**Logical Volume Manager (LVM)**

**Mục lục**

[1. Giới thiệu về Logical Volume Manager (LVM)] (#Gioithieu)

* [1.1 LVM là gì] (#LVM)
* [1.2 Vai trò của LVM] (#Vaitro)
* [1.3 Các thành phần trong LVM] (#Thanhphan)

[2. Hướng dẫn sử dụng LVM] (#Huongdan)

* [2.1 Chuẩn bị] (#Chuanbi)
* [2.2 Tạo Logical Volume trên LVM] (#Taolvm)
* [2.3 Thay đổi dung lượng Logical Volume trên LVM] (#Thaydoilv)
* [2.4 Thay đổi dung lượng Volume Group trên LVM] (#Thaydoivg)
* [2.5 Xóa Logical Volume, Volume Group, Physical Volume] (#Xoa)

[3. Tổng kết] (#Tongket)

**1.Giới thiệu về Logical Volume Manager (LVM)**

###1.1 LVM là gì

Logical Volume Manager (LVM): là phương pháp cho phép ấn định không gian đĩa cứng thành những logical Volume khiến cho việc thay đổi kích thước trở nên dễ dàng hơn (so với partition). Với kỹ thuật Logical Volume Manager (LVM) bạn có thể thay đổi kích thước mà không cần phải sửa lại table của OS. Điều này thật hữu ich với những trường hợp bạn đã sử dụng hết phần bộ nhớ còn trống của partition và muốn mở rộng dung lượng của nó

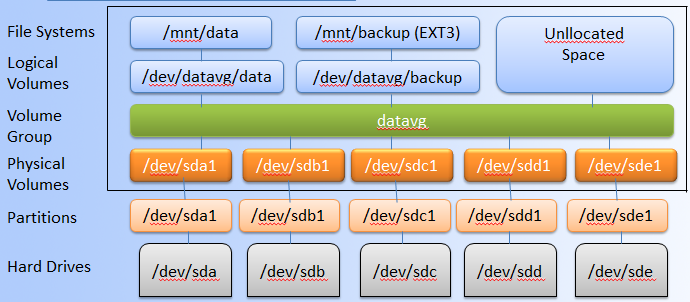
###1.2 Vai trò của LVM

LVM là kỹ thuật quản lý việc thay đổi kích thước lưu trữ của ổ cứng

* Không để hệ thống bị gián đoạn hoạt động
* Không làm hỏng dịch vụ
* Có thể kết hợp Hot Swapping (thao tác thay thế nóng các thành phần bên trong máy tính)

###1.3 Các thành phần trong LVM

**Mô hình các thành phần trong LVM**

[](https://camo.githubusercontent.com/713a3058b8a31f2686108f71d0ba494fc8317adb/687474703a2f2f692e696d6775722e636f6d2f556154617475622e706e67)

**Hard drives – Drives**

Thiết bị lưu trữ dữ liệu, ví dụ như trong linux nó là /dev/sda

**Partition**

Partitions là các phân vùng của Hard drives, mỗi Hard drives có 4 partition, trong đó partition bao gồm 2 loại là primary partition và extended partition

* **Primary partition:**
* Phân vùng chính, có thể khởi động
* Mỗi đĩa cứng có thể có tối đa 4 phân vùng này
* **Extended partition:**
* Phân vùng mở rộng, có thể tạo những vùng luân lý

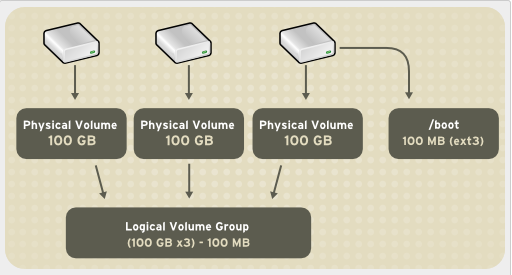
**Physical Volumes**

Là một cách gọi khác của partition trong kỹ thuật LVM, nó là những thành phần cơ bản được sử dụng bởi LVM. Một Physical Volume không thể mở rộng ra ngoài phạm vi một ổ đĩa.

Chúng ta có thể kết hợp nhiều Physical Volume thành Volume Groups

**Volume Group**

Nhiều Physical Volume trên những ổ đĩa khác nhau được kết hợp lại thành một Volume Group

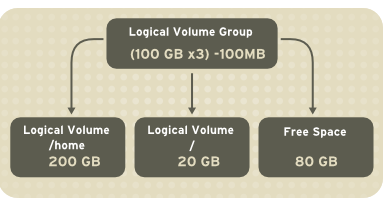
[](https://camo.githubusercontent.com/7f4dbca82d3d0590cca078dea173360c6725667c/687474703a2f2f692e696d6775722e636f6d2f6a496371744d532e706e67)

Volume Group được sử dụng để tạo ra các Logical Volume, trong đó người dùng có thể tạo, thay đổi kích thước, lưu trữ, gỡ bỏ và sử dụng.

*Một điểm cần lưu ý là boot loader không thể đọc /boot khi nó nằm trên Logical Volume Group. Do đó không thể sử dụng kỹ thuật LVM với /boot mount point*

**Logical Volume**

Volume Group được chia nhỏ thành nhiều Logical Volume, mỗi Logical Volume có ý nghĩa tương tự như partition. Nó được dùng cho các mount point và được format với những định dạng khác nhau như ext2, ext3, ext4,...

[](https://camo.githubusercontent.com/c89e722962cf4b73c9ec9ab33f2e46fd0978b6ba/687474703a2f2f692e696d6775722e636f6d2f5362515a4b6e752e706e67)

Khi dung lượng của Logical Volume được sử dụng hết ta có thể đưa thêm ổ đĩa mới bổ sung cho Volume Group và do đó tăng được dung lượng của Logical Volume

Ví dụ bạn có 4 ổ đĩa mỗi ổ 5GB khi bạn kết hợp nó lại thành 1 volume group 20GB, và bạn có thể tạo ra 2 logical volumes mỗi disk 10GB

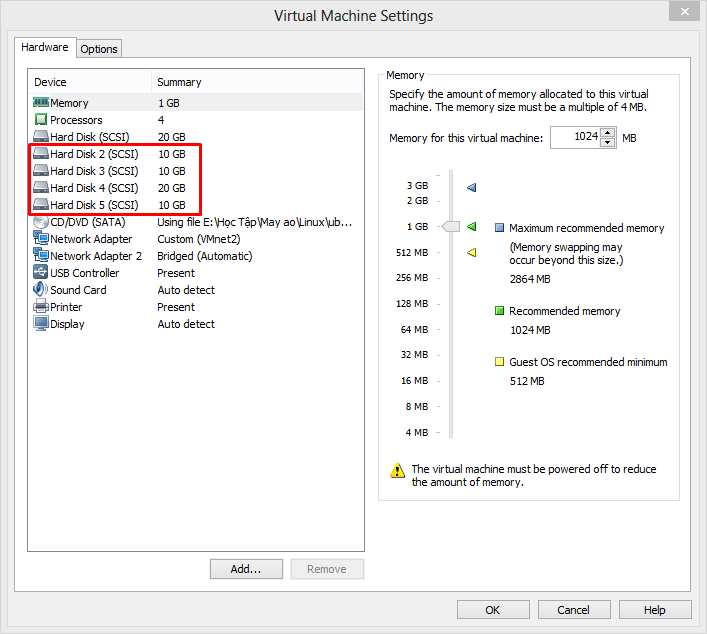
**File Systems**

* Tổ chức và kiểm soát các tập tin
* Được lưu trữ trên ổ đĩa cho phép truy cập nhanh chóng và an toàn
* Sắp xếp dữ liệu trên đĩa cứng máy tính
* Quản lý vị trí vật lý của mọi thành phần dữ liệu

**2. Hướng dẫn sử dụng LVM**

**2.1 Chuẩn bị**

* Tạo máy ảo trên vmware Workstation cài hệ điều hành ubuntu server 12.04
* Add thêm một số ổ cứng vào máy ảo

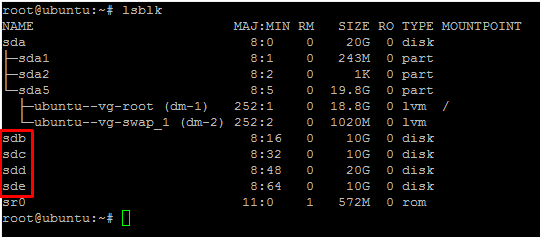
[](https://camo.githubusercontent.com/1485440c045b19c55c86177f4750a221d0742a1e/687474703a2f2f692e696d6775722e636f6d2f356764416e52722e706e67)

**2.2 Tạo Logical Volume trên LVM**

**B1. Kiểm tra các Hard Drives có trên hệ thống**

Bạn có thể kiểm tra xem có những Hard Drives nào trên hệ thống bằng cách sử dụng câu lệnh lsblk

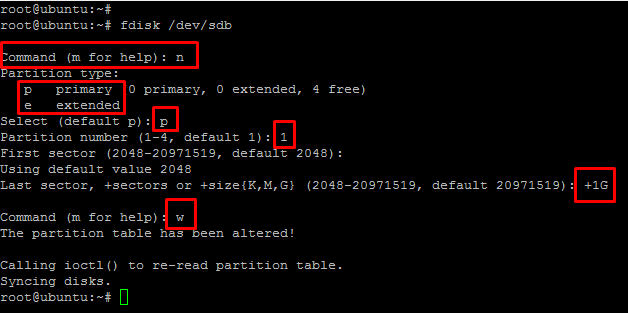
# lsblk

[](https://camo.githubusercontent.com/96644d5cf8a6c3bc028c01e652bfa3004e63f70f/687474703a2f2f692e696d6775722e636f6d2f364c5a634f76562e706e67)

Trong đó sdb, sdc, sdd, sde là các Hard Drives mà mình mới thêm vào

**B2. Tạo Partition**

Từ các Hard Drives trên hệ thống, bạn tạo các partition. Ở đây, từ sdb, mình tạo các partition bằng cách sử dụng lệnh sau fdisk /dev/sdb

[](https://camo.githubusercontent.com/d2588d6aa1de699c29ffda323d8885e02704d45f/687474703a2f2f692e696d6775722e636f6d2f657a4a6c6d67652e706e67)

 Trong đó bạn chọn `n` để bắt đầu tạo partition

 Bạn chọn `p` để tạo partition primary

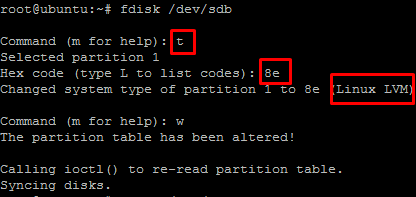
 Bạn chọn `1` để tạo partition primary 1

 Tại `First sector (2048-20971519, default 2048)` bạn để mặc định

 Tại `Last sector, +sectors or +size{K,M,G} (2048-20971519, default 20971519)` bạn chọn `+1G` để partition bạn tạo ra có dung lượng 1 G

 Bạn chọn `w` để lưu lại và thoát.

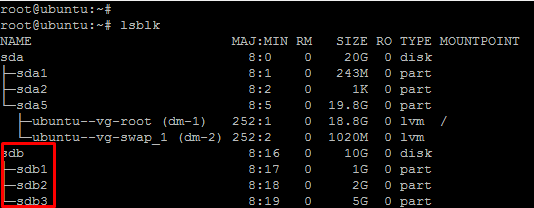
Tiếp theo bạn thay đổi định dạng của partition vừa mới tạo thành LVM

[](https://camo.githubusercontent.com/344f9bd1b93e4ca8690c699da6ad6061c0ac38d5/687474703a2f2f692e696d6775722e636f6d2f496839726950552e706e67)

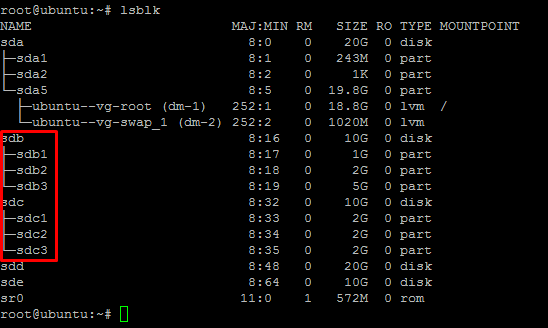
 Bạn chọn `t` để thay đổi định dạng partition

 Bạn chọn `8e` để đổi thành LVM

Tương tự, bạn tạo thêm các partition primary từ sdb

[](https://camo.githubusercontent.com/7b92a6b0b3ee41d1c10b4eda05a9c2af5bbb16eb/687474703a2f2f692e696d6775722e636f6d2f617746714b67742e706e67)

Tạo các partition primary từ sdc bằng lệnh fdisk /dev/sdc

[](https://camo.githubusercontent.com/b9a14fd0100306a37de15b1a31a31743f1810a14/687474703a2f2f692e696d6775722e636f6d2f4f3453454f42452e706e67)

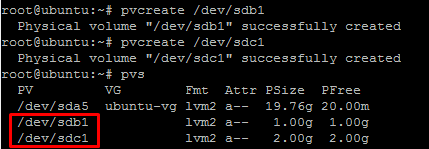
**B3. Tạo Physical Volume**

Tạo các Physical Volume là /dev/sdb1 và /dev/sdc1 bằng các lệnh sau:

# pvcreate /dev/sdb1

# pvcreate /dev/sdc1

Bạn có thể kiểm tra các Physical Volume bằng câu lệnh pvs hoặc có thể sử dụng lệnh pvdisplay

[](https://camo.githubusercontent.com/de854c4e33e89258373523d1f2313bc12fa1466e/687474703a2f2f692e696d6775722e636f6d2f42784d3241736e2e706e67)

**B4. Tạo Volume Group**

Tiếp theo, mình sẽ nhóm các Physical Volume thành 1 Volume Group bằng cách sử dụng câu lệnh sau:

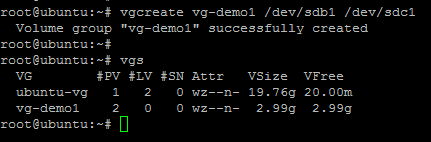
# vgcreate vg-demo1 /dev/sdb1 /dev/sdc1

Trong đó vg-demo1 là tên của Volume Group

Có thể sử dụng câu lệnh sau để kiểm tra lại các Volume Group đã tạo

# vgs

# vgdisplay

[](https://camo.githubusercontent.com/fa36acc5e43c7520839f811dddc37a6abcb3be2a/687474703a2f2f692e696d6775722e636f6d2f5559576c64484f2e706e67)

**B5. Tạo Logical Volume**

Từ một Volume Group, chúng ta có thể tạo ra các Logical Volume bằng cách sử dụng lệnh sau:

# lvcreate -L 1G -n lv-demo1 vg-demo1

-L: Chỉ ra dung lượng của logical volume

-n: Chỉ ra tên của logical volume

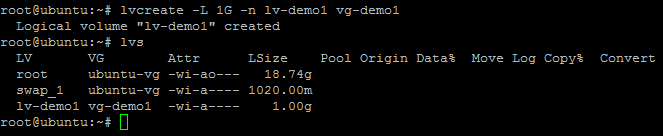
Trong đó lv-demo1 là tên Logical Volume, vg-demo1 là Volume Group mà mình vừa tạo ở bước trước

*Lưu ý là chúng ta có thể tạo nhiều Logical Volume từ 1 Volume Group*

Có thể sử dụng câu lệnh sau để kiểm tra lại các Logical Volume đã tạo

# lvs

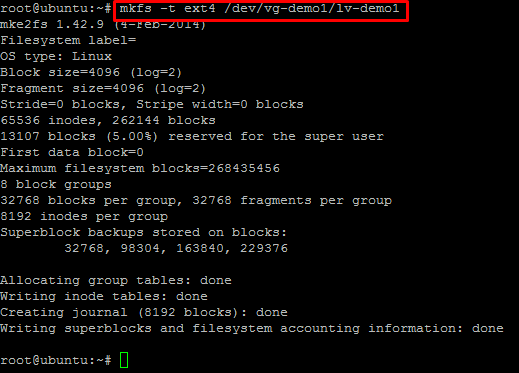
# lvdisplay

[](https://camo.githubusercontent.com/0b7c8b3e929d58b163b8058554420d31aa5c36be/687474703a2f2f692e696d6775722e636f6d2f417045656557722e706e67)

**B6. Định dạng Logical Volume**

Để format các Logical Volume thành các định dạng như ext2, ext3, ext4, ta có thể làm như sau:

# mkfs -t ext4 /dev/vg-demo1/lv-demo1

[](https://camo.githubusercontent.com/4c0afe9588cdf09879ebaa1c3d8f2b41fef359eb/687474703a2f2f692e696d6775722e636f6d2f4f7563355434502e706e67)

**B7. Mount và sử dụng**

Trong bài lab này, mình sẽ tạo ra một thư mục để mount Logical Volume đã tạo vào thư mục đó

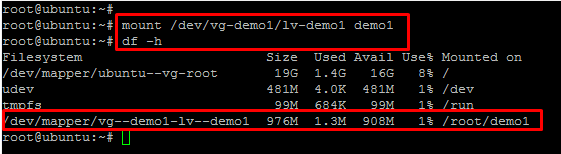
# mkdir demo1

Tiến hành mount logical volume lv-demo1 vào thư mục demo1 như sau:

# mount /dev/vg-demo1/lv-demo1 demo1

Kiểm tra lại dung lượng của thư mục đã được mount:

# df -h

[](https://camo.githubusercontent.com/5da2826eba29a41b5dae735aa7580c4776deb463/687474703a2f2f692e696d6775722e636f6d2f72647a467931432e706e67)

**2.3 Thay đổi dung lượng Logical Volume trên LVM**

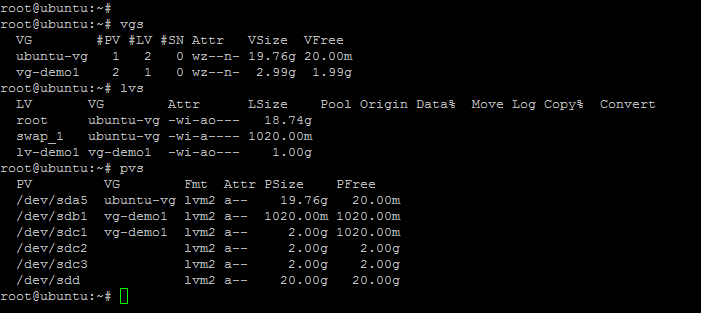
Ở phần trước, mình đã tiến hành tạo Logical Volume trong LVM. Ở phần này, chúng ta sẽ tìm hiểu làm thế nào để có thể thay đổi dung lượng của 1 Logical Volume đã được tạo ở phần trước.

Trước khi thay đổi dung lượng, các bạn cần phải kiểm tra các thông tin hiện có:

# vgs

# lvs

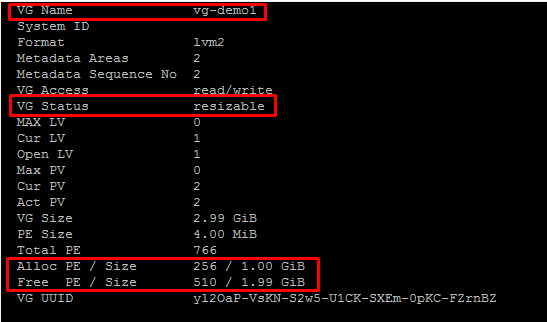
# pvs

[](https://camo.githubusercontent.com/2df8e97a0827d8a5beb0e188e7e3a64265a3ed8b/687474703a2f2f692e696d6775722e636f6d2f76765464694a6b2e706e67)

Ở đây, mình đã tạo được Logical Volume là lv-demo1, và giả sử Logical Volume này dung lượng đã đầy và chúng ta cần tăng kích thước của nó.

Logical Volume này thuộc Volume Group vg-demo1, để tăng kích thước, bước đầu tiên phải kiểm tra xem Volume Group còn dư dung lượng để kéo giãn Logical Volume không. Logical Volume thuộc 1 Volume Group nhất định, Volume Group đã cấp phát hết thì Logical Volume cũng không thể tăng dung lượng được. Để kiểm tra, ta dùng lệnh sau:

# vgdisplay

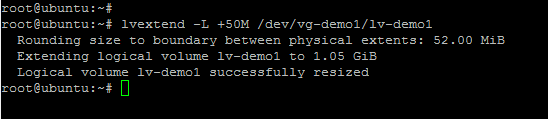
[](https://camo.githubusercontent.com/51d95d41faac2b90c42e26a809ac8cf43aad7403/687474703a2f2f692e696d6775722e636f6d2f623236504169562e706e67)

Volume Group ở đây vẫn còn dung lượng để cấp phát, ta có thể nhận thấy điều này qua 2 trường thông tin là VG Status resizable và Free PE / Size 510 / 1.99 GiB với dung lượng Free là 510\*4 = 2040 Mb

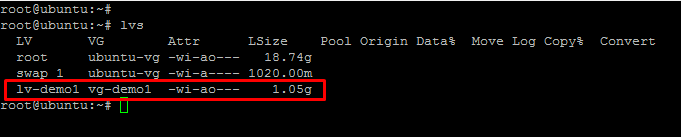
Để tăng kích thước Logical Volume ta sử dụng câu lệnh sau:

# lvextend -L +50M /dev/vg-demo1/lv-demo1

Với -L là tùy chọn để tăng kích thước

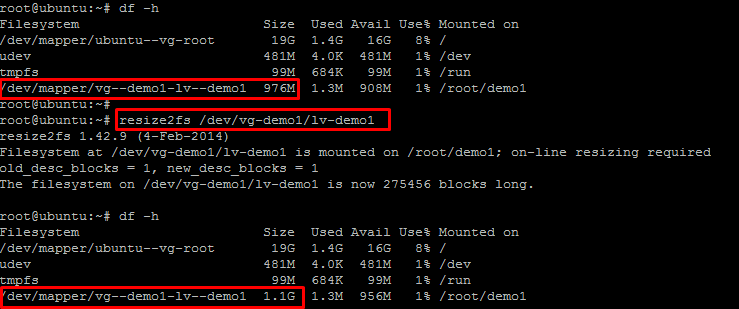
[](https://camo.githubusercontent.com/b6c77b8ec45db132673a80e613edd148126fd75b/687474703a2f2f692e696d6775722e636f6d2f5a614b6631376a2e706e67)

Kiểm tra lại bằng cách dùng lệnh # lvs

[](https://camo.githubusercontent.com/0e0d7e11c2c9e902c0fe69f6487dd828d15e429e/687474703a2f2f692e696d6775722e636f6d2f614c546371696e2e706e67)

Sau khi tăng kích thước cho Logical Volume thì Logical Volume đã được tăng nhưng file system trên volume này vẫn chưa thay đổi, bạn phải sử dụng lệnh sau để thay đổi:

# resize2fs /dev/vg-demo1/lv-demo1

[](https://camo.githubusercontent.com/39674d6771574d452648f5e981f81fbee319a105/687474703a2f2f692e696d6775722e636f6d2f687061547677642e706e67)

Để giảm kích thước của Logical Volume, trước hết các bạn phải umount Logical Volume mà mình muốn giảm

# umount /dev/vg-demo1/lv-demo1

Tiến hành giảm kích thước của Logical Volume

# lvreduce -L 20M /dev/vg-demo1/lv-demo1

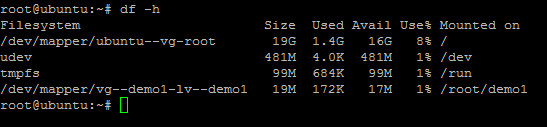
Sau đó tiến hành format lại Logical Volume

# mkfs.ext4 /dev/vg-demo1/lv-demo1

Cuối cùng là mount lại Logical Volume

# mount /dev/vg-demo1/lv-demo1 demo1

Kiểm tra kết quả ta được như sau:

[](https://camo.githubusercontent.com/5e54bdaa73c4c881b719252f75e7e04d624ed92c/687474703a2f2f692e696d6775722e636f6d2f313936396e76422e706e67)

**2.4 Thay đổi dung lượng Volume Group trên LVM**

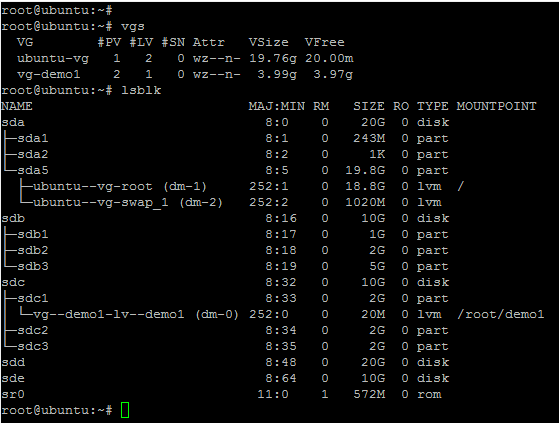
Ở phần trước mình có thể tăng kích thước của Logical Volume nhưng với điều kiện Volume Group của Logical Volume đó còn dung lượng. Phần này chúng ta sẽ tìm hiểu xem làm thế nào có thể mở rộng thêm kích thước của Volume Group cũng như thu hồi dung lượng của nó.

Việc thay đổi kích thước của Volume Group chính là việc nhóm thêm Physical Volume hay thu hồi Physical Volume ra khỏi Volume Group

Trước tiên, các bạn cần kiểm tra lại các partition và Volume Group

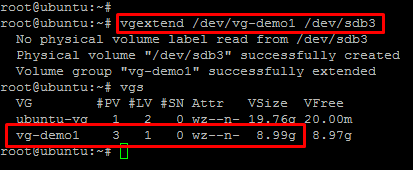
# vgs

# lsblk

[](https://camo.githubusercontent.com/4aee575ae25ebb010a75a885d0840815e08ceb32/687474703a2f2f692e696d6775722e636f6d2f366637307276502e706e67)

Tiếp theo, nhóm thêm 1 partition vào Volume Group như sau:

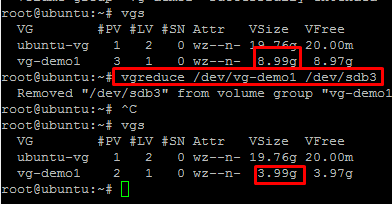
# vgextend /dev/vg-demo1 /dev/sdb3

[](https://camo.githubusercontent.com/5b27e0e0b351be2ea578cf7fa009402c5191ced0/687474703a2f2f692e696d6775722e636f6d2f6f38304e5a48782e706e67)

Ở đây, muốn nhóm vào Volume Group phải là Physical Volume nên hệ thống đã tự động tạo cho mình Physical Volume và nhóm vào Volume Group.

Chúng ta có thể cắt 1 Physical Volume ra khỏi Volume Group như sau:

# vgreduce /dev/vg-demo1 /dev/sdb3

[](https://camo.githubusercontent.com/18857f2af968273ac8c6cab6244f3a1e73047809/687474703a2f2f692e696d6775722e636f6d2f35436f556f54792e706e67)

**2.5 Xóa Logical Volume, Volume Group, Physical Volume**

**Xóa Logical Volumes**

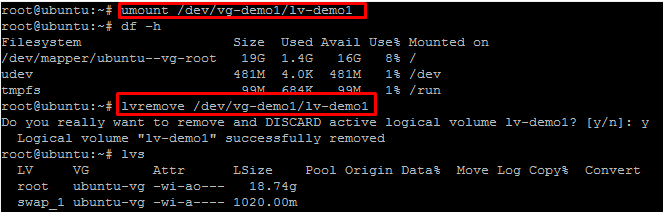
Trước tiên ta phải Umount Logical Volume

# umount /dev/vg-demo1/lv-demo1

Sau đó tiến hành xóa Logical Volume bằng câu lệnh sau:

# lvremove /dev/vg-demo1/lv-demo1

Ta kiểm tra lại kết quả

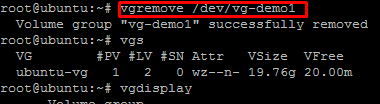
[](https://camo.githubusercontent.com/76d86230827fc0fc5948f67a42c70c054c52b286/687474703a2f2f692e696d6775722e636f6d2f624b79684e61462e706e67)

**Xóa Volume Group**

Trước khi xóa Volume Group, chúng ta phải xóa Logical Volume

Xóa Volume Group bằng cách sử dụng lệnh sau:

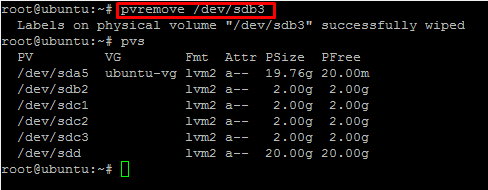
# vgremove /dev/vg-demo1

[](https://camo.githubusercontent.com/6168129f99067c287c7c91b760208bfe353c2ea5/687474703a2f2f692e696d6775722e636f6d2f6f33446330304c2e706e67)

**Xóa Physical Volume**

Cuối cùng là xóa Physical Volume:

# pvremove /dev/sdb3

[](https://camo.githubusercontent.com/2d0e565b90352ea8b5eea0c112b7ff5ffa9421d9/687474703a2f2f692e696d6775722e636f6d2f344b72724f39752e706e67)

Vậy là mình đã hoàn thành một bài lab đơn giản về LVM.

**3. Tổng kết**

Bài viết trên mình đã tổng hợp lại nhũng kiến thức cơ bản trong quá trình mình tìm hiểu về LVM, hy vọng nó có ích cho các bạn.