

Enterprise Linux 實戰講座

Logical Volume Manager (二)

前言：

LVM (一) 介紹 LVM 的運作原理並利用 Disk Druid 來實作 LVM。但 Disk Druid 此工具只有在安裝時才可使用，如果我們在系統安裝完畢後，需要實作 LVM 又該何解呢?此篇文章將介紹利用指令方式來實作 LVM 機制，其中會告訴讀者如何放大 LV (虛擬 Partition) 及加大 VG (虛擬硬碟) 的大小，還有快速備份 LV 的方法 (LV Snapshot) 及如何刪除 LVM 中的 VG。

LVM 相關指令

筆者列出 PV、VG、LV 比較表 (如表 1)，我們也可用下列簡圖 (圖 1) 來代表彼此之間的關係。接下來我們要介紹 PV、VG、LV 相關指令。

表 1：PV、VG、LV 意義

名詞	意義	作業系統相對應裝置
PV	實體分割區(Partition)	/dev/hda#
VG	虛擬硬碟(磁區群組)	/dev/<VG name>/ 目錄
LV	虛擬分割區(邏輯群組)	/dev/<VG name>/<LV name>

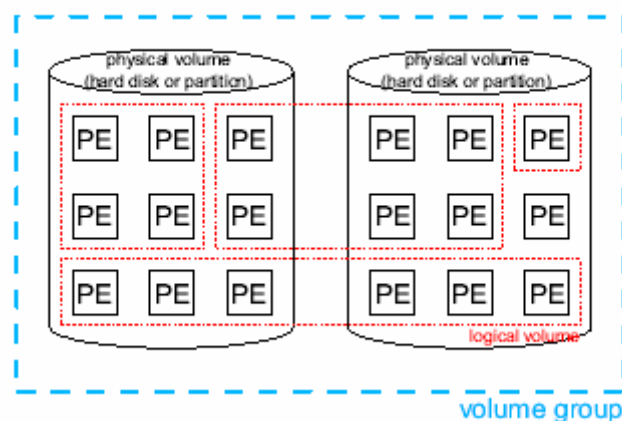


圖 1：PV、VG、LV 關係圖

- PV 指令

pvscan 掃描系統中所有 PV 的資訊

```
[root@lvm root]# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/hda2" of VG "rootvg" [2.92 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/hda3" of VG "rootvg" [2.92 GB / 0 free]
pvscan -- ACTIVE   PV "/dev/hda5" of VG "rootvg" [2.92 GB / 1.83 GB free]
pvscan -- total: 3 [8.78 GB] / in use: 3 [8.78 GB] / in no VG: 0 [0]
```

pvcreate <pv>

建立 Physical Volume

在 Linux LVM 的機制 PV 是由實體的 Partition 轉換而成，這個 Partition 必須為 Partition id 為 **8e** 的 Partition。當我們建立 PV 時，LVM 機制保留此 Partition 最前面的區塊用來記錄有關 LVM 的屬性（例如將來加入 VG 後，PE 的大小），此塊空間我們稱為 VGDA (**V**olume **G**roup **D**escriptor **A**rea)。

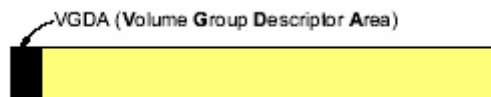


圖 2: VGDA 示意圖

pvmove [-n <lv>] <source pv> [<destination pv>]

搬移 Physical Volume 內資料

pvmove 這個指令並不是搬移 PV，而是搬移 PV 內的資料，其實就是將 PV 中的 PE 資料搬移至同一個 VG 中的另一個 PV 內（註：一定要在同一個 VG 內的 PV），這個指令通常是用於我們要將某個 PV 從 VG 中移除（註：利用 vgreduce 指令）時，先將 PV 中的資料搬移到另一個 PV 中，再進行移除 VG 的動作。

pvdisplay <pv>

顯示 Physical Volume 的屬性

pvdisplay 這個指令就簡單多了，目的是顯示 PV 的資訊，執行訊息如下：

```
# pvdisplay /dev/hda10
pvdisplay -- "/dev/hda10" is a new physical volume of 54.88 MB
```

- **VG 指令**

vgscan 掃描系統中所有 VG 的資訊

```
[root@lvm root]# vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- found active volume group "rootvg"
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume
group
```

vgcreate [-s <pe size>] <vg name> <pv> [<pv>...]

建立 Volume Group

我們知道 VG 是由一個或一個以上的 PV 所組成的，當 PV 加入 VG 後會被切成一塊一塊的 PE（預設為 4MB），所以語法當然就沒有問題。筆者在從事 Linux 教學工作時，常強調先懂原理，再看指令，才不會覺得指令的語法艱澀難懂。

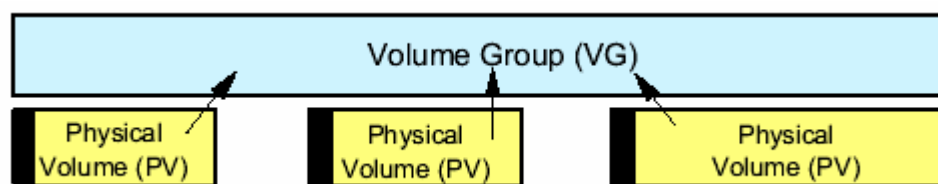


圖 3：VG 示意圖

```
# vgcreate rootvg /dev/hda10 →用 /dev/hda10 這個 PV 組成 rootvg
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "rootvg"
vgcreate -- volume group "rootvg" successfully created and activated
```

由上面的 vgcreate 的執行訊息「using **default physical extent size 4 MB**」，便可清楚看到當 PV 加入 VG 後會被切成一塊一塊的 PE（預設為 4MB）。rootvg 後也可接一個以上的 PV，代表一次用多個 PV 組成 rootvg。

vgdisplay [<vg>]

顯示 Volume Group 資訊

此指令會顯示 VG 的大小，其中 PE 的大小，還有此 VG 共有幾塊 PE...等資訊。

vgremove [<vg>]

刪除 Volume Group

刪除 VG 之前，必須讓此 VG 停止作用 (deactive)，目的就是讓所有程式不能再存取此 VG 內的資料。讓 VG 停止作用的指令為

vgchange -a n <vg>。

- **LV 指令**

lvscan 掃描系統中所有 LV 的資訊

```
[root@lvm root]# lvscan
```

```
lvscan -- ACTIVE          "/dev/rootvg/rootlv" [5.86 GB]
```

```
lvscan -- ACTIVE          "/dev/rootvg/homelv" [600 MB]
```

```
lvscan -- ACTIVE          "/dev/rootvg/swaplv" [500 MB]
```

```
lvscan -- 3 logical volumes with 6.93 GB total in 1 volume group
```

```
lvscan -- 3 active logical volumes
```

lvcreate -L <size> [-n <lv name>] <vg> [<pv>...]

建立 Logical Volume

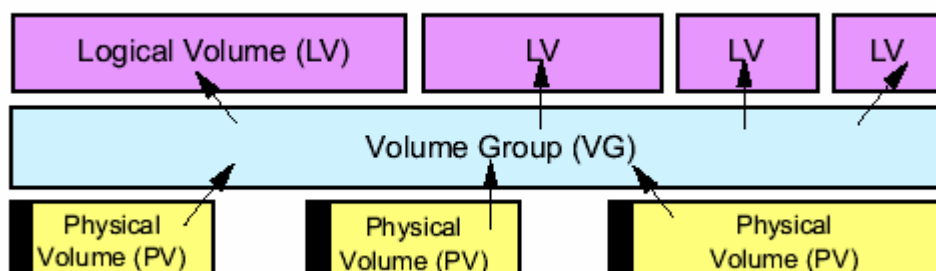


圖 4：LV 示意圖

LV 是 VG 中的一塊空間，就像原來 Linux 中的硬碟中的 partition，所以要建立 LV 就像在硬碟中建立 Partition 一樣，必須指定要從那個硬碟 (VG) 切割，其大小為多少。常見的指令用法如下：

```
#lvcreate -L 512M -n lv01 rootvg
```

從 rootvg 中切出一個 512MB 的空間的 LV 並指定其名稱爲 lv01

```
#lvcreate -l 32 -n lv01 rootvg
```

從 rootvg 中挪出 32 個 PE 的空間來建立 LV, 並指定其名稱爲 lv01

```
lvdisplay <lv> [<lv>...]
```

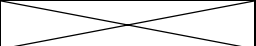
顯示 Logical Volume 資訊

```
lvremove <lv> [<lv>...]
```

刪除 Logical Volume

筆者將上述介紹 LVM 相關的指令整理如下表：

表 2：LVM 相關指令

	Physical Volume	Volume Group	Logical Volume
scan (掃描整個系統)	pvscan	vgscan	lvscan
create (建立)	pvcreate	vgcreate	lvcreate
display (顯示)	pvdisplay	vgdisplay	lvdisplay
remove (刪除)		vgremove	lvremove

實戰演練

演練一：建立 LVM 的檔案系統 (/data 檔案系統 大小 30MB)

演練二：動態放大檔案系統 (將 /data 由 30 MB → 40MB)

演練三：動態加大虛擬 HD (Volume Group) 大小

演練四：Logical Volume Snapshots (快速備份)

演練五：刪除 VG

演練一：建立 LVM 的檔案系統 (/data 檔案系統 大小 30MB)

在實作之前，請各位回想一下傳統 Linux 新增檔案系統的作法。假如現在我要新增一個 /data 檔案系統，大小爲 30MB，流程應爲 create new partition (fdisk) → create filesystem (mke2fs) → mount to /data directory (圖 5)。

在原有 Linux 檔案系統的架構中，一個檔案系統 (假設是 /data 檔案系

統)一定對應某個硬碟的 Partition (/dev/hda10)。/dev/hda#是實體硬碟從第 X 個磁柱到 Y 個磁柱，所以/data 檔案系統實際上是對應到硬碟的某個真實的空間。

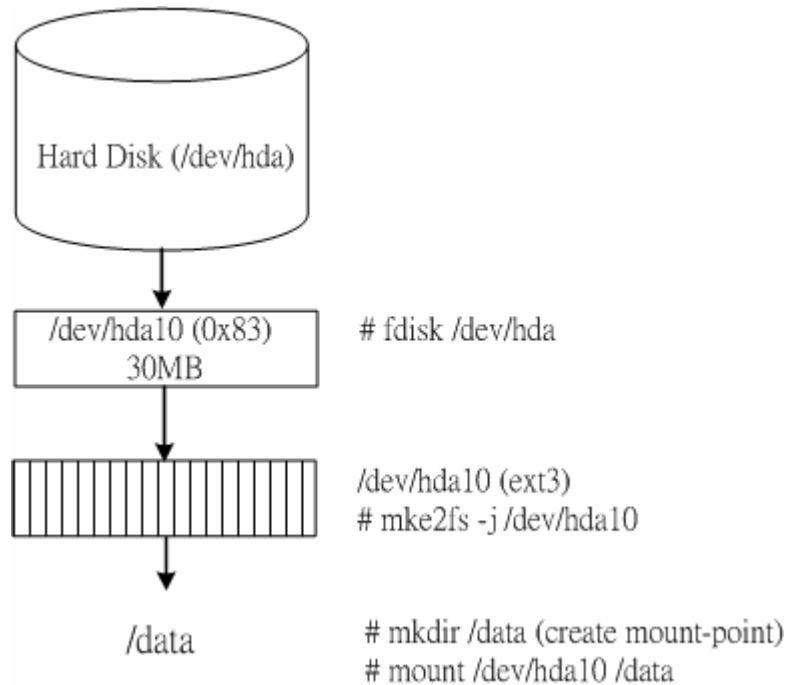


圖 5. 傳統 Linux 建立 /data 檔案系統流程圖

步驟及原理：

假設同一個需求，我們利用 LVM 的需求來實作。RedHat 9.0 kernel 預設已支援 LVM 機制。現在我們要利用 LVM 的機制建立 30 MB /data 檔案系統，筆者時常建議實作 LVM 之前，先想一想整個原理及流程，除了可以加深記憶也比較不會流於只是在敲指令。首先我們得先湊出一個 **VG (虛擬硬碟)**，而 VG 是由 PV 所組成，而 PV 是由實體的 Partition 所轉換而成的。所以我們第一件事應該是建立 LVM 專用的 **Partition**，然後將其轉成 **PV**，再利用它組成 **VG**。(圖 6)

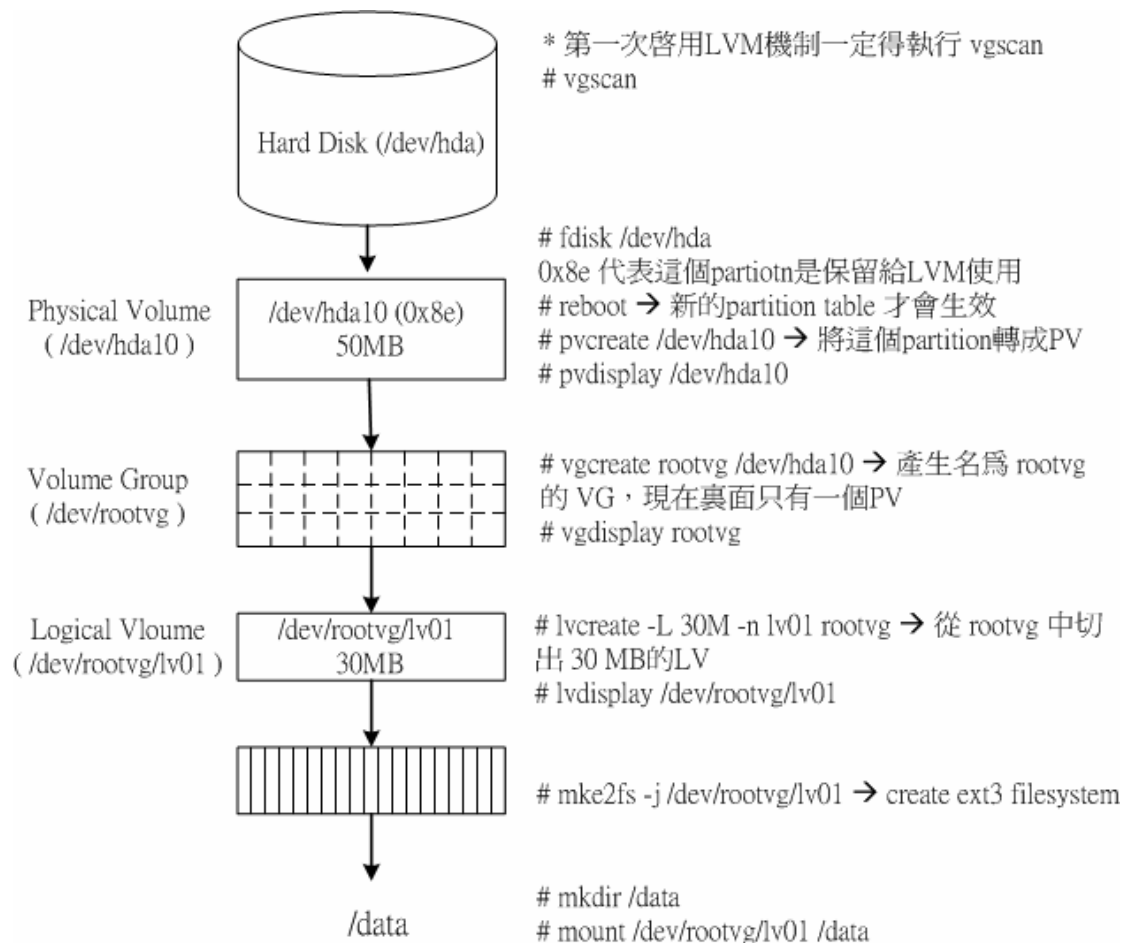


圖 6. Linux LVM create /data filesystem 流程圖

Step1. 第一次啓用 LVM 機制一定得執行 vgscan

```
# vgscan 第一次啓用 LVM 機制一定得執行 vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of
your volume group
```

Step2. 新增 LVM partition

```
# fdisk /dev/hda
```

The number of cylinders for this disk is set to 8924.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs
(e.g., DOS FDISK, OS/2 FDISK)

```
Command (m for help): n
First cylinder (6698-8924, default 6698):
Using default value 6698
Last cylinder or +size or +sizeM or +sizeK (6698-8924, default 8924):
+50M
```

Command (m for help): p

```
Disk /dev/hda: 255 heads, 63 sectors, 8924 cylinders
Units = cylinders of 16065 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	26	208813+	83	Linux
/dev/hda2		27	1301	10241437+	83	Linux
/dev/hda3		1302	2576	10241437+	83	Linux
/dev/hda4		2577	8924	50990310	f	Win95 Ext'd (LBA)
/dev/hda5		2577	2831	2048256	82	Linux swap
.....						
/dev/hda10		6698	6704	56196	83	Linux

Command (m for help): t

```
Command (m for help): t
Partition number (1-10): 10
```

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

```
Changed system type of partition 10 to 8e (Linux LVM)
```

Command (m for help): p

```
.....
/dev/hda10      6698      6704      56196      8e  Linux LVM
```

Step3.將/dev/hda10 轉成 PV

```
# pvcreate /dev/hda10 → 將這個 /dev/hda10 轉成 PV
```

```
pvcreate -- physical volume "/dev/hda10" successfully created
```

```
# pvdisplay /dev/hda10
```

```
pvdisplay -- "/dev/hda10" is a new physical volume of 54.88 MB
```


Step4. 組成虛擬硬碟 rootvg

```
# vgcreate rootvg /dev/hda10 →用 /dev/hda10 這個 PV 組成
rootvg
           rootvg 相當以前 Linux 的一顆硬碟 /dev/hda
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 255.99 Gigabyte
vgcreate -- doing automatic backup of volume group "rootvg"
vgcreate -- volume group "rootvg" successfully created and activated
```

```
# vdisplay rootvg
VG Name          rootvg
VG Access        read/write
VG Status        available/resizable
VG #             0
MAX LV           256
Cur LV          1
Open LV          1
MAX LV Size      255.99 GB
Max PV           256
Cur PV          1
Act PV           1
VG Size         48 MB →這個虛擬硬碟 rootvg 的大小為 48MB
PE Size         4 MB
Total PE         12
Alloc PE / Size  11 / 44 MB
Free PE / Size   1 / 4 MB
VG UUID          kPdjjl-b0fK-1w0D-WagY-Bg6l-VsZk-ReVpPW
```

Step5. 從虛擬硬碟 rootvg 中切出一塊 LV，名稱 lv01、大小為 30MB。

```
# lvcreate -L 30M -n lv01 rootvg
lvcreate -- rounding size up to physical extent boundary
lvcreate -- doing automatic backup of "rootvg"
lvcreate -- logical volume "/dev/rootvg/lv01" successfully created
```

```
# lvdisplay /dev/rootvg/lv01
--- Logical volume ---
LV Name          /dev/rootvg/lv01
```

VG Name	rootvg
LV Write Access	read/write
LV Status	available
LV #	1
# open	1
LV Size	32 MB
Current LE	8
Allocated LE	8
Allocation	next free
Read ahead sectors	1024
Block device	58:0

Step6.將 **lv01** 格式化，即是建立 **ext3** 檔案系統的動作。

```
# mke2fs -j /dev/rootvg/lv01
mke2fs 1.32 (09-Nov-2002)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
8192 inodes, 32768 blocks
1638 blocks (5.00%) reserved for the super user
First data block=1
4 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577

Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 33 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Step7.建立 **mount point** 並利用其存取 **lv01** 的資料(即傳統 **linux mount** 的觀念)。

```
# mkdir /data ← 建立 mount-point
# mount /dev/rootvg/lv01 /data
# df -h
Filesystem                Size  Used Avail Use% Mounted on
/dev/hda2                  9.6G  3.2G  5.9G  35% /
/dev/hda1                  197M   20M  168M  11% /boot
/dev/hda6                   29G   1.6G   25G   6% /Database
none                       440M    0  440M   0% /dev/shm
/dev/hda3                   9.6G  4.0G  5.1G  44% /usr
/dev/rootvg/lv01           31M  4.1M   25M  14% /data
```

演練二：動態放大檔案系統（將 /data 由 30 MB → 40MB）

這麼辛苦了建立 /data 檔案系統，所為何事呢？在原先 Linux 檔案系統的架構下，如果 /data 不敷使用，即使硬碟還有剩餘的空間，依舊是無法拿來使用，但在 LVM 的架構下可就不一樣了。由以上的流程圖我們知道 rootvg 這顆虛擬的硬碟中應該還有 20MB 的剩餘空間，在 LVM 下我們可以把這些空間再分配給 /data 檔案系統，感覺上很像動態放大 /data 檔案系統。

Step1. umount /data

```
# umount /data
```

Step2. 利用 e2fsadm 指令再新增 10MB 空間給 lv01

```
# e2fsadm -L +10M /dev/rootvg/lv01
e2fsadm -- rounding size up to physical extent boundary
e2fsck 1.27 (8-Mar-2002)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/rootvg/lv01: 11/8192 files (0.0% non-contiguous), 5166/32768
blocks
lvextend -- extending logical volume "/dev/rootvg/lv01" to 44
MB
lvextend -- doing automatic backup of volume group "rootvg"
lvextend -- logical volume "/dev/rootvg/lv01" successfully
```

```

extended
resize2fs 1.27 (8-Mar-2002)
Begin pass 1 (max = 2)
Extending the inode table
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 3 (max = 4)
Scanning inode table
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/rootvg/lv01 is now 45056 blocks long.

e2fsadm -- ext2fs in logical volume /dev/rootvg/lv01
successfully extended to 44 MB
放大 lv01 成功

```

Step3. mount /data

```

# mount /dev/root/lv01 /data
# df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda2	9.6G	3.2G	5.9G	35%	/
/dev/hda1	197M	20M	168M	11%	/boot
.....					
/dev/rootvg/lv01	42M	4.1M	36M	10%	/data

```

👁 /data 大小由 30MB 變成 40MB

```

演練三：動態加大虛擬 HD (Volume Group) 大小

問題又來，假設 rootvg 這顆虛擬硬碟已沒有剩餘的空間，怎麼辦？沒關係我們可以放大 rootvg 的空間，怎麼放大 rootvg 的空間呢？只要再新增一個 PV 並將其加入 rootvg，那麼 rootvg 空間就會加大，感覺上很像動態放大硬碟的空間。

Step1. 再新增一個 50MB partition 並轉成 PV

```

# fdisk /dev/hda → add new 0x8e partition /dev/hda11 (50MB)
# pvcreate /dev/hda11

```

Step2. 並把它加入 rootvg，撐大 rootvg 這顆虛擬硬碟的大小

```

# vgextend rootvg /dev/hda11
vgextend -- INFO: maximum logical volume size is 255.99 Gigabyte

```

```
vgextend -- doing automatic backup of volume group "rootvg"
vgextend -- volume group "rootvg" successfully extended ←成功
功地放大 rootvg
```

```
# vdisplay rootvg
VG Name          rootvg
VG Access        read/write
VG Status        available/resizable
VG #             0
MAX LV           256
Cur LV          1
Open LV          1
MAX LV Size      255.99 GB
Max PV           256
Cur PV          2
Act PV           2
VG Size          96 MB ←變成 96 MB
PE Size          4 MB
Total PE         24
Alloc PE / Size  11 / 44 MB
Free PE / Size   13 / 52 MB
VG UUID          kPdjjl-b0fK-1w0D-WagY-Bg6l-VsZk-ReVpPW
```

演練四：Logical Volume Snapshots（快速備份）

這是一個非常先進的機制，相同的功能在 IBM AIX 要最新版本 5.2 才有提供這個機制。它的用途是來快速備份 LV 內的資料，步驟如下：

Step1. 先將 /data filesystem 變成 Read-Only

```
#mount -o remount,ro /data
```

Step2. Snapshots/dev/rootvg/lv01

```
#lvcreate -s -L 30M -n lv01snap /dev/rootvg/lv01
```

```
#mount -o remount,rw /data ← Snapshot 快照後就可恢復可讀寫
```

Step3. 讀取 Snapshot LV 的資料

```
#mkdir /backup
```

```
#mount -o ro /dev/rootvg/snap01 /backup
```

此時讀者可以檢查/data 跟 /backup 目錄的內容會完全一模一樣。

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda2       9.6G  3.2G  5.9G  35% /
/dev/hda1       197M   20M  168M  11% /boot
.....
/dev/rootvg/lv01    42M  4.1M   36M  10% /data
/dev/rootvg/lv01back 42M  4.1M   36M  10% /databackup
```

```
# ls -l /data /backup
/backup:
total 16
-rw-r--r--  1 root  root           0 Sep  7  2003 file
drwx-----  2 root  root    12288 Sep  7  2003 lost+found
-rw-r--r--  1 root  root     4004 Sep  7  2003 passwd

/data:
total 16
-rw-r--r--  1 root  root           0 Sep  7  2003 file
drwx-----  2 root  root    12288 Sep  7  2003 lost+found
-rw-r--r--  1 root  root     4004 Sep  7  2003 passwd
```

演練五：刪除 VG

最後我們演練將之前所建立的 rootvg 刪除，刪除 VG 時，必須先將 VG 內的 LV 先移除，並將 VG 停止作用(deactivate),才能此 VG 刪除。

Step1. 刪除 LV

```
# umount /backup/
```

```
# lvremove /dev/rootvg/lv01back
```

```
lvremove -- do you really want to remove "/dev/rootvg/lv01back"? [y/n]: y
lvremove -- doing automatic backup of volume group "rootvg"
lvremove -- logical volume "/dev/rootvg/lv01back" successfully removed
```

如果讀者有實作 LV Snapshot，必須先移除 snapshot LV 才能移除原有的 LV。

```
#umount /data
```

```
#lvremove /dev/rootvg/lv01
```

```
lvremove /dev/rootvg/lv01
lvremove -- do you really want to remove "/dev/rootvg/lv01"? [y/n]: y
lvremove -- doing automatic backup of volume group "rootvg"
lvremove -- logical volume "/dev/rootvg/lv01" successfully removed
```

Step2. 將 VG 停止作用(deactivate)

```
# vgchange -a n rootvg
```

```
vgchange -- volume group "rootvg" successfully deactivated
```

Step3. 刪除 VG

```
# vgremove rootvg
```

```
vgremove -- volume group "rootvg" successfully removed
```

後記

Linux LVM機制就介紹至此，其實LVM是個大哉問。讀者如果有進一步的興趣，建議各位可以到IBM紅皮書網站 <http://www.redbooks.ibm.com> 查詢相關書籍。Enterprise Linux 3.0 上的LVM 為 1.03 版，嚴格說來跟IBM AIX上 LVM仍有一段差距，例如AIX 上Mirror VG的功能在 1.03 版就仍未實作完成。但令人期待的是kernel 2.6 會把Sistina 公司所開發LVM 2.0 包含在內，屆時Linux上的LVM 應可跟AIX 一較高下。