 0
UUID=31e80e7f-9d38-4823-b2b0-498078c98d31 /boot xfs defaults 0 0
/dev/mapper/rhel-swap none swap defaults 0 0
75394572-62fd-4509-a9c1-6e61f63655d1 /data1 xfs defaults 0 0
/dev/sdb2 /data2 ext4 defaults 1 2
[root@suria /]# d
-bash: d: command not found
[root@suria /]#

```

Test the fstab entry:

`mount -a`

Note:

The command for unmounting a filesystem in Linux:

`umount /data`