

# Enterprise Linux 實戰講座

## Logical Volume Manager (一)

### 前言：

從 RedHat Linux 8.0 開始便可以使用 Logical Volume Manager (LVM, 邏輯磁區管理員) 來做硬碟空間的分配。什麼是 LVM 呢? LVM (Logical Volume Manager) 原先運用於 IBM AIX Unix 系統, 是一種分配硬碟空間到邏輯磁區的方法, 這種方式比傳統 Partition 分配容易做容量的調整, 它讓眾多硬碟機結合、彷彿融為一體, 可動態放大檔案系統的機制, 是讓 Linux 邁向高階伺服器的一大利器。

### 概念：

使用 LVM, 硬碟中的 Partition 必須加入『Logical Volume Group 邏輯磁區群組 (簡稱 **VG**)』, 這種 Partition 在 LVM 中稱為『Physical Volume 實體磁區 (簡稱 **PV**)』, 各位可以把 Logical Volume Group 想成是一個大硬碟, 然後再從其中切出一塊空間『Logical Volume 邏輯磁區 (簡稱 **LV**)』對應至系統的檔案系統。

除了 /boot 分割區, 其餘的檔案系統皆可在 Volume Group 在這顆虛擬的大硬碟中。/boot 分割區不可以在 VG 中, 因為開機管理程式將無法讀取到它。假如您想要根目錄 / 分割區在 VG 中, 您必須另外建立一個 /boot 分割區, 使其不屬於 VG 的一員。(圖 1)

VG 可以分割為幾個『Logical Volume 邏輯磁區』, 將會指定它們的掛載點, 例如 /home 與 /, 以及檔案系統類型, 如 ext3。當某個檔案系統的容量全滿時, VG 中的剩餘空間可以增加到此檔案系統所對映的邏輯磁區以增大此檔案系統大小。

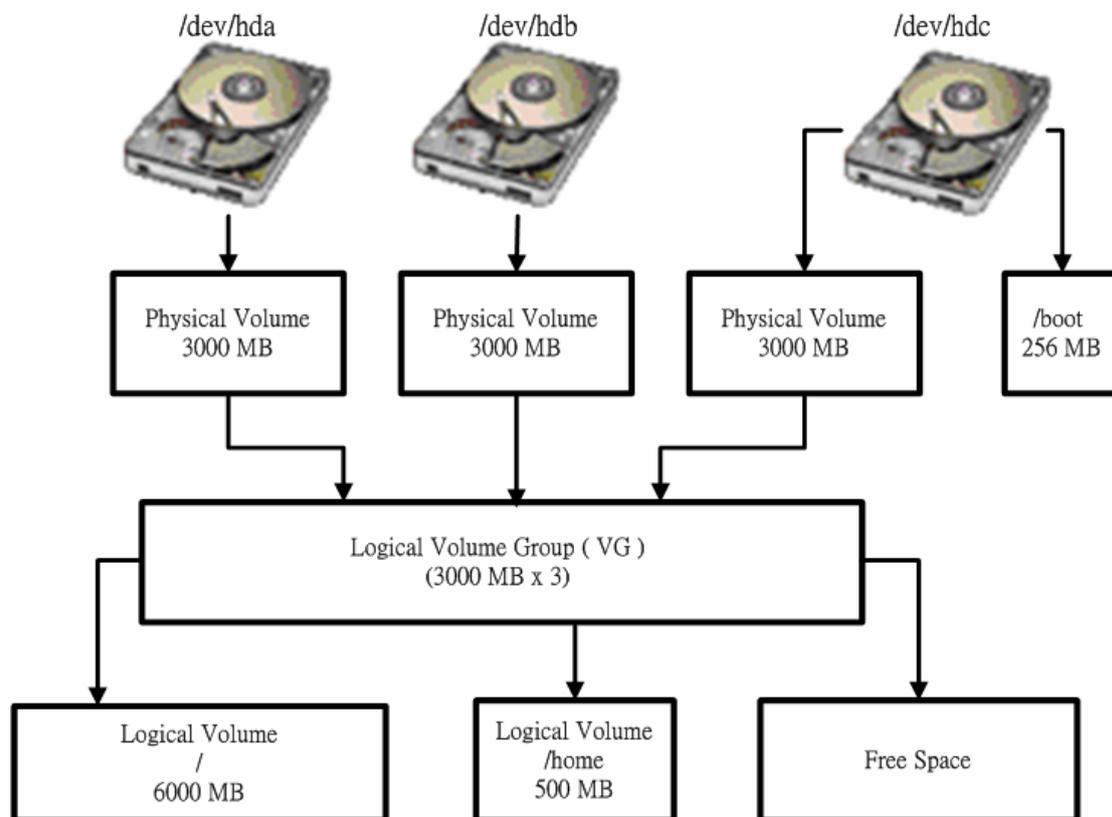


圖 1、Linux LVM 與檔案系統之間的關係圖

## LVM 運作機制:

接下來我們進一步介紹建立 LVM 的機制，首先，我們得先認識幾個名詞。LVM 機制原先為 IBM AIX 特有的硬碟管理機制，它的最主要的目的是為了克服 Unix 作業系統規劃給某個檔案系統 特定的空間後就無法再改變。例如：各位可能當初在安裝系統時，認為 /home 只要 500MB 就足夠，後來因為使用者愈來愈多，原先規劃的空間可能就不敷使用。

傳統的解法就是將原 /home 的資料備份出來，重新切割一塊新的 Partition 格式化成新的檔案系統，再將原有的資料還原回去。即使現有的硬碟還有剩餘的空間，你也無法馬上利用這些空間，還是得重新切個新的 Partition。但 LVM 利用在實體的硬碟和檔案系統 加一層 **Logical** 的對映機制來達到動態放大檔案系統的功能進而解決傳統 Unix 所面臨的問題。

## LVM 的重要名詞 ( VG、PV、LV ) :

我們先談 VG 及 PV 這兩個名詞。在原先 IBM AIX 的紅皮書中的定義中，VG 是 LVM 機制最重要的東西，就是一堆 PV 的組合，一個 VG 可能包含一個 PV 或數個 PV。這樣解釋，各位一定聽不懂，筆者當年接觸 AIX 時也是搞得一個頭兩個大，覺得有點像用另一個不懂的名詞（PV）來解釋不懂的名詞（VG）。

換個角度來看，其實各位可以把 VG 想成原先 Linux 實體觀念中的一顆硬碟。而在實體世界中硬碟是由一個或多個 Partition 組成，在 LVM 中 VG 是由一個或一個以上的 PV 組成。所以在 Linux LVM 機制中的 PV 其實是硬碟中的 Partition 所轉換變成的。

那什麼是 LV，可不是那個服飾名牌 LV，當然搞懂了 LV 也不見得會幫你/妳賺到 LV 的包包，不過倒是可以幫你的 Linux 檔案系統 加上一件華麗的衣裳。好了言歸正傳，LV 是 VG 中的一塊空間，就像原來 Linux 中的 partition /dev/hda#，是用來對應到（mount）系統中的某個目錄（mount-point）。

- **Physical Volume (PV)**: 在 AIX 原有的定義將實體的硬碟稱為 PV，在 Linux 上各位可想成原先的一個 partition 就是一個 PV。
- **Volume Group (VG)**: 就是一堆 PV 的組合，也就是說一個 VG 可能包含一個 PV 或數個 PV。
- **Physical Extent (PE)**: 當數個 PV 組合成一個 VG 時，LVM 會在所有的 PV 做類似格式化的工作，將每個 PV 切成一塊一塊的空間，這一塊一塊的空間就稱為 PE，通常是 4MB。
- **Logical Volume (LV)**: 由一群 LE 組成，在 LVM 的機制下一個檔案系統（假設是 /home filesystem）實際對應的不是 hda#，而是一個 LV。
- **Logical extent (LE)**: LV 的組成單位，它的大小為 PE 的倍數，通常為 1 : 1 的關係。

筆者將 LVM 中最重要的三個名詞及其意義整理簡列如下表：

表1 PV、VG、LV 比較表

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>

下圖是筆者用 Linux 的角度來看 LVM 所重畫的 Linux LVM 的架構圖。

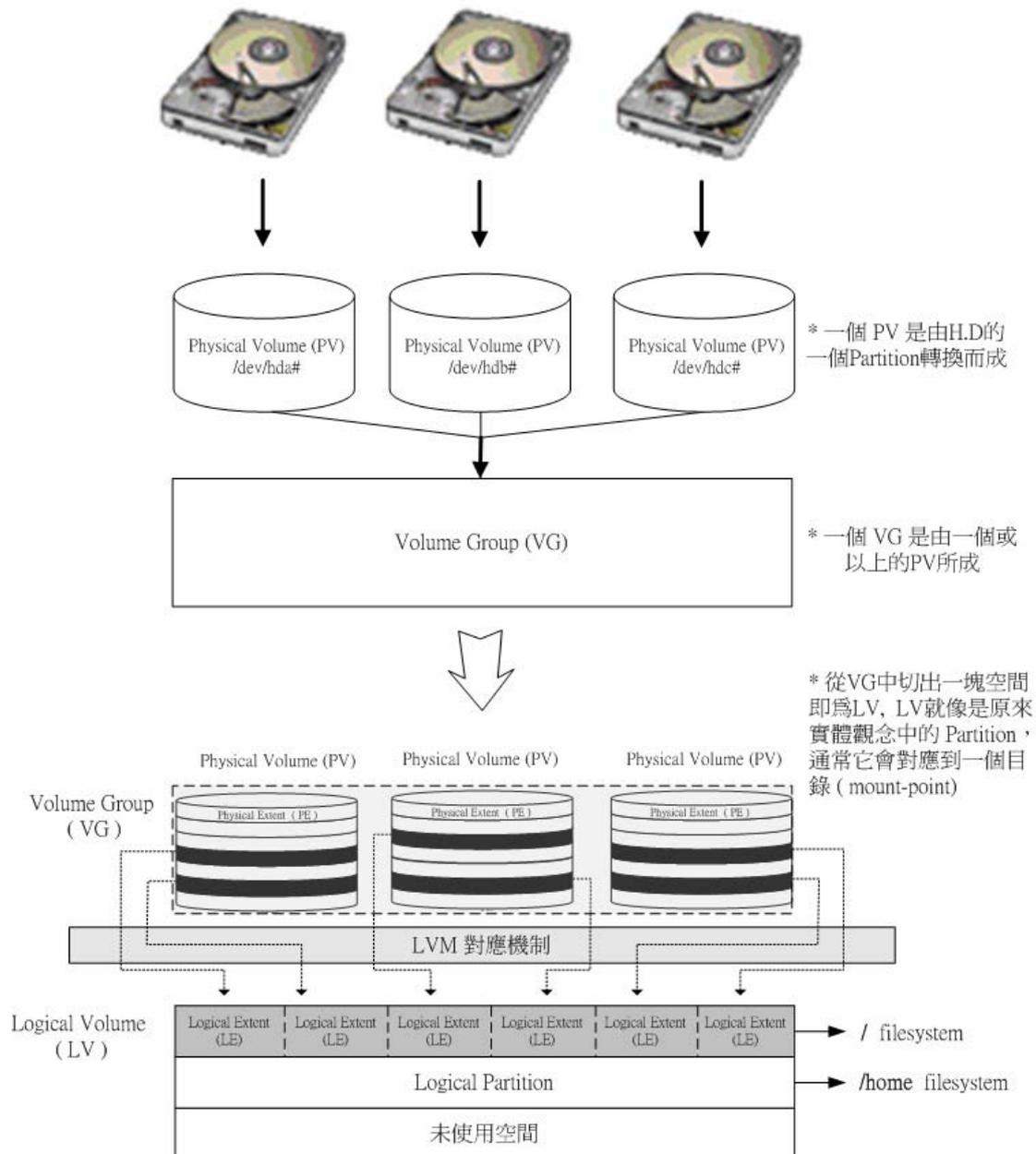


圖 2、Linux LVM 架構圖

### 實戰演練：

有兩種方式可以建立 LVM 的機制，一是藉由 Disk Druid (只有安裝時才提供)，另法是利用 LVM 指令來建立管理。下面為各位示範是利用 **Disk Druid** 來實作 LVM，最後示範如何動態放大 /home 檔案系統。

測試環境：

作業系統：RedHat Enterprise Linux AS 3.0 (亦適用於 RedHat 8.0、9.0、Fedora)

硬碟：10GB IDE 硬碟

1./boot 256 MB → Linux Partition (圖 3~6)

( /boot 分割區不可以在 VG 中，因為開機管理程式將無法讀取到它 )



圖 3、RedHat Enterprise Linux As 3.0 安裝畫面



圖 4、選擇繁體中文



圖 5、選用 Disk Druid 來分割磁碟 (Disk Druid 只有在安裝時才可使用)



圖 6、Disk Druid 設定畫面

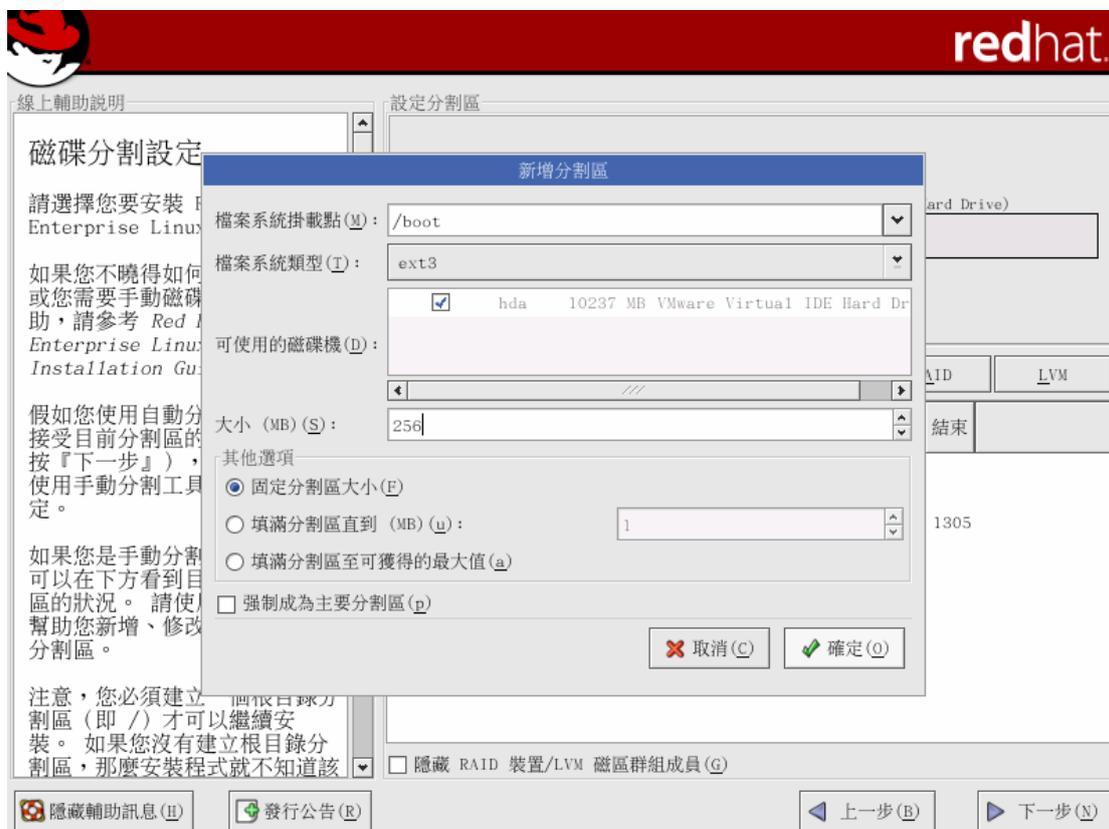


圖 7、新增 /boot 分割區

2. 然後切出三個大小 3000MB Partition，hda2、hda3、hda4，檔案系統類型為 LVM，此步驟相當於建立三個 Physical Volume（圖 7,8），最後這個硬碟還剩下 988 MB。

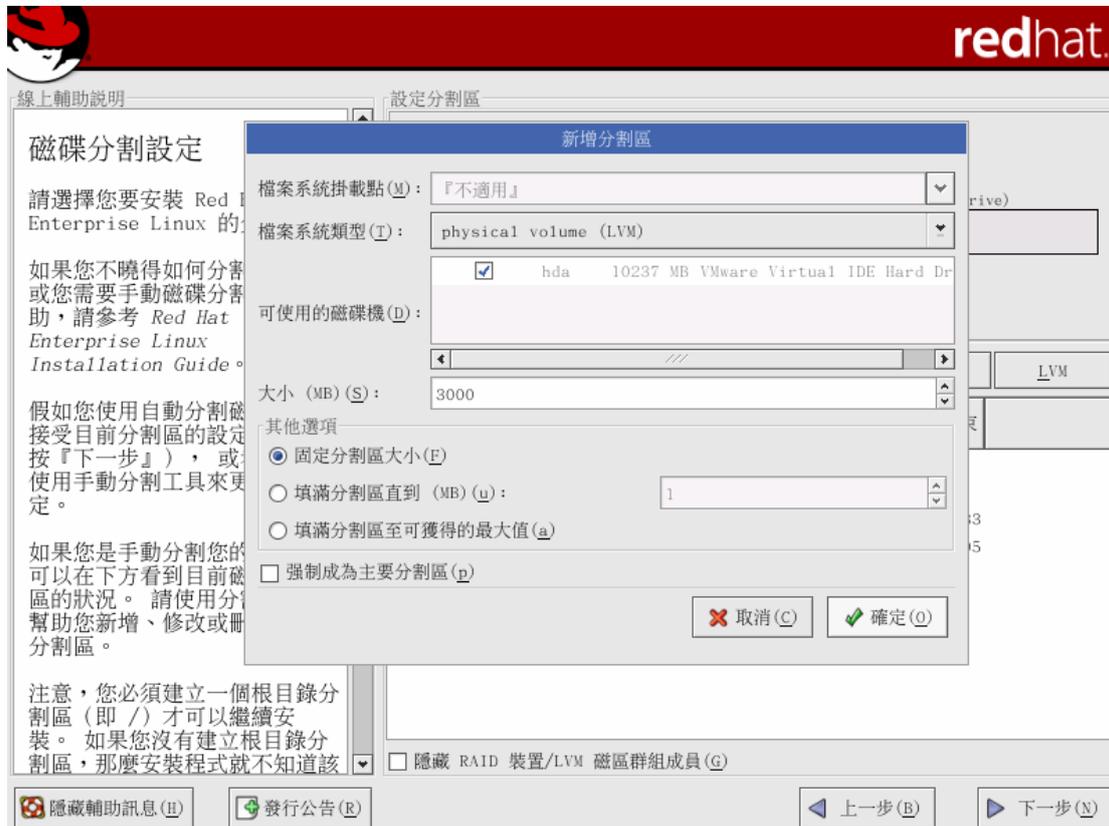


圖 8、新增 Physical Volume (PV)

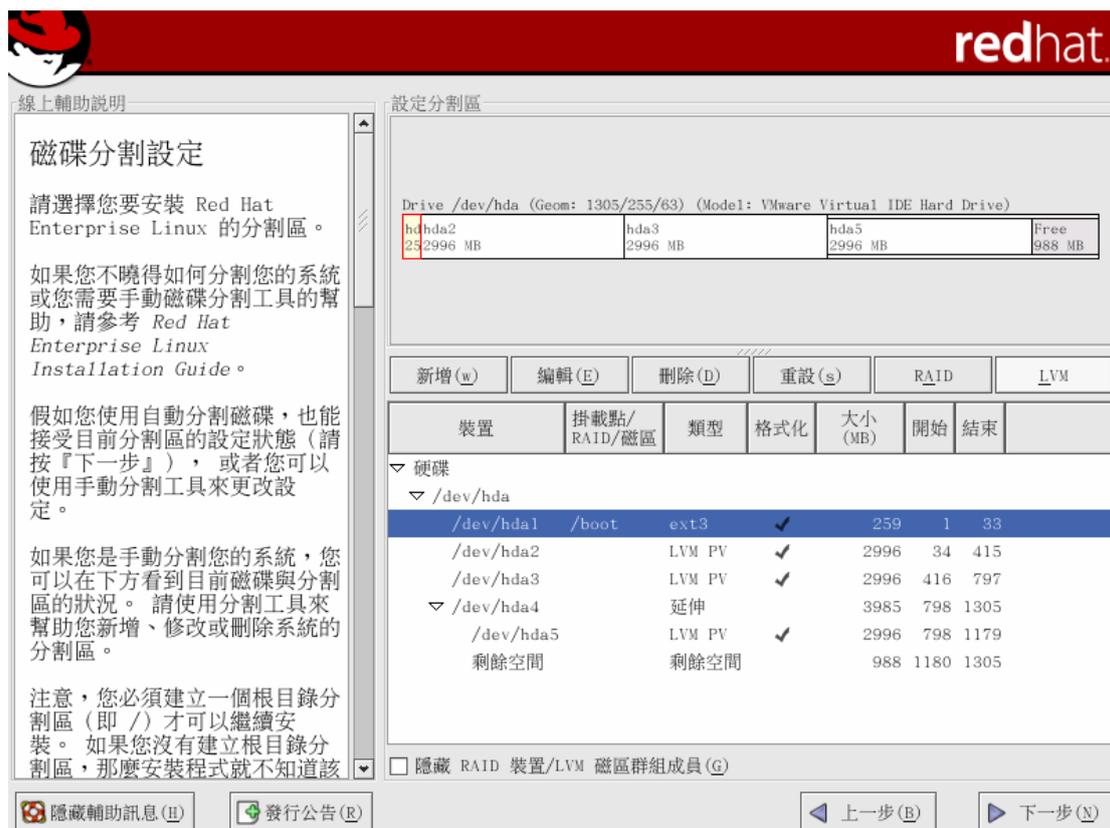


圖 9、建立三個 Physical Volume (PV)

3. 將這三個 Physical Volume (PV) 組合成一個 VG，將此 VG 命名為 rootvg。各位可以把此時的 rootvg 想成是一個虛擬硬碟，地位就如同 hda。所以接下來我們的工作便是指定每個檔案系統的空間。

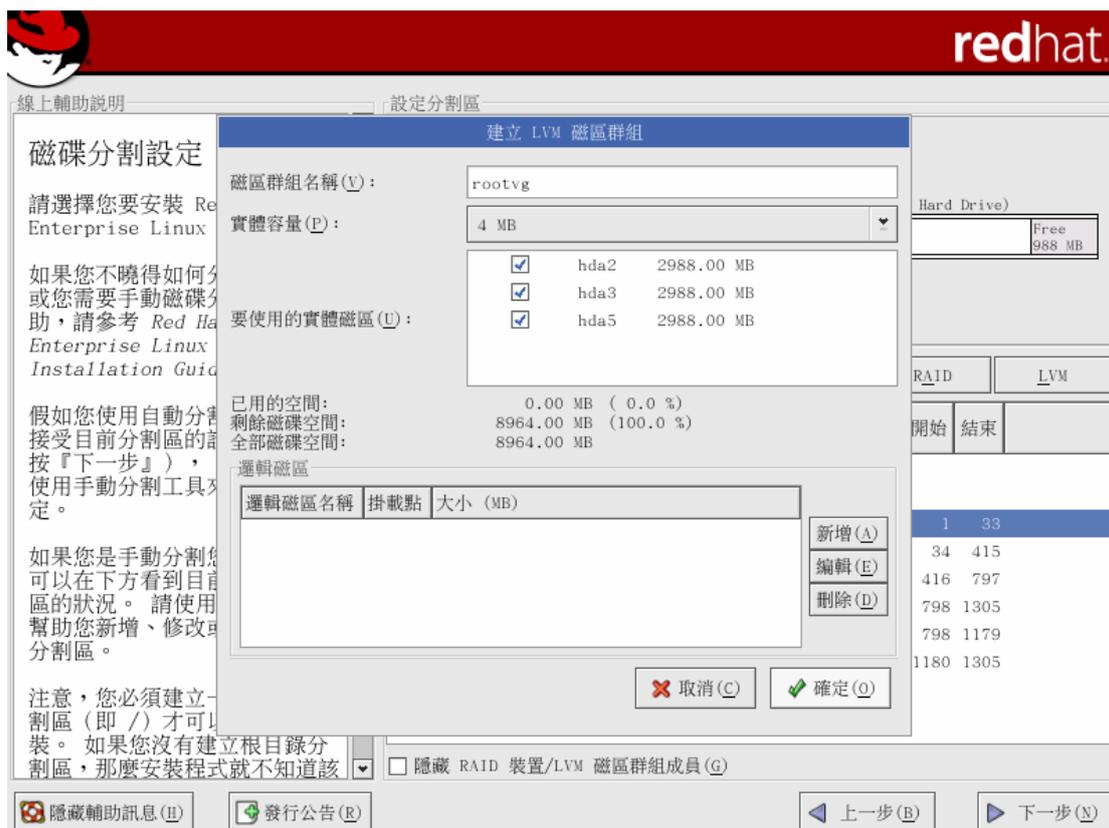


圖 10、將 PV 組合成 VG ( rootvg )

4. 在 LVM 機制，當你要建立一個檔案系統時，必須先新增一個 LV ( 很像以前的 partition hda# )。新增 LV 時，LVM 會從 VG 中找出未使用的空間組合成你需要的大小，這些空間實際可能對應到任一個硬碟；但使用者不用去在意真正存放在那一個硬碟。也正因這種 Logical 的方式，我們才可將檔案系統的容量放大，因為每個檔案系統 對應的已不是從前實體硬碟的 Partition ( 從第幾個磁柱到第幾個磁柱 )，而是 Logical 概念的 LV。只要 VG 還有未用的空間，LVM 就可將其對應到某個 LV，所以 LV 的大小就可增加，藉此達到檔案系統容量可動態放大的功能。

如圖 11~15，筆者指定：

rootlv = 6000MB → / 檔案系統

homelvj = 500MB → /home 檔案系統

swaplv = 500MB → SWAP

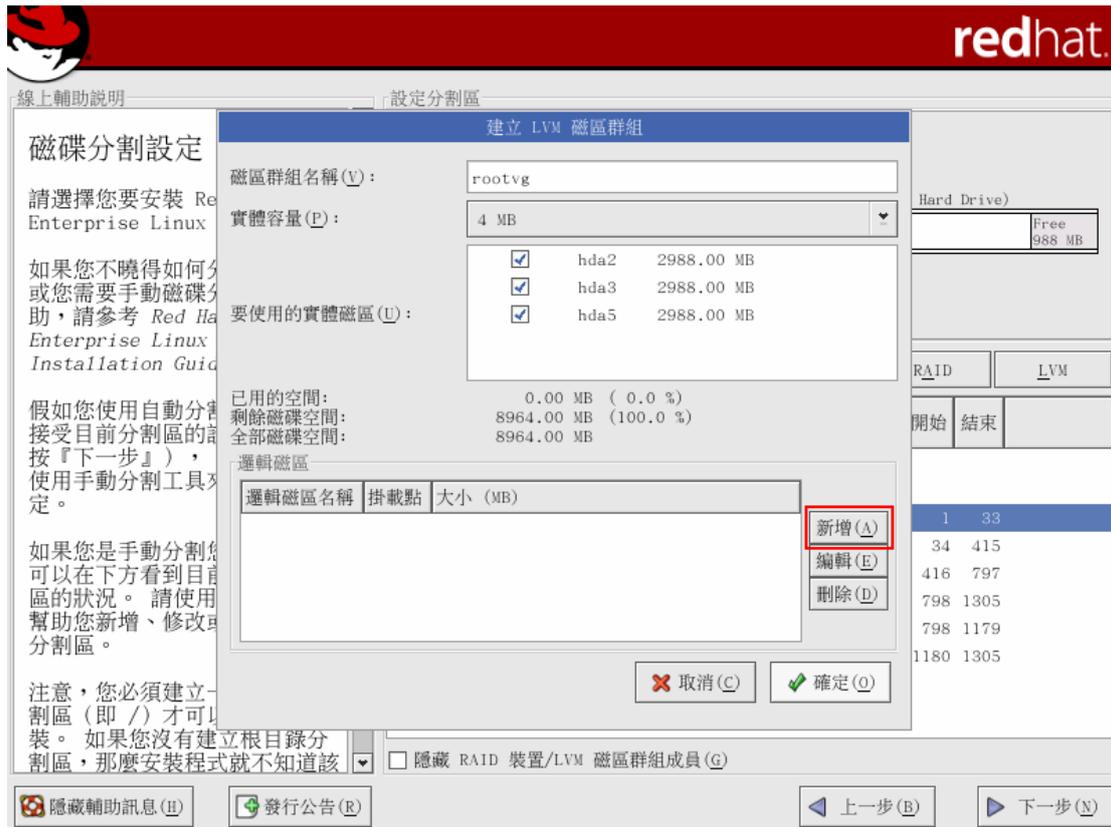


圖 11、新增邏輯磁區 (LV)



圖 12、新增邏輯磁區名稱 rootlv、其掛載點設定為 /



圖 13、新增邏輯磁區名稱 home1v、其掛載點設定為 /



圖 14、新增 swap 邏輯磁區



圖 15、Disk Druid 分割完成圖

## 5. 安裝套件為預設選項 (圖 16)



圖 16、套件採用預設的選項

安裝完畢開機時各位會看到下列訊息，便代表 LVM 設定成功。

```
Scanning logical volumes
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- found inactive volume group "rootvg"
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
.....
Activating logical volumes
vgchange -- volume group "rootvg" successfully activated
```

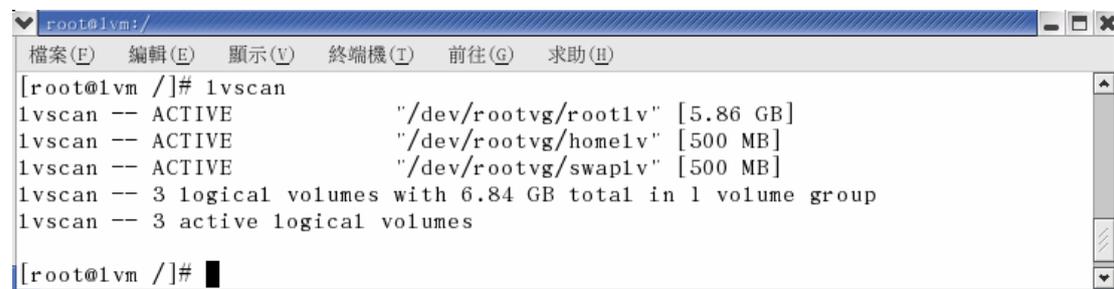
6.開機後利用 `df -h` 檢查硬碟、LVM、檔案系統的狀況。



```
[root@lvm ~]# df -h
檔案系統          容量  已用  可用  已用% 掛載點
/dev/rootvg/rootlv  5.8G  1.9G  3.7G   35%  /
/dev/hdal          251M   14M  225M    6%  /boot
/dev/rootvg/home1v  485M   8.1M  452M    2%  /home
none              125M    0  125M    0%  /dev/shm
/dev/cdrom         626M  626M    0  100% /mnt/cdrom
[root@lvm ~]#
```

圖 17、/home 檔案系統容量約 500MB

7.利用 `lvscan` 檢查原來 LV 分配情形



```
[root@lvm ~]# lvscan
lvscan -- ACTIVE          "/dev/rootvg/rootlv" [5.86 GB]
lvscan -- ACTIVE          "/dev/rootvg/home1v" [500 MB]
lvscan -- ACTIVE          "/dev/rootvg/swap1v" [500 MB]
lvscan -- 3 logical volumes with 6.84 GB total in 1 volume group
lvscan -- 3 active logical volumes
[root@lvm ~]#
```

圖 18、利用 `lvscan` 檢查系統正常啓用的邏輯磁區

8.動態將 /home 檔案系統 放大 100MB

```
#umount /home
#e2fsadm -L+100 /dev/rootvg/home1v
註：-L+100 代表增加 100MB
#mount /home
```

```

root@lvm: /
檔案(E) 編輯(E) 顯示(V) 終端機(T) 前往(G) 求助(H)
[root@lvm /]# umount /home
[root@lvm /]# e2fsadm -L +100 /dev/rootvg/homelv 將 homelv 放大100 MB
e2fsck 1.32 (09-Nov-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/homelv: 11/128016 files (0.0% non-contiguous), 24395/512000 blocks
lvextend -- extending logical volume "/dev/rootvg/homelv" to 600 MB
lvextend -- doing automatic backup of volume group "rootvg"
lvextend -- logical volume "/dev/rootvg/homelv" successfully extended

resize2fs 1.32 (09-Nov-2002)
Begin pass 1 (max = 12)
Extending the inode table      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 2 (max = 2)
Relocating blocks             XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 3 (max = 63)
Scanning inode table          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 5 (max = 54)
Moving inode table            XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/rootvg/homelv is now 614400 blocks long.

e2fsadm -- ext2fs in logical volume /dev/rootvg/homelv successfully extended to 600 MB 成功放大100MB
[root@lvm /]# mount /home

```

圖 19、將/home 檔案系統放大 100MB 過程圖

#lvscan → 檢查變更後 LV 分配情形

#df -H 檢查檔案系統使用情形

```

root@lvm: /
檔案(E) 編輯(E) 顯示(V) 終端機(T) 前往(G) 求助(H)
[root@lvm /]# df -h
檔案系統 容量 已用 可用 已用% 掛載點
/dev/rootvg/rootlv 5.8G 1.9G 3.7G 35% /
/dev/hdal 251M 14M 225M 6% /boot
none 125M 0 125M 0% /dev/shm
/dev/cdrom 626M 626M 0 100% /mnt/cdrom
/dev/rootvg/homelv 582M 8.3M 544M 2% /home
[root@lvm /]# 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
[root@lvm /]# █

```

圖 20、/home 檔案系統容量變為 600MB

讀者此時可以檢查原來在 /home 目錄的資料並不會因此而不見。在 AIX 上甚至可以不用做 umount 就將檔案系統放大，筆者認為未來 Linux 也可達到此功能。不過只需簡單 umount → e2fsadm → mount 就可將檔案系統放大，不用似傳統的解決方法需要冗長的停機時間，讓 Linux 更有資格擔任企業關鍵性的伺服器。

## 後記

IBM AIX 上為人稱道的 LVM 機制，如今順利地移植到 Linux，讓 Linux 向企業級的應用再邁向一步。在本文，我們先介紹如何利用 Disk Druid 來實作 LVM，不過 Disk Druid 這個工具只有在安裝系統時才能使用。如果一開始安裝時未利用 Disk Druid 設定 LVM 就得利用指令的方式來設定 LVM，這部份筆者將會在 LVM(二)講解。