Red Hat Enterprise Linux 6
虛擬伺服器管理

Red Hat Enterprise Linux 的 Load Balancer 外掛程式
版 6

Landmann

Red Hat Enterprise Linux 6 虛擬伺服器管理

Red Hat Enterprise Linux 的 Load Balancer 外掛程式
版 6

Landmann
rlandmann@redhat.com

摘要

若建立 Load Balancer 外掛程式，系統可透過 PIRANHA 配置工具來配置專門的 Linux Virtual Server（LVS），以進行路由和負載平衡技術，並為生產服務提供高可用性與高延展性的解決方案。本書詳述了 Red Hat Enterprise Linux 高效能系統與服務的配置，以及 Red Hat Enterprise Linux 6 的 Load Balancer 外掛程式配置。

內容目錄

簡介

本文件提供了關於安裝、配置和管理 Load Balancer 外掛程式元件的相關資訊。Load Balancer 外掛程式透過了專門的路由技術，將網路流量導向至一群伺服器上，達成負載平衡的目的。

本文件的讀者應該要有 Red Hat Enterprise Linux 的進階知識，並了解叢集、儲存裝置、以及伺服器運算的概念。

本文件的架構如下：

- 章 1, *Load Balancer 外掛程式總覽*
- 章 2, *初始 Load Balancer 外掛程式配置*
- 章 3, *設定 Load Balancer 外掛程式*
- 章 4, *以 **Piranha Configuration Tool** 配置 Load Balancer 外掛程式*
- 附錄 A, *搭配使用 Load Balancer 外掛程式與 High Availability 外掛程式*

欲知關於 Red Hat Enterprise Linux 6 的相關資訊，請參閱以下資源：

- *Red Hat Enterprise Linux 安裝指南* — 提供關於安裝 Red Hat Enterprise Linux 6 的資訊。
- *Red Hat Enterprise Linux 建置指南* — 提供建置、配置、與管理 Red Hat Enterprise Linux 6 的資訊。

欲取得更多有關於 Load Balancer 外掛程式以及 Red Hat Enterprise Linux 6 相關產品的相關資訊，請參閱以下資源：

- *Red Hat Cluster Suite 總覽* — 提供了 High Availability 外掛程式、Resilient Storage 外掛程式，以及 Load Balancer 外掛程式的高層總覽。
- *配置與管理 High Availability 外掛程式*提供了有關於為 Red Hat Enterprise Linux 6 配置與管理 High Availability 外掛程式（亦稱為 Red Hat Cluster）的相關資訊。
- *邏輯卷冊管理指南* — 描述邏輯卷冊管理員（LVM，Logical Volume Manager），包括在叢集環境裡執行 LVM 的資訊。
- *全球檔案系統 2：配置與管理* — 提供了安裝、配置與維護 Red Hat Resilient Storage 外掛程式（亦稱為 Red Hat Global File System 2）的相關資訊。
- *DM Multipath* — 提供了有關於 Red Hat Enterprise Linux 6 Device-Mapper Multipath 功能使用上的相關資訊。
- *發行公告* — 提供了目前最新版本的 Red Hat 產品的相關資訊。

本文件與其它 Red Hat 文件（格式為 HTML、PDF、與 RPM 版本）皆可在 Red Hat Enterprise Linux 文件光碟中，或線上 http://www.redhat.com/docs/ 找到。

1. 文件規範

本指南使用了幾種規範，以強調特定文字與詞組，並提示讀者注意特定的資訊。

在 PDF 和書面版本中，本指南使用了來自於 Liberation Fonts 字體組的 typefaces。倘若 Liberation Fonts 字體已安裝在您系統上的話，該字體也會被使用於 HTML 版本中。若是沒有的話，本指南便會顯示其它相對應的 typefaces 字體。請注意：預設上 Red Hat Enterprise Linux 5 以上的版本已包含 Liberation Fonts 字體。

1.1. 排版規範

此處使用四種排版規範，用來強調特定文字與詞組。這些規範及其適用情況如下。

固定寬度粗體字型（`Mono-spaced Bold`）

用來強調系統的輸入字元，包括 shell 指令、檔案名稱與路徑。同時也會被使用來強調按鍵與組合鍵。例如：

> 若要查看位於您目前工作目錄中的 **my_next_bestselling_novel** 檔案的話，請在 shell 提示符號下輸入 **cat my_next_bestselling_novel** 指令，再按下 **Enter** 鍵執行該指令。

以上包含了一個檔案名稱、一組 shell 指令，以及一個按鍵，並且全部以固定寬度粗體字型來顯示。

「組合鍵」可以透過每個按鍵及之間的「加號」來組合表示。例如：

> 請按下 **Enter** 鍵執行指令。
>
> 請按下 **Ctrl**+**Alt**+**F2** 切換至虛擬終端機。

第一段強調了使用者應按下特定的按鍵，第二段則表示使用者應按下一組組合鍵，亦即同時按下三個按鍵。

若討論到原始碼的話，段落中所提及的 class 名稱、method、function、variable 名稱與回傳值，都將會如上一般地以「**固定寬度粗體字型**」顯示。例如：

> 和檔案相關的 class，其中包含了 **filesystem**（檔案系統）、**file**（檔案）以及 **dir**（目錄）。各個 class 都有著與它關聯的權限組。

浮動寬度粗體字型（Proportional Bold）

這代表在系統上所會看見的文字或詞組，這包含了應用程式名稱、對話方塊文字、被標記的按鈕、核取方塊與收音機按鈕的標籤、選單標題以及子選單標題。例如：

> 由主選單選取「**系統** → **偏好設定** → **滑鼠**」來啓動「**滑鼠偏好設定**」。請在「**按鈕**」分頁中點選「**左手操作滑鼠**」核取方塊並按下「**關閉**」，將主滑鼠按鍵由左鍵切換至右鍵（這可讓滑鼠適合以左手使用）。
>
> 要將一個特殊字元插入至 **gedit** 檔案，請從主選單選擇「**應用程式** → **附屬應用程式** → **字元對應表**」。然後從「**字元對應表**」的選單中選擇「**搜尋** → **尋找**」，接下來在「**搜尋**」欄位裡輸入字元名稱，然後按下「**下一個**」。您所搜尋的字元會在「**字元表**」中反白出現。雙擊這個字元，這樣它會出現在「**準備複製的文字**」欄位，然後請按下「**複製**」按鈕。現在請切換到您的文件，然後從 **gedit** 的選單選擇「**編輯** → **貼上**」。

以上文字包含了應用程式名稱、系統全域的選單名稱與項目、應用程式特屬的選單名稱、以及在 GUI 介面中所看到的按鈕與文字等，全部皆以浮動寬度粗體字型來顯示，並且可透過內文來辨別。

Mono-spaced Bold Italic（固定寬度粗體斜體字型）或是 ***Proportional Bold Italic***（浮動寬度粗體斜體字型）

不管是固定或浮動寬度的粗體字，加上斜體後便表示可替換的文字或是變數文字。斜體字型代表您不會照字面輸入的文字，或是會依照情況而改變的文字。比方說：

> 若要透過使用 ssh 連至遠端機器，請在 shell 提示符號下輸入 **ssh *username*@*domain.name***。若遠端機器為 **example.com** 而您在該機器上的用戶名稱為 john 的話，請輸入 **ssh john@example.com**。
>
> **mount -o remount *file-system*** 指令會將 file-system 檔案系統重新掛載。比方說，若要重新掛載 **/home** 檔案系統的話，指令為 **mount -o remount /home**。
>
> 使用 **rpm -q *package*** 指令來查看目前已安裝套件的版本。系統將會回傳此結果：***package-version-release***。

請注意上方的重體兼斜體字 — username、domain.name、file-system、package、version、以及

release。這些字都是可以取代的：不是您下指令時所打的文字，就是系統所顯示的文字。

除了用來顯示工作標題這樣的標準用法，斜體字也可代表第一次出現的重要新詞彙。比方說：

Publican 是一種 *DocBook* 發佈系統。

1.2. 引述規範

終端機輸出與原始碼資料會列在文字框裡面。

傳送至終端機的輸出設置為 **mono-spaced roman**（固定寬度 roman 字型），並且以此方式顯示：

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

原始碼排列亦設置為 **mono-spaced roman**（固定寬度 roman 字型），並以彩色強調語法：

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                  struct kvm_assigned_pci_dev *assigned_dev)
{
         int r = 0;
         struct kvm_assigned_dev_kernel *match;

         mutex_lock(&kvm->lock);

         match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                       assigned_dev->assigned_dev_id);
         if (!match) {
                 printk(KERN_INFO "%s: device hasn't been assigned before, "
                   "so cannot be deassigned\n", __func__);
                 r = -EINVAL;
                 goto out;
         }

         kvm_deassign_device(kvm, match);

         kvm_free_assigned_device(kvm, match);

out:
         mutex_unlock(&kvm->lock);
         return r;
}
```

1.3. 注意和警告

最後，我們將使用三種視覺上的形式，來強調可能會被遺漏掉的資訊。

注意

「注意」代表某些作業上的提示、捷徑或是其它完成工作的方式。忽略「注意」並不會帶來太大的負面影響，不過您可能會忽略掉某些能夠較輕鬆完成工作的方式。

重要

「重要」方塊會將容易遺漏掉的項目詳細列出：例如變更配置只會套用到目前的作業階段，或是要套用更新則必須重新啓動服務等等。倘若您忽略掉重點方塊，雖然不會造成資料遺失，不過卻會造成工作上的不便與其它影響。

警告

讀者不該忽略任何「警告」。忽略「警告」很有可能會造成資料遺失。

2. 意見

若您在本指南中發現了任何錯誤，或是您希望提供任何改善本指南的意見，非常歡迎您與我們取得聯繫！請透過 Bugzilla（http://bugzilla.redhat.com/bugzilla/），針對於 **Documentation-cluster** 元件提交報告。

請記得提供指南的識別元：

```
Virtual_Server_Administration(EN)-6 (2010-10-14T16:28)
```

若您提供本指南的識別元，我們便能清楚知道您所擁有的指南版本為何。

若您希望提供改善文件上的相關意見，請嘗試盡可能地詳細描述。若您發現了錯誤，請包含章節部份號碼以及相近的文字片段，以方便我們找尋這些錯誤。

章 1. Load Balancer 外掛程式總覽

Load Balancer 外掛程式是一組整合軟體元件，它提供了 Linux Virtual Server（LVS）以進行真實伺服器群之間的 IP 負載平衡。Load Balancer 外掛程式會在一對配置相等的電腦上執行：一個是*啓用中的 LVS 路由器*，另一個則是*備用的 LVS 路由器*。前者負責兩項工作：

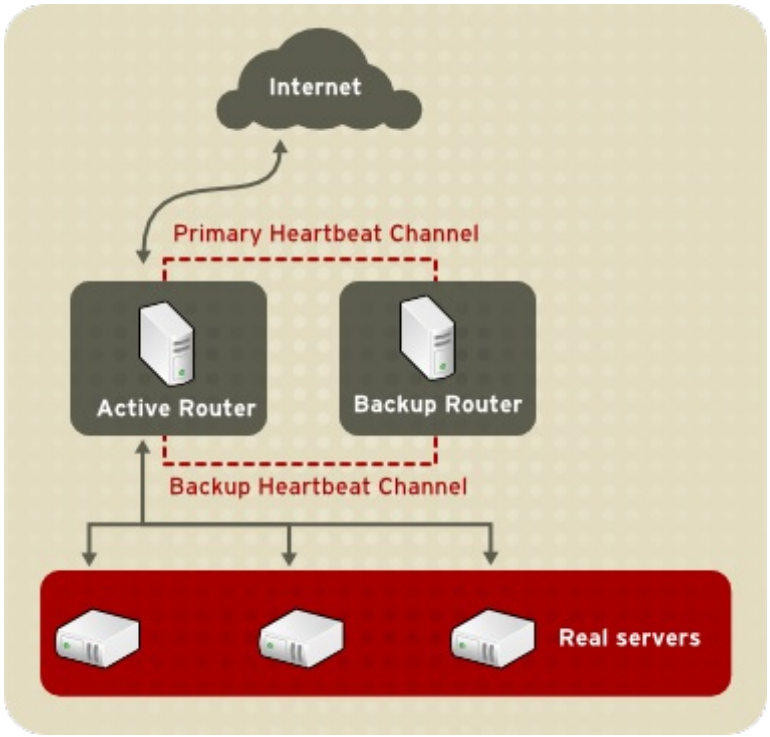
- 在真實伺服器之間，平衡負載。
- 檢查每個真實伺服器的服務之完整性。

備用 LVS 路由器會監控啓用中的 LVS 路由器，如果它失效的話，就會接手工作。

本章提供了 Load Balancer 外掛程式元件與功能的總覽，並包含以下部份：

1.1. 基本 Load Balancer 外掛程式配置

圖形 1.1, “基本 Load Balancer 外掛程式配置” 顯示了基本的 Load Balancer 外掛程式配置，並且包含了兩個 layer。第一層包含了兩個 LVS 路由器 — 一個是啓用中的 LVS 路由器，一個則是備用的 LVS 路由器。各個 LVS 路由器皆有兩張網路介面卡，一張連上網際網路，另一張則連上私密網路。這能讓它們調節兩個網路之間的網路流量。在此範例中，啓用中的路由器使用了*網路位址轉譯*（或 *NAT*），將網際網路的流量轉送至第二層上的數個真實伺服器中，並提供必要的服務。因此，此範例中的真實伺服器連至了一個專用的私密網路區段，並且會透過啓用中的 LVS 路由器，來回傳送所有的公共流量。由外部網路用戶端的角度來看，該伺服器會被視為是單一個體。



圖形 1.1. 基本 Load Balancer 外掛程式配置

送抵 LVS 路由器的服務需求會被指定一組*虛擬 IP* 位址，簡稱 *VIP*。這是組公開、可路由的位址，也是網站管理者以一組完整網域名稱（例如 www.example.com）與其相聯的位址，並且它會被指定至一或多台*虛擬伺服器*上。虛擬伺服器（virtual server）是一項設定來監聽特定虛擬 IP 的服務。欲取得更多有關於使用 **Piranha Configuration Tool** 來配置虛擬伺服器上的相關資訊，請參閱 節 4.6, “虛擬伺服器”。當進行備援（failover）時，一台 LVS 路由器的 VIP 位址會轉移至另一台 LVS 路由器上，如此一來便可保留這組 IP 位址（亦稱為*浮動 IP 位址*。

VIP 位址可以是將 LVS 路由器連至網際網路的裝置之別名。舉例來說，如果 eth0 連到了網際網路，那麼有多個虛擬伺服器的別名皆可被設為 **eth0:1**。此外，各個虛擬伺服器皆可與個別的裝置與服務相聯。比方說，HTTP 的流量可在 **eth0:1** 上處理，並且 FTP 流量可在 **eth0:2** 上處理。

一次只有一台啓用的 LVS 路由器。啓用中的 LVS 路由器負責將由虛擬 IP 位址的服務請求導向至真實伺服器上。這種導向是基於受支援的八種負載平衡驗算法則之一，節 1.3, “Load Balancer 外掛程式排程總覽” 中含有詳細描述。

同時，啓用中的 LVS 路由器會動態監控真實伺服器的特定服務之整體健康狀態，方法是透過基本的 *send/expect scripts*（發送/預期回應的程序檔）。若要協助偵測需要動態資料的服務的健康狀態（例如 HTTPS 或 SSL），管理員可調用外部可執行檔。如果真實伺服器上有項服務無法運作，啓用中的路由器就會停止發送工作到該伺服器上，直到它回歸正常為止。

備用路由器會扮演待命系統的角色。LVS 路由器會定期透過主要的外部公用介面卡，交換 heartbeat 訊息，同時在備援情況發生時，透過私有介面卡交換訊息。如果備用節點在預期的時間內收不到 heartbeat 訊息，它便會開始備援，並成為啓用中的路由器。在備援過程中，備用的路由器會使用 *ARP spoofing* 這項技術，接手失效路由器的 VIP 位址 — 藉由這項技術，備用 LVS 路由器會宣佈它自己是送往失效節點 IP 封包位址的目的地。當失效節點又開始啓動服務時，備用節點便會恢復為它備援的角色。

使用於 圖形 1.1, “基本 Load Balancer 外掛程式配置” 中的基本兩層式設定適用於不常變動資料的叢集上 — 例如靜態網頁 — 因為各別的真實伺服器不會在各個節點之間自動同步資料。

1.1.1. 在真實伺服器之間進行資料複製與共享

因為 Load Balancer 外掛程式中沒有內建元件，以共享真實伺服器之間的相同資料，因此管理員擁有兩項基本選項：

- 同步真實伺服器 pool 之間的資料
- 在拓樸中加入第三層，以存取共享資料

對於不允許大量使用者上傳或更改真實伺服器上資料的伺服器來說，第一個選項會比較適合。如果真實伺服器允許大量使用者修改資料，例如電子商務網站，我們建議您使用第三層。

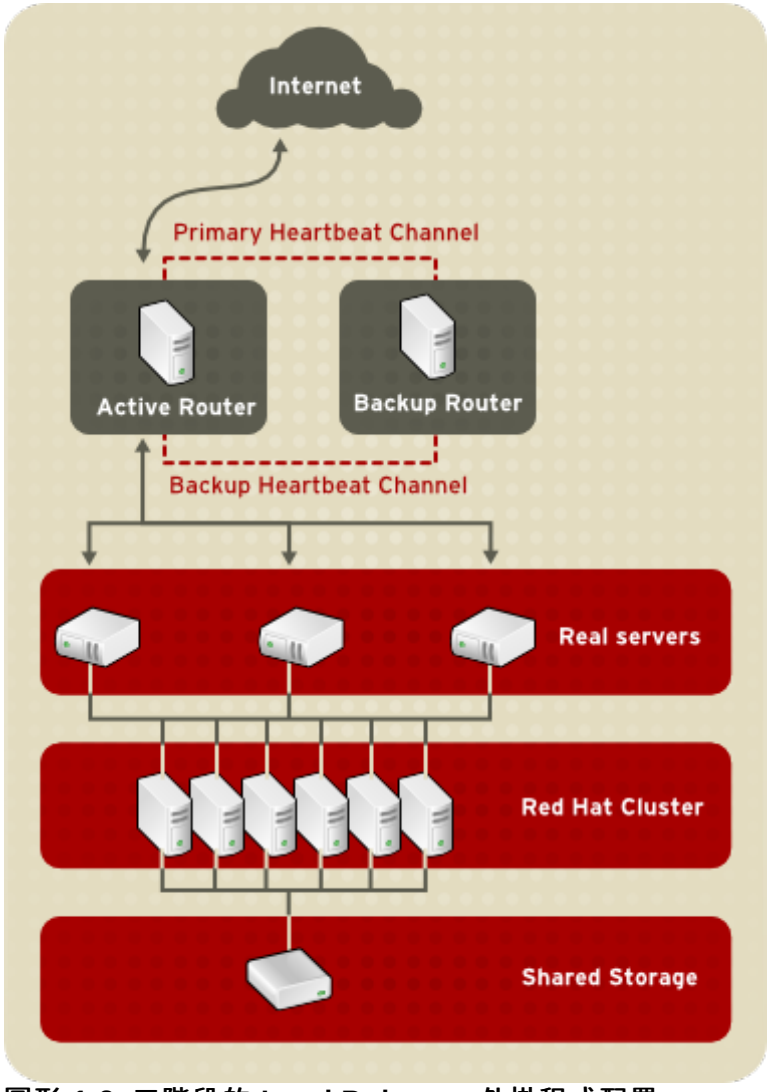
1.1.1.1. 配置真實伺服器以同步資料

系統管理員能選擇透過許多種方式來同步真實伺服器 pool 之間的資料。比方說您可使用 shell script，如此一來，若網站工程人員更新了某個網頁，該網頁將會被同時地發佈至所有的伺服器上。此外，系統管理員亦可使用像是 **rsync** 的程式來定時地在所有節點上複製受到變更的資料。

然而，若有大量使用者持續地上載檔案，或進行資料庫交易的話，此類型的資料同步將無法有效運作。若負載過高，使用*三階段拓撲*會是較適當的方式。

1.2. 三階段的 Load Balancer 外掛程式配置

圖形 1.2, “三階段的 Load Balancer 外掛程式配置” 顯示了典型的三層式 Load Balancer 外掛程式拓撲。在此範例中，啓用中的 LVS 路由器會將請求由網際網路轉送至真實伺服器。各個真實伺服器皆會透過網路，以存取共享資料來源。



圖形 1.2. 三階段的 Load Balancer 外掛程式配置

此配置適用於繁忙的 FTP 伺服器，在此情況下，可存取的資料皆儲存在一個中央、高可用性的伺服器上，並且各個真實伺服器皆會透過一個已匯出的 NFS 目錄或 Samba share 來存取這些資料。此拓樸也適用於存取中央、高可用性資料庫的網站。此外，使用 Red Hat Cluster Manager 的 active-active 配置，管理員可以設定一個高可用性的叢集，以同時地進行這些工作。

上述範例中的第三階段不使用 Red Hat Cluster Manager，不過若是沒有使用高可用性的方式，將會產生嚴重的單點失效（single point of failure）。

1.3. Load Balancer 外掛程式排程總覽

使用 Load Balancer 外掛程式的好處之一，就是它能在一群真實伺服器上進行靈活、IP 層級的負載平衡。此靈活性來自於管理員在配置 Load Balancer 外掛程式時所能選擇的各種排程演算法。Load Balancer 外掛程式負載平衡比其它靈活性較低的方式更為強大，比方說*輪詢排程 DNS*，它特屬的 DNS 階級特性以及來自於客戶端機器的快取，可能會造成負載不平衡。此外，LVS 路由器所使用的低層過濾優於應用程式層級的請求轉送，因為網路封包層級的負載平衡可形成最小的運算負擔，並允許較佳的延展性。

當啓用中的路由器使用排程時，它會考慮到真實伺服器上的動作，以及當轉送服務請求時，管理員所指定的*權重*。使用指定的權重能給予各別的機器任意的優先權。使用此類型的排程，您可透過各種硬體與軟體的組合來建立一個真實伺服器的群組，並且啓用中的路由器便可平均地分散各個真實伺服器上的負載。

Load Balancer 外掛程式的排程機制是由一組稱為 *IP 虛擬伺服器*或 *IPVS* 模組的 kernel 更新檔所提供的。這些模組會啓用 *layer 4*（*L4*）傳輸層切換，這主要設計來與一個單獨 IP 位址上的多重伺服器搭配使用。

若要有效地追蹤和轉送封包至真實伺服器上，IPVS 會在 kernel 中建立一個 *IPVS 表格*。此表格會被啓用中的 LVS 路由器，使用來將來自於虛擬伺服器位址的請求轉送至 pool 中的真實伺服器上，或由真實伺服器上轉送回。IPVS 表格會不斷地透過一個名為 *ipvsadm* 的工具程式更新 — 根據叢集成員的可用性來將其新增或移除。

1.3.1. 排程演算法

IPVS 表格所使用的結構取決於管理員為虛擬伺服器所選擇的排程演算法。若要您可進行叢集的服務以及其排程能有最佳的靈活性，Red Hat Enterprise Linux 提供了以下所列出的排程演算法。欲取得有關於如何指定排程演算法上的相關資訊，請參閱 節 4.6.1, “虛擬伺服器子畫面”。

輪詢排程

按照順序地分配請求給各個真實伺服器。使用此演算法，所有的真實伺服器皆會被視為平等，無論容量或負載為何。此排程模式類似輪詢排程 DNS，不過卻較為精確，因為它是基於網路連線，而不是基於主機。此外，Load Balancer 外掛程式的輪詢排程，不會受到快取 DNS 查詢所造成的負載不平衡的影響。

加權輪詢排程

相繼地將各項請求分配給各個真實伺服器，但會分配更多工作給效能較強大的伺服器。判別效能好壞的方式，在於使用者所給予的權重，權重可以由動態負載資訊來上下調整。欲取得更多有關於伺服器加權上的相關資訊，請參閱 節 1.3.2, “伺服器加權與排程”。

如果真實伺服器之間有著效能上的顯著差異，我們建議使用輪詢排程。然而，如果負載需求的變化非常大，那麼權重高的伺服器可能會收到過多的請求。

最少連線

會將請求分配給連線數較少的電腦。因為它透過 IPVS 表格來追蹤連至真實伺服器的實體連線，因此這是一種動態排程演算法，使其成為請求負載變動高時的較佳選擇。它適用於各個節點效能皆大致相似的真實伺服器群組。如果伺服器群組的效能不同，加權最少連線排程法會是較佳的選擇。

加權最少連線（預設）

將較多請求分配給連線數少的伺服器上（根據它們的效能來進行判斷）。伺服器效能是以使用者所指定的權重為準，權重可以藉由動態負載資訊來上下調整。當使用的伺服器硬體配備不一時，增加權重可以讓最少連線這個演算法成為理想的方案。欲取得更多有關於真實伺服器權重上的相關資訊，請參閱 節 1.3.2, “伺服器加權與排程”。

基於本機的最少連線排程

將較多的請求分配給連線數較少的伺服器上（根據其目的地 IP 位址）。此演算法則適用於代理/快取伺服器叢集上。它會將送往一個 IP 位址的封包導向至擁有該位址的伺服器上；除非該伺服器已經超載，同時另一台伺服器的負載只有一半，這樣它就會將 IP 位址分配給負載較輕的真實伺服。

基於本機的最少連線排程以及複製排程

將較多的請求送往與目的地 IP 相比，連線數較少的伺服器上。此演算法則亦可使用於代理/快取伺服器叢集中。與前述不同的是，它會將目的地 IP 位址映射到真實伺服器節點的子集中。接著請求會被導向至此子集中，連線數最少的伺服器上。如果用有此 IP 位址的所有節點皆超載的話，那麼此演算法便會複製新的伺服器到這個目標 IP 位址，方法是從整體的伺服器群中，將連線數最少的真實伺服器加入該目標 IP 的真實伺服器子集裡。在這之後，最高負載的節點會從真實伺服器的子集中移除

，以避免過度複製。

目的地雜湊排程法

藉由在靜態雜湊表中尋找目的地 IP，來將請求分配至真實伺服器群中。此演算法的設計適用於代理/快取伺服器叢集中。

來源雜湊排程法

藉由在靜態雜湊表中尋找來源 IP，來將請求分配至真實伺服器群中。此演算法的設計適用於含有多重防火牆的 LVS 路由器。

1.3.2. 伺服器加權與排程

Load Balancer 外掛程式的管理員可為真實伺服器群中的各個節點指定*權重*。此權重是個數值，它會被帶入任何*可辨識權重*的排程演算法（例如加權最少連線）中，並協助 LVS 路由器平均分配負載給效能不同的硬體。

權重是以權重和權重之間的比例來計算的。比方說，若有一部真實伺服器的權重為 1，另一部伺服器的權重為 5，那麼每當權重為 1 的伺服器被分配了 1 個連線時，權重為 5 的伺服器便會被分配 5 個連線。真實伺服器的預設權重為 1。

雖然分配權重給真實伺服器群中的各種硬體配置，能有效率地協助平衡叢集的負載，不過當有一部真實伺服器被帶入真實伺服器群中，並且虛擬伺服器被排程使用加權最少連線時，這可能會造成暫時性的負載不平衡。比方說，假設真實伺服器群中含有三部伺服器。伺服器 A 和 B 的權重為 1，第三個伺服器，伺服器 C 的權重為 2。若伺服器 C 基於某些原因停擺，伺服器 A 和 B 便會平均地分配被遺棄的負載。然而，一旦伺服器 C 又重新上線時，LVS 路由器會將它視為零連線，並將所有連入的請求分配給它，直到它與伺服器 A B 同等。

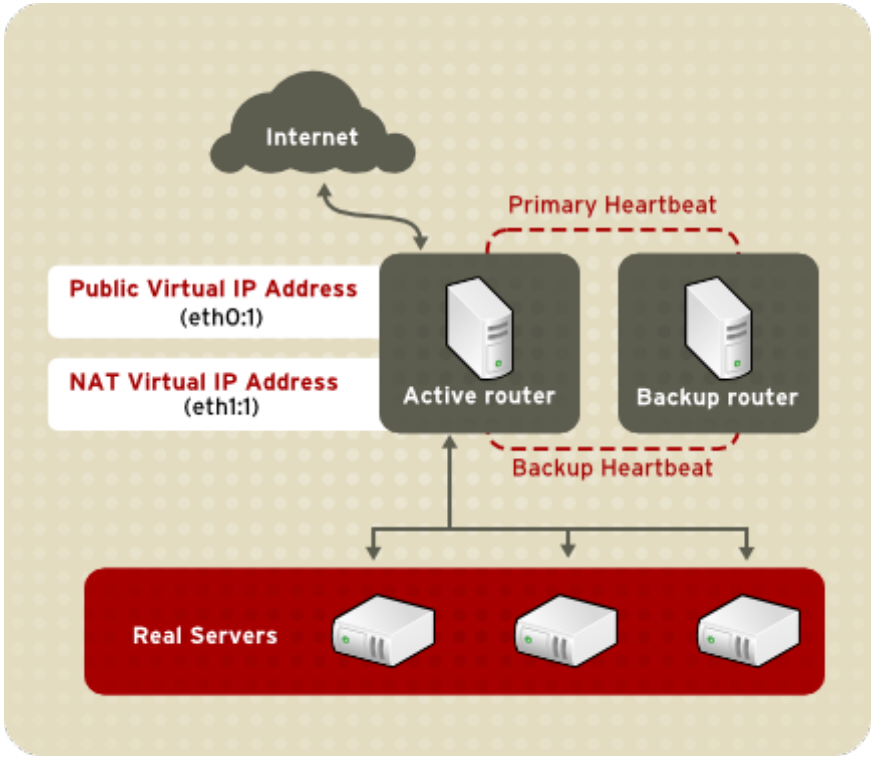
若要避免此情況發生，系統管理員可使虛擬伺服器成為一個 *quiesce*（靜默）伺服器 — 每當新的真實伺服器節點上線時，最少連接表便會重設為零，如此一來 LVS 路由器就會將請求導向，就有如所有真實伺服器都是新加入叢集一般。

1.4. 路由法則

Red Hat Enterprise Linux 使用了*網路位址轉譯*或 *NAT 路由*來進行 Load Balancer 外掛程式，這能讓系統管理員在使用可用的硬體時，以及將 Load Balancer 外掛程式整合入一個既有的網路時，能擁有極佳的靈活性。

1.4.1. NAT 路由

圖形 1.3, “以 NAT 路由轉送實做 Load Balancer 外掛程式”，描述了 Load Balancer 外掛程式如何使用 NAT 路由，以將請求移動於網際網路與私有網路之間。



圖形 1.3. 以 NAT 路由轉送實做 Load Balancer 外掛程式

在此範例中，啓用中的 LVS 路由器有兩張網路卡。連上網際網路的網路卡有位於 eth0 上的*真實 IP 位址*，同時有浮動 IP 位址連到了 eth0:1 上。私有網路的網路卡在 eth1 上含有真實的 IP 位址，也有浮動的 IP 位址連到了 eth1:1 上。在備援情況發生時，網際網路與私有網路的虛擬介面卡會由備用 LVS 路由器所接手。所以私有網路上的真實伺服器皆會使用 NAT 路由器的浮動 IP 做為預設路徑，來與啓用中的 LVS 路由器溝通，如此一來，真實伺服器回應來自於網際網路上的請求時就不會有問題。

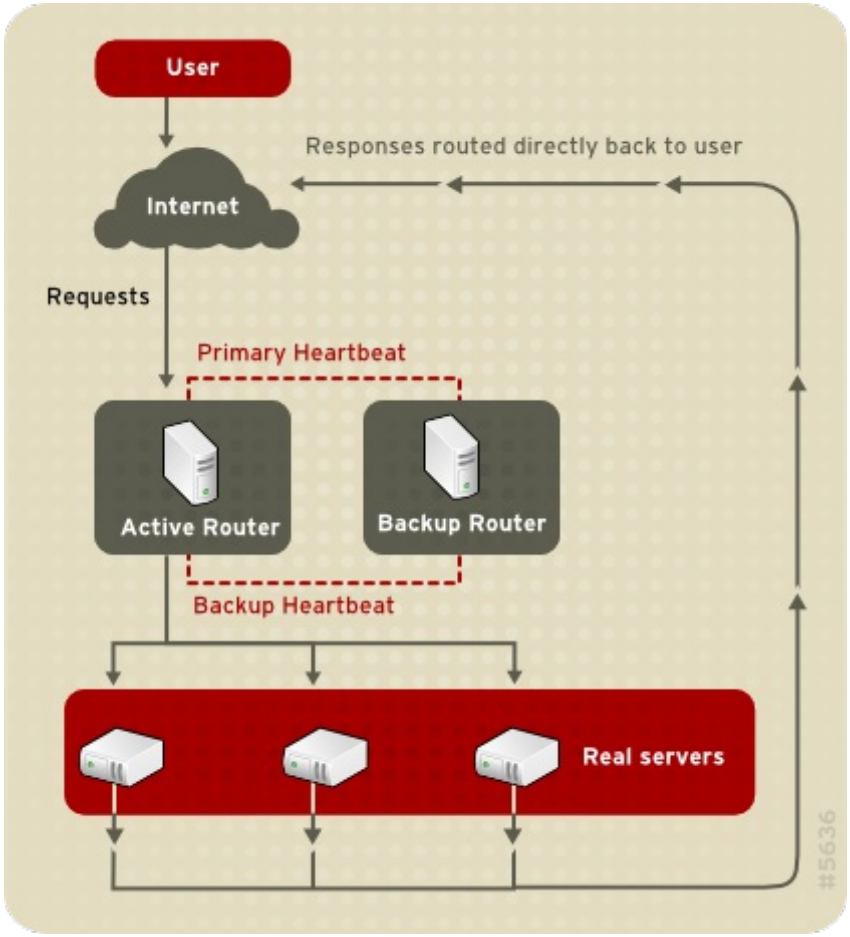
在此範例中，LVS 路由器的公用浮動 IP 位址，與私有 NAT 的浮動 IP 位址是兩張實體網路卡的別名。因為每個浮動 IP 位址都可能與它在 LVS 路由器節點上的實體裝置相聯，所以並不需要使用到超過兩張網路卡。

藉由這種拓樸，啓用中的 LVS 路由器會收到請求，並將其導向到正確的伺服器上。真實伺服器會處理這些請求，然後將封包傳回給 LVS 路由器。LVS 路由器會使用網路位址轉譯，來將封包中真實伺服器的位址替代成 LVS 路由器的公開 VIP 位址。這個過程稱為 *IP masquerading（IP 偽冒）*，因為真實伺服器的位址已被隱藏起來，請求的用戶端是看不見的。

使用此 NAT 路由時，真實伺服器可以是執行各種作業系統的任何電腦。NAT 路由的主要缺點就是，LVS 路由器可能會成為大型叢集建置環境中的瓶頸，因為它必須處理進、出的請求。

1.4.2. 直接路由

與其它 Load Balancer 外掛程式的網路拓撲比較起來，建立一個使用了直接路由的 Load Balancer 外掛程式設定，能提供較佳的效能。直接路由能讓真實伺服器直接處理封包，並直接將其導向至請求的使用者；而非透過 LVS 路由器來傳送所有的連出封包。直接路由藉由將 LVS 路由器的工作轉為只處理進入的封包，降低了網路效能降低的可能性。



圖形 1.4. 以直接路由轉送實做 Load Balancer 外掛程式

在典型的直接路由 Load Balancer 外掛程式設定中，LVS 路由器會透過虛擬 IP（VIP）來接收連入的伺服器請求，並使用排程演算法來將這些請求導向至真實伺服器。每個真實伺服器會處理請求，再將結果直接傳回給用戶端，跳過 LVS 路由器。直接路由能提供延展性，因為真實伺服器可以在不需要增加 LVS 路由器的負擔的情況下加入，協助將連出的封包由真實伺服器送往用戶端；這在網路負載過重的情況下，可能會成為瓶頸。

1.4.2.1. 直接路由轉送和 ARP 限制

雖然在 Load Balancer 外掛程式中使用直接路由有許多好處，但它還是有限制的。其中透過直接路由的 Load Balancer 外掛程式的最常見問題，就是*位址解析協定*（Address Resolution Protocol，ARP）。

在一般的情形下，位於網際網路的使用者會發送請求到一組 IP 位址。網路路由器會利用 ARP 取得目的地的 MAC 位址，並將 MAC 位址及 IP 位址相聯，以發送請求。ARP 會廣播詢問所有連上網路的電腦，擁有正確 IP/MAC 位址的電腦就會收到此封包。IP/MAC 的關聯會儲存在 ARP 快取中，此快取會定時清除（通常是每十五分鐘），然後再被重新填入 IP/MAC 的關聯。

ARP 請求用在直接路由 Load Balancer 外掛程式設定的問題，是因為用戶請求的 IP 必須跟 MAC 位址相聯才能被處理，因此 Load Balancer 外掛程式系統的虛擬 IP 位址也必須與某個 MAC 位址相聯。然而，因為 LVS 路由器與真實伺服器的 VIP 皆為相同的，ARP 請求會廣播到與此 VIP 相聯的所有機器上。這會造成幾項問題，例如 VIP 直接與一台真實伺服器相聯，並直接處理請求。這將會完全跳過 LVS 路由器，並違背了設定 Load Balancer 外掛程式的用意。

要解決這項問題，請確保連入的請求應該總是傳送至 LVS 路由器，而不是任何的真實伺服器。基於下列原因，這可以透過使用 **`arptables_jf`** 或是 **`iptables`** 封包過濾工具來達成：

- **arptables_jf** 可避免 ARP 與真實伺服器的 VIP 相聯。
- **iptables** 這項方式透過了一開始不配置真實伺服器上的 VIP，來完全地避開了 ARP 的問題。

欲取得更多有關於在一個直接路由的 Load Balancer 外掛程式環境中，使用 **arptables** 或 **iptables** 上的相關資訊，請參閱 節 3.2.1, “直接路由與 **arptables_jf**” 或 節 3.2.2, “直接路由與 **iptables**”。

1.5. Persistence 與 Firewall Mark

在某些情況下，我們會希望讓一台用戶端重複連線到同樣的真實伺服器上，而不是由 Load Balancer 外掛程式的負載平衡演算法來將該請求傳送至最佳的伺服器上。這樣的範例有多視窗的網頁表格、cookies、SSL 以及 FTP 連線。在這些範例中，除非連線是由同一台伺服器所處理，否則用戶端多半會連線失敗。Load Balancer 外掛程式提供了兩種方法來處理這項問題：*persistence*（持續連線）與 *firewall mark*（防火牆標記）。

1.5.1. Persistence

Persistence 啓用後，運作起來就像計時器一樣。當用戶端連上一項服務時，Load Balancer 外掛程式會將最後一次的連線記上一段時間。如果同樣的 IP 位址在這段時間內再次連上，用戶端便會被導向至先前連上的同一台伺服器 — 跳過負載平衡機制。如果連線發生在這段時間之外，那麼就會根據排程準則來處理。

Persistence 也能讓管理員指定一組子網路遮罩，以套用至用戶端 IP 位址測試，作為一個用來控制哪些位址應擁有較高的 persistence 等級，並將連線分組至該子網路。

連向不同連接埠的群組連線，對於使用多於一個連接埠來進行通訊的協定（例如 FTP）來說，是很重要的。然而，persistence 並不是用來處理群組連線連往不同連接埠的最好方法。在這種情況下，最好使用 *firewall mark*。

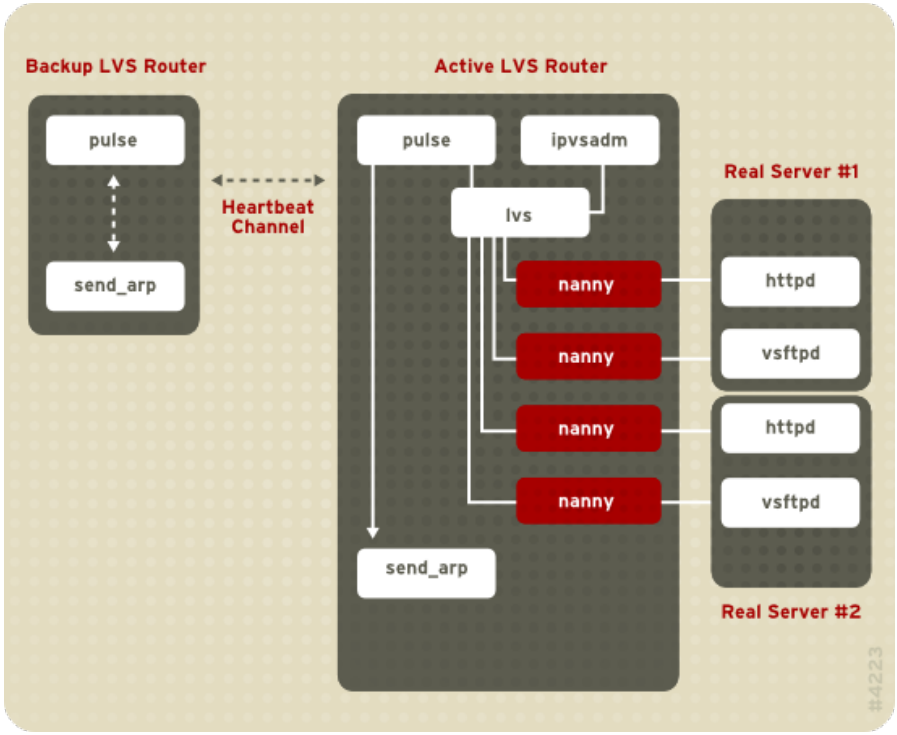
1.5.2. Firewall Mark

Firewall mark 是解決一個或是一組通訊協定使用多連接埠的問題時，最容易又有效率的方法。舉例來說，如果 Load Balancer 外掛程式建置於電子商務網站上，firewall mark 就可用來將 HTTP 連線全部設在連接埠 80；HTTPS 連線設在連接埠 443。透過分配相同的 firewall mark 給各個協定的虛擬伺服器上，交易的狀態資訊就可以被保留住，因為 LVS 會在連線開啓後，將所有的請求轉送往同一台真實伺服器上。

因為此方法既有效率又容易使用，Load Balancer 外掛程式的管理員應該對群組連線盡可能使用 firewall mark 而非 persistence。然而，管理員還是應在虛擬伺服器中加入 persistence 功能，與 firewall mark 搭配運作，以確保用戶端會在一定時間內，重新連上同一台伺服器。

1.6. Load Balancer 外掛程式 — 方塊圖形

LVS 路由器使用了一組程式來監控叢集成員以及叢集服務。圖形 1.5, “Load Balancer 外掛程式元件” 描述了這些在啓用中以及備用 LVS 路由器上的程式，如何互相運作以管理叢集。



圖形 1.5. Load Balancer 外掛程式元件

pulse deamon 會各別在主動式的 LVS 路由器以及被動式的 LVS 路由器上執行。在備用的 LVS 路由器上，**pulse** 會發送 *heartbeat（心跳）* 至啓用中的路由器的公共介面卡，以確認啓用中的路由器依然運作正常。在啓用中的 LVS 路由器上，**pulse** 會啓動 **lvs** daemon，以回應來自於備用 LVS 路由器的 *heartbeat* 查詢。

一旦啓動之後，**lvs** daemon 將會調用 **ipvsadm** 工具程式，以配置、維護 kernel 中的 IPVS（IP 虛擬伺服器，IP Virtual Server）路由表，並為每台真實伺服器上每個已配置好的虛擬伺服器，啓動 **nanny** 程序。每項 **nanny** 程序皆會檢查一部真實伺服器上一項已配置好的服務之狀態，然後告知 **lvs** daemon 是否有任何異常之處。如果偵測到異常，**lvs** daemon 便會指示 **ipvsadm** 由 IPVS 路由表中移除該真實伺服器。

如果備用路由器沒有收到啓用中的路由器所傳來的回應，它會開始啓動備援程序，方法是調用 **send_arp**，以重新指定所有虛擬 IP 位址到備用節點的網路卡硬體位址（*MAC* 位址），透過公開與私有網路介面卡，發送指令到啓用中的 LVS 路由器，以關閉該路由器上的 **lvs** daemon，並啓動備用節點上的 **lvs** daemon，以接受已配置的虛擬機器之請求。

1.6.1. Load Balancer 外掛程式元件

節 1.6.1.1, “**pulse**” 顯示了一列包含了 LVS 路由器中各個軟體的詳細清單。

1.6.1.1. pulse

這是啓動所有其它與 LVS 路由器相關的 daemon 的控制程序。在開機時，此 daemon 會由 **/etc/rc.d/init.d/pulse** script 所啓動。然後它會讀取配置檔案 **/etc/sysconfig/ha/lvs.cf**。在啓用中的 LVS 路由器上，**pulse** 會啓動 LVS daemon。在備用的路由器上，**pulse** 會判斷啓用中的路由器的健康狀態，方法是以使用者配置的間隔，送出心跳訊息（heartbeat）。如果啓用中的 LVS 路由器在此間隔中沒有回應，它便會啓動備援程序。當進行備援程序時，備用路由器上的 **pulse** 會指示啓用中的路由器的 **pulse** daemon 關閉所有 LVS 服務，並啓動 **send_arp** 程式以重新指定浮動 IP 位址給備用路由器的 MAC 位址，然後啓動 **lvs** daemon。

1.6.1.2. lvs

一旦被 **pulse** 調用，**lvs** daemon 便會在 active LVS 路由器上執行。它會讀取配置檔案 **/etc/sysconfig/ha/lvs.cf**、調用 **ipvsadm** 工具程式來建立、維護 IPVS 路由表，並為每個已設定的 Load Balancer 外掛程式服務指定 **nanny** 程序。如果 **nanny** 回報一台伺服器無法使用，**lvs** 便會指示 **ipvsadm** 工具程式將真實伺服器由 IPVS 路由表中移除。

1.6.1.3. ipvsadm

這項服務會更新核心裡的 IPVS 路由表。**lvs** daemon 會設定、管理 Load Balancer 外掛程式，方法是調用 **ipvsadm** 來新增、改變或刪除 IPVS 路由表中的項目。

1.6.1.4. nanny

nanny 監控 daemon 會在啓用中的 LVS 路由器上執行。透過此 daemon，啓動中的 LVS 路由器會判斷各個真實伺服器的健康狀況，然後（選用性地）監控伺服器的工作負載。各個真實伺服器上所定義的各項服務，皆有一項執行的獨立程序。

1.6.1.5. /etc/sysconfig/ha/lvs.cf

這是 Load Balancer 外掛程式的配置檔案。無論是直接或間接，所有的 daemon 皆會由此檔案取得其配置資訊。

1.6.1.6. Piranha Configuration Tool

這是監控、設定、管理 Load Balancer 外掛程式的網站工具。這是維護 **/etc/sysconfig/ha/lvs.cf** Load Balancer 外掛程式配置檔案的預設工具。

1.6.1.7. send_arp

在發生備援時，浮動 IP 位址會從一個節點改到另一個節點。此時這個程式會送出 ARP 廣播。

章 2, 初始 Load Balancer 外掛程式配置 包含了您在將 Red Hat Enterprise Linux 配置成 LVS 路由器之前，所應進行的重要安裝後配置步驟。

章 2. 初始 Load Balancer 外掛程式配置

在安裝了 Red Hat Enterprise Linux 之後，您必須進行一些基本步驟以設定 LVS 路由器與真實伺服器。此章節詳細涵蓋了這些初始步驟。

注意

當 Load Balancer 外掛程式啓動之後，成為啓用節點的 LVS 路由器節點亦稱為*主要節點*。進行 Load Balancer 外掛程式的配置時，請在主要節點上使用 **Piranha Configuration Tool**。

2.1. 在 LVS 路由器上進行服務配置

Red Hat Enterprise Linux 安裝程式會安裝設定 Load Balancer 外掛程式所需的所有元件，不過在配置 Load Balancer 外掛程式之前，您必須先啓用適當的服務。請針對於兩部 LVS 路由器設置適當的服務，以便使其在開機時啓動。在 Red Hat Enterprise Linux 中有三個可用來將服務設為能在開機時啓用的主要工具：指令列程式 **chkconfig**、基於 ncurses 的程式 **ntsysv** 以及圖形化的 **Services Configuration Tool**。所有的這些工具皆需要 root 存取權限。

注意

若要取得 root 權限，請開啓一個 shell 提示列，並使用 **su -** 加上 root 密碼。例如：

```
$ su - root 密碼
```

在 LVS 路由器上，有三項需要被設為在開機時會被啓用的服務：

- **piranha-gui** 服務（唯有在主要節點上）
- **pulse** 服務
- **sshd** 服務

若您正在進行 multi-port 服務的叢集或是使用防火牆標記的話，您也必須將 **iptables** 服務啓用。

建議您將這些服務設為啓用於 runlevel 3 以及 runlevel 5 中。若要這麼作，請使用 **chkconfig**，請為各項服務輸入下列指令：

```
/sbin/chkconfig --level 35 daemon on
```

在以上指令中，請將 ***daemon*** 取代為您欲啓用的服務之名稱。若希望取得系統上的服務清單，並得之它們啓用於哪個 runlevel 上，請輸入下列指令：

```
/sbin/chkconfig --list
```

警告

透過使用 **chkconfig** 來啓動上述任何一項服務並不會實際地啓動 daemon。若要這麼作，請使用 **/sbin/service** 指令。請查看 節 2.3, “啓用 Piranha Configuration Tool 服務” 以取得如何使用 **/sbin/service** 指令的範例。

欲取得更多有關於 runlevel 和透過 **ntsysv** 與 **Services Configuration Tool** 來配置服務上的相關資訊，請參閱 *Red Hat Enterprise Linux System Administration Guide* 中，標題為「*Controlling Access to Services*」的章節。

2.2. 為 Piranha Configuration Tool 設定密碼

第一次在主要的 LVS 路由器上使用 **Piranha Configuration Tool** 之前，您必須透過建立一組密碼來限制其存取權限。若要這麼作，請以 root 身份登入並輸入下列指令：

```
/usr/sbin/piranha-passwd
```

在輸入了這項指令後，請在提示出現時建立管理員密碼。

警告

若要確保密碼的安全性，它不該包含名詞、常用的縮寫，或是任何語言的字典詞彙。請勿以未加密的方式將密碼保留在系統上的任何地方。

若密碼在一個有效的 **Piranha Configuration Tool** session 中遭到更改的話，管理員將會被提示輸入新的密碼。

2.3. 啓用 Piranha Configuration Tool 服務

當您為 **Piranha Configuration Tool** 設置了一組密碼後，請啓用或重新啓用位於 **/etc/rc.d/init.d/piranha-gui** 中的 **piranha-gui** 服務。若要這麼作，請以 root 身份輸入下列指令：

```
/sbin/service piranha-gui start
```

或是

```
/sbin/service piranha-gui restart
```

輸入這項指令將會透過調用 **/usr/sbin/piranha_gui -> /usr/sbin/httpd** 這個符號連結來啓動一個私密的 Apache HTTP Server session。基於安全性上的考量，**httpd** 的 **piranha-gui** 版本將會在一個不同的程序中以 piranha 使用者的身份執行。**piranha-gui** 與 **httpd** 服務之間的槓桿效應即代表：

1. Apache HTTP Server 必須被安裝在系統上。
2. 透過 **service** 指令來停用或重新啓用 Apache HTTP Server 將會停用 **piranha-gui** 服務。

警告

若在一個 LVS 路由器上輸入 **/sbin/service httpd stop** 或 **/sbin/service httpd restart** 指令的話，您必須透過輸入下列指令來啓用 **piranha-gui** 服務：
/sbin/service piranha-gui start

若要開始配置 Load Balancer 外掛程式，使用 **piranha-gui** 服務便已足夠。然而，若您要遠端配置 Load Balancer 外掛程式的話，您便也需要使用到 **sshd** 服務。直到透過使用 **Piranha Configuration Tool** 所進行的配置完成以前，您都*無須*啓用 **pulse** 服務。欲取得更多有關於啓用 **pulse** 服務上的相關資訊，請查看 節 4.8, “啓用 Load Balancer 外掛程式”。

2.3.1. 配置 Piranha Configuration Tool Web Server Port

Piranha Configuration Tool 就預設值會在 port 3636 上執行。若要更改此連接埠號，請修改 **piranha-gui** Web server configuration file **/etc/sysconfig/ha/conf/httpd.conf** 第二部份中的 **Listen 3636** 這一行。

若要使用 **Piranha Configuration Tool**，您至少需要一個純文字的網站瀏覽器。若您要在主要 LVS 路由器上啓用一個網站瀏覽器的話，請開啓 **http://*localhost*:3636** 這個位置。您可在任何地方藉由將 ***localhost*** 更改為主要 LVS 路由器的主機名稱或 IP 位址，以透過網站瀏覽器來連上 **Piranha Configuration Tool**。

當您的瀏覽器連至了 **Piranha Configuration Tool** 時，您必須登入才可存取配置服務。請在「**使用者名稱**」欄位中輸入 **piranha**，並在「**密碼**」欄位中輸入以 **piranha-passwd** 所設置的密碼。

現在 **Piranha Configuration Tool** 已開始運作，您可開始考慮限制何人能透過網路來存取此工具。下個部份中詳述了如何完成這項作業。

2.4. Piranha Configuration Tool 的存取權限

Piranha Configuration Tool 會提示輸入一組有效的使用者名稱與密碼組合。然而，因為所有傳送至 **Piranha Configuration Tool** 的資料皆為純文字，因此我們建議您只開放對於信任網路或本機的存取權限。

最易於使用的存取限制方式就是使用 Apache HTTP Server 的內建存取控制機制，並編輯 **/etc/sysconfig/ha/web/secure/.htaccess**。在修改了該檔案之後，您無須重新啓用 **piranha-gui** 服務，因為伺服器會在它每次存取該目錄時檢查 **.htaccess** 檔案。

就預設值，此目錄的存取控制允許任何人檢視該目錄的內容。預設的存取權限看似：

```
Order deny,allow
Allow from all
```

若要將 **Piranha Configuration Tool** 限制為只能存取 localhost，請修改 **.htaccess** 檔案，並只允許來自於 loopback 裝置（127.0.0.1）的存取。欲取得更多有關於 loopback 裝置上的相關資訊，請查看 *Red Hat Enterprise Linux Reference Guide* 中的 *Network Scripts* 章節。

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

您亦可允許特定主機或子網，如以下範例所示：

```
Order deny,allow
Deny from all
Allow from 192.168.1.100
Allow from 172.16.57
```

在此範例中，只有來自於 IP 位址為 192.168.1.100、以及 172.16.57/24 網路上的機器的網站瀏覽器可存取 **Piranha Configuration Tool**。

警告

編輯 **Piranha Configuration Tool** **.htaccess** 檔案將會限制 **/etc/sysconfig/ha/web/secure/** 目錄中的配置分頁的存取權限，不過卻不會影響到 **/etc/sysconfig/ha/web/** 中的登入與說明分頁。若要限制此目錄的存取權限，請在 **/etc/sysconfig/ha/web/** 目錄中建立一個 **.htaccess** 檔案，並將 **order**、**allow** 以及 **deny** 這幾行修改成和 **/etc/sysconfig/ha/web/secure/.htaccess** 一樣。

2.5. 啓動封包轉送（Packet Forwarding）

若要 LVS 路由器能正確地將網路封包轉送至真實伺服器上的話，各個 LVS 路由器節點都必須將 kernel 中的 IP 轉送開啓。請以 root 身份登入並將 **/etc/sysctl.conf** 中的 **net.ipv4.ip_forward = 0** 一行更改為：

```
net.ipv4.ip_forward = 1
```

當您重新啓動系統後，變更將會即刻生效。

若要檢查 IP 轉送是否已開啓，請以 root 身份輸入下列指令：

```
/sbin/sysctl net.ipv4.ip_forward
```

若以上指令回傳了 **1**，那麼這就代表 IP 轉送已啓用。若回傳了 **0** 的話，您可透過使用以下指令來手動式地將它開啓：

```
/sbin/sysctl -w net.ipv4.ip_forward=1
```

2.6. 在真實伺服器上配置服務

若真實伺服器是 Red Hat Enterprise Linux 系統的話，請將適當的伺服器 daemons 設定為在開機時會被啓用。這些 daemon 可包含網站服務的 **httpd** 或是 FTP（或）Telnet 的 **xinetd**。

您可能也會需要遠端存取真實伺服器，因此您應該安裝並執行 **sshd** daemon。

章 3. 設定 Load Balancer 外掛程式

Load Balancer 外掛程式包含了兩個基本群組：LVS 路由器與實體伺服器。要避免單點失效的問題，每個群組應該包含至少兩組成員系統。

LVS 路由器群組應該包含兩台相同或非常類似的 Red Hat Enterprise Linux 系統。一台是主 LVS 路由器，而另一台會處於熱備援模式，因此兩者的能力必須相當。

在選擇與配置實體伺服器群組的硬體之前，請先決定要採用三種 Load Balancer 外掛程式拓樸中的哪一種。

3.1. NAT Load Balancer 外掛程式網路

NAT 拓樸能讓使用者善用現有硬體；但現有硬體還是受限於處理大量負載的能力，因為所有進出的封包都會通過 Load Balancer 外掛程式路由器。

網路佈局

從網路佈局的角度來看，使用 NAT 路由的 Load Balancer 外掛程式是最容易配置的拓樸，因為連到網際網路只有一個存取點。位於私有網路上的實體伺服器會透過 LVS 路由器，將所有的請求傳回。

硬體

就硬體層面來說，NAT 拓樸是最具有彈性的，因為實體伺服器不需要是 Linux 電腦，也可以正確運作。在 NAT 拓樸中，每個實體伺服器只需要一張網路卡，用來回應 LVS 路由器。另一方面，LVS 路由器則需要兩張網路卡，在兩個網路之間進行路由。因為這個拓樸會在 LVS 路由器上產生網路瓶頸，所以 LVS 路由器上可以安裝 gigabit 的乙太網路卡，以增加路由器的網路處理能力。如果 LVS 路由器上安裝了 gigabit 乙太網路卡，任何連接實體伺服器與 LVS 路由器的交換器都必須有兩組以上的 gigabit 乙太網路連接埠，才能有效處理負載。

軟體

因為 NAT 拓樸在某些配置中需要使用 **iptables**，所以除了 **Piranha Configuration Tool** 之外，還有不少軟體配置需要進行。尤其是 FTP 服務與 firewall mark 需要手動配置 LVS 路由器，這樣才能正確路由封包。

3.1.1. 配置搭配了 NAT 的 Load Balancer 外掛程式的網路介面卡

若要設定搭配 NAT 的 Load Balancer 外掛程式，您必須先配置 LVS 路由器上的公用與私有網路介面卡。在此範例中，LVS 路由器的公用網路介面卡（**eth0**）將位於 192.168.26/24 網路（這並非可路由的 IP 位址，但我們將假設 LVS 路由器前有個防火牆）上，而連上實體伺服器的私有介面卡（**eth1**）將位於 10.11.12/24 網路上。

因此，在啓用中、或*主要* LVS 路由器節點上，公用介面卡的網路 script **/etc/sysconfig/network-scripts/ifcfg-eth0** 會看似：

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.26.9
NETMASK=255.255.255.0
GATEWAY=192.168.26.254
```

LVS 路由器上的私有 NAT 介面卡的 **/etc/sysconfig/network-scripts/ifcfg-eth1** 會看似：

```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.11.12.9
NETMASK=255.255.255.0
```

在此範例中，LVS 路由器的公用網路之虛擬 IP 為 192.168.26.10，而 NAT 或私有網路的虛擬 IP 為 10.11.12.10。因此，實體伺服器的封包請求必須要能回傳至 NAT 介面卡的虛擬 IP。

重要

本節的乙太網路介面卡配置範例是給 LVS 路由器的實際 IP 位址用的，而「*不是*」浮動 IP 位址。要配置空開與私有浮動 IP 位址，系統管理者應該使用 **Piranha Configuration Tool**，如 節 4.4, “全域設定” 與 節 4.6.1, “虛擬伺服器子畫面” 所示。

在配置了主 LVS 路由器節點的網路介面卡之後，請配置備用 LVS 路由器的真實網路介面卡 — 請注意這些 IP 位址不會跟網路上任何電腦的 IP 位址互相衝突。

重要

請確定備用節點上的每一張介面卡皆能為主節點所在的同一個網路提供服務。舉例來說，如果 eth0 連上主節點上的公用網路，它也必須連上備用節點上的公用網路。

3.1.2. 真實伺服器上的路由轉送

在 NAT 拓樸中配置實體伺服器的網路介面卡時，一定要記得設定 LVS 路由器的 NAT 浮動 IP 位址。在此範例中為 10.11.12.10。

注意

一旦真實伺服器的網路介面卡設定妥當，真實伺服器就無法用其它方法 ping 或連接到公用網路上。這是正常的。然而，您可以 ping LVS 路由器的私有介面卡之實際 IP 位址，在本例中為 10.11.12.8。

因此真實伺服器的 **/etc/sysconfig/network-scripts/ifcfg-eth0** 檔案會看似：

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.11.12.1
NETMASK=255.255.255.0
GATEWAY=10.11.12.10
```

警告

如果真實伺服器有超過一組網路介面卡，配置了 **GATEWAY=** 一行，那麼只有第一組才會連到閘道器。因此，若配置了 **eth0** 與 **eth1**，而 **eth1** 用於 Load Balancer 外掛程式，那麼真實伺服器將可能無法正確地路由轉送網路請求。
最好的方法是關閉額外的網路介面卡，方法是在 **/etc/sysconfig/network-scripts/** 目錄的網路 script 中設定 **ONBOOT=no**，或確定第一組網路卡的閘道器設定正確。

3.1.3. 在 LVS 路由器上啓用 NAT 路由轉送

在基本的 NAT Load Balancer 外掛程式配置中，每個叢集服務皆只使用一組連接埠，例如 HTTP 使用埠號 80，系統管理者只需要在 LVS 路由器上啓用封包轉送功能，就可以正確地在外部世界與真實伺服器間進行路由轉送。欲取得如何開啓封包轉送功能上的相關資訊，請參閱 節 2.5, “啓動封包轉送（Packet Forwarding）”。然而，當叢集服務需要超過一組連接埠，以連往同樣的真實伺服器時，將需要進行更多的配置。欲取得如何使用 firewall mark 來建立多連接埠服務的相關資訊，請參閱 節 3.4, “多連接埠服務與 Load Balancer 外掛程式”。

一旦在 LVS 路由器上啓用了轉送功能、真實伺服器已設定完畢，並且叢集服務已處於執行狀態時，請使用 **Piranha Configuration Tool** 來配置 Load Balancer 外掛程式，如 章 4, *以 **Piranha Configuration Tool** 配置 Load Balancer 外掛程式* 中所述。

警告

請不要手動編輯網路 script 或使用網路配置工具來配置 **eth0:1** 或 **eth1:1** 的浮動 IP。請如 節 4.4, “**全域設定**” 與 節 4.6.1, “**虛擬伺服器子畫面**” 中所示，使用 **Piranha Configuration Tool**。

完成後，請設定 **pulse**，如 節 4.8, “啓用 Load Balancer 外掛程式” 中所示。一旦 **pulse** 設定好並處於執行狀態後，啓用中的 LVS 路由器便會將請求導向至真實伺服器群上。

3.2. 透過直接路由進行 Load Balancer 外掛程式

如 節 1.4.2, “直接路由” 中所述，直接路由能讓真實伺服器直接處理封包，並將封包轉送給發出請求的使用者，而不透過 LVS 路由器。直接路由需要真實伺服器直接連到 LVS 路由器的網路區段，同時也要能處理、導向向外的封包。

網路佈局

在直接路由 Load Balance 外掛程式的設定中，LVS 路由器需要收到進入的請求，然後將這些請求路由轉送至適當的真實伺服器上以進行處理。真實伺服器接下來需要「*直接*」將回應路由轉送至用戶端。因此，舉例來說，若用戶端位於網際網路上，並透過 LVS 路由器來將封包轉送至真實伺服器上，真實伺服器就必須要能夠透過網際網路直接與用戶端連上。這可透過為真實伺服器配置閘道器，以將封包轉送至網際網路上。伺服器群中的每台真實伺服器皆有各別的閘道器（而且每個閘道器都有自己連線至網際網路的方式），如此便可提供最大的網路頻寬與擴充性。然而，對於典型的 Load Balancer 外掛程式設定來說，真實伺服器可透過一個閘道器（也就是一組網路連線）來與外界進行通訊。

重要

我們「*不建議*」您使用 LVS 路由器來當作真實伺服器的閘道器，因為那會增加不必要的設定複雜度，以及 LVS 路由器的網路負載，這會重新引發 NAT 路由轉送中既有的網路問題。

硬體

使用直接路由的 Load Balancer 外掛程式系統的硬體需求與其它 Load Balancer 外掛程式拓樸相似。LVS 路由器需要執行 Red Hat Enterprise Linux 以處理傳送入的請求，並為真實伺服器進行負載平衡；真實伺服器並不一定要是 Linux 電腦，才能運作無誤。LVS 路由器需要一或多張網路卡，端視有否備用路由器而定。您可使用兩張網路卡以簡化配置，並將網路流量分開 — 進入的請求由一張網路卡處理，路由轉送至真實伺服器的封包則由另一張網路卡處理。

由於真實伺服器會跳過 LVS 路由器，直接把封包轉送回給用戶端，所以您需要一個閘道器連往網際網路。為了最佳效能與高可用性，每台真實伺服器皆可連上自己的獨立閘道器，每個閘道器皆有自己專用的連外網路（例如網際網路，或是企業內部網路），以便與用戶端進行通訊。

軟體

除了 **Piranha Configuration Tool**，還有些配置需完成，尤其是透過直接路由使用 Load Balancer 外掛程式時，面臨 ARP 問題的管理者更需要注意。詳情請參閱 節 3.2.1, “直接路由與 **arptables_jf**” 或 節 3.2.2, “直接路由與 **iptables**”。

3.2.1. 直接路由與 arptables_jf

若要透過使用 **arptables_jf** 來配置直接路由，每台真實伺服器皆必須配置虛擬 IP 位址，這樣才能直接路由轉送封包。針對 VIP 的 ARP 請求會被真實伺服器完全忽略，任何要送出、包含 VIP 的 ARP 封包皆會被更改，不再包含 VIP 而是包含真實伺服器的 IP 位址。

使用 **arptables_jf** 時，應用程式可能會與真實伺服器所服務的各別 VIP 或連接埠綁定在一起。舉例來說，**arptables_jf** 這個方法能讓 Apache HTTP Server 的多個執行個體明確地與系統上的各別 VIP 綁定執行。和 **iptables** 選項相比較之下，使用 **arptables_jf** 會有顯著的效能提昇。

然而，使用 **arptables_jf** 方法，VIP 無法透過使用標準的 Red Hat Enterprise Linux 系統配置工具，設定成在開機時啓動。

若要將每台真實伺服器配置為忽略每個虛擬 IP 位址的 ARP 請求，請進行以下步驟：

1. 為每台真實伺服器的每個虛擬 IP 位址建立 ARP 列表（real_ip 是與真實伺服器通訊的 IP 位址，通常會是 **eth0** 的 IP 位址）：

   ```
   arptables -A IN -d <virtual_ip> -j DROP
   arptables -A OUT -s <virtual_ip> -j mangle --mangle-ip-s <real_ip>
   ```

   這會導致真實伺服器忽略所有針對虛擬 IP 位址的 ARP 請求，並改變任何向外的 ARP 回應，把原先包含虛擬 IP 的 ARP 回應改為包含真實伺服器的 IP。唯一應該要回應任何 VIP 的 ARP 請求的節點，為目前啓用中的 LVS 節點。
2. 當在每台真實伺服器上完成此步驟之後，請在每台真實伺服器上輸入下列指令，以儲存 ARP 表格項目：

   ```
   service arptables_jf save
   chkconfig --level 2345 arptables_jf on
   ```

chkconfig 指令會讓系統在開機時重新載入 arptables 配置 — 在網路啓動之前。

3. 使用 **ifconfig**，在所有真實伺服器上配置虛擬 IP 位址，以建立 IP 別名。例如：

```
# ifconfig eth0:1 192.168.76.24 netmask 255.255.252.0 broadcast
192.168.79.255 up
```

或使用 **iproute2** 的工具程式 **ip**，例如：

```
# ip addr add 192.168.76.24 dev eth0
```

正如之前所述，虛擬 IP 位址不能透過 Red Hat 系統配置工具來設定為在開機時啓動。解決方式是在 **/etc/rc.d/rc.local** 中加入這些指令。

4. 配置 Piranha 以進行直接路由。詳情請參閱 *章 4, 以 **Piranha Configuration Tool** 配置 Load Balancer 外掛程式*。

3.2.2. 直接路由與 iptables

您也可以建立 **iptables** 防火牆規則，使用直接路由法來解決 ARP 的問題。要使用 **iptables** 來配置直接路由，您必須新增規則，以建立通透的代理服務。如此一來真實伺服器便可以服務送往 VIP 位址的封包，即使 VIP 位址不存在系統上亦然。

iptables 的方法會比配置 **arptables_jf** 簡單。此方法亦可完全避免 LVS 的 ARP 問題，因為虛擬 IP 位址只存在於啓用中的 LVS 導向程式上。

然而，與 **arptables_jf** 相較之下，使用 **iptables** 會有效能上的問題，因為每個封包皆需要經過轉送與更換表頭的過程。

您也可以使用 **iptables** 來重新使用連接埠。舉例來說，要讓兩組 Apache HTTP Server 服務同時使用 80 號連接埠，是不可能的事情，因為這兩組服務皆與 ***INADDR_ANY*** 綁定，而不是虛擬 IP 位址。

要使用 **iptables** 配置直接路由，請進行以下步驟：

1. 在每一台真實伺服器上，請為真實伺服器提供服務的每組 VIP、連接埠、與通訊協定（TCP 或 UDP）執行以下指令：

   iptables -t nat -A PREROUTING -p <tcp|udp> -d <vip> --dport <port> -j REDIRECT

   這項指令會導致真實伺服器處理送往 VIP 與連接埠的封包。

2. 儲存每台真實伺服器上的配置：

```
# service iptables save
# chkconfig --level 2345 iptables on
```

以上指令會讓系統在開機時重新載入 **iptables** 配置 — 在網路啓動之前。

3.3. 結合所有配置

在決定了要使用哪種路由方法之後，硬體應透過網路連在一起。

重要

LVS 路由器上的介面卡裝置必須配置好，以存取同樣的網路。舉例來說，如果 **eth0** 連上公用網路，而 **eth1** 連上了私有網路，那麼備用 LVS 路由器上的同樣裝置，也應該連上對應、相同的網路。同時，開機時第一個介面卡所列出的閘道器會被加入路由表；並且在其它介面卡中列出的閘道器則會被忽略。這在配置真實伺服器時，是非常重要的考量因素。

硬體連上之後，請配置主 LVS 與備用 LVS 路由器的網路介面卡。這可透過 **system-config-network** 的圖形應用程式完成，或手動編輯網路 script。欲取得 **system-config-network** 使用上的相關資訊，請參閱《*Red Hat Enterprise Linux 建置指南*. *網路配置*》一章。該章的其它部份有修改網路介面卡的範例，包括手動修改或透過 **Piranha Configuration Tool** 修改。

3.3.1. Load Balancer 外掛程式網路的一般提示

在嘗試使用 **Piranha Configuration Tool** 配置 Load Balancer 外掛程式之前，請配置 LVS 路由器上的公用和私有網路的真實 IP 位址。每種拓樸的相關章節皆有網路位址的範例，但請依照實際情況來輸入網路位址。以下是一些有用的指令，可用來啓動網路介面卡或檢查其狀態。

啓用真實網路介面卡

若要啓動真實網路介面卡，請以 root 身份執行以下指令，並以實際的網路卡編號取代 ***N***（例如 **eth0** 或 **eth1**）。

```
/sbin/ifup ethN
```

警告

請「*不要*」使用 **ifup** 指令來啓動任何透過 **Piranha Configuration Tool** 所配置的浮動 IP 位址（例如 **eth0:1** 或 **eth1:1**）。請使用 **service** 指令來啓動 **pulse**（詳情請參閱 節 4.8, “啓用 Load Balancer 外掛程式”）。

停用真實網路介面卡

若要停用真實網路介面卡，請以 root 身份執行以下指令，並以網路介面卡的數字取代 ***N***（例如 **eth0** 與 **eth1**）。

```
/sbin/ifdown ethN
```

檢查網路介面卡的狀態

如果您在任何時候，想要檢查哪一張網路卡處於啓用狀態，請執行：

```
/sbin/ifconfig
```

要檢視電腦的路由表，請執行以下指令：

```
/sbin/route
```

3.4. 多連接埠服務與 Load Balancer 外掛程式

建立多連接埠的 Load Balancer 外掛程式服務時，任何拓樸下的 LVS 路由器都需要經過額外的配置。透過 firewall mark，多連接埠的服務可手動建立，將不同、但相關的協定（例如 HTTP 的 80 連接埠與 HTTPS 的 443）結合在一起；或是直接使用真實的多連接埠協定（例如 FTP）。無論是哪一種情況，LVS 路由器都使用 firewall mark 來辨識送往不同連接埠的封包；但即使使用同樣的 firewall mark，皆應該分開處理。而且，當結合了永久功能（persistence）時，firewall mark 能確保來自用戶端的連線會導向至同樣的主機，只要連線是發生在 persistence 參數所指定的時間內即可。欲取得在虛擬伺服器上設定 persistence 功能的詳細資訊，請參閱 節 4.6.1, “虛擬伺服器子畫面”。

不幸的是，真實伺服器上所使用的負載平衡機制 — IPVS —，會辨認指定給封包的 firewall mark，但它自己無法指定 firewall mark。「*指定*」firewall mark 這項工作必須由網路封包過濾程式 **iptables** 來進行，而這是在 **Piranha Configuration Tool** 之外進行的。

3.4.1. 指定 firewall mark

要為封包指定 firewall mark，通過某個特定連接埠，系統管理員必須使用 **iptables**。

本節描述了如何合併處理 HTTP 與 HTTPS；然而，FTP 是另一個常見的叢集多連接埠協定。如果您將 Load Balancer 外掛程式用於 FTP 服務上，請參閱 節 3.5, “配置 FTP” 以取得配置詳情。

要記得何時使用防火牆規則的基本規則，是 **Piranha Configuration Tool** 裡面每個使用 firewall mark 的通訊協定，都必須有相對應的 **iptables** 規則，以指定記號給網路封包。

在建立網路封包過濾規則時，請確定之前沒有相同的規則。若要這麼作，請開啓 shell 提示列，以 root 身份登入並輸入：

```
/sbin/service iptables status
```

如果 **iptables** 不處於執行狀態，您會馬上看到以下提示。

如果 **iptables** 處於執行狀態，它會顯示防火牆規則。如果規則並不存在，請輸入以下指令：

```
/sbin/service iptables stop
```

如果現有的規則很重要，請在處理之前，檢查 **/etc/sysconfig/iptables** 的內容，並將所需的規則複製至安全的地方。

底下的規則皆指定給同樣的 firewall mark 80 給傳進來、通往浮動 IP 位址 ***n.n.n.n***、連接埠為 80 與 443 的封包。

```
/sbin/modprobe ip_tables
```

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport 80 -j
MARK --set-mark 80
```

```
/sbin/iptables -t mangle-A PREROUTING -p tcp -d n.n.n.n/32 --dport 443 -j
MARK --set-mark 80
```

欲取得將 VIP 指派給公用網路介面卡的相關指示，請參閱 節 4.6.1, “虛擬伺服器子畫面”。並請注意，您必須以 root 身份登入，並在第一次輸入規則之前，載入 **iptables** 模組。

在以上的 **iptables** 指令裡，***n.n.n.n*** 應該以 HTTP 與 HTTPS 虛擬伺服器的浮動 IP 所取代。把送往 VIP 的任何交通與連接埠指定為 firewall mark 80，這會由 IPVS 所辨識並正確轉送，這樣有其淨效益。

警告

以上指令會馬上生效；但開機以後並不會存在。若要確保網路封包篩選的設定在重新開機之後還有作用，請參閱 節 3.6, “儲存網路封包篩選設定”

3.5. 配置 FTP

FTP（檔案傳輸通訊協定，File Transport Protocol）是一種老式、複雜的多連接埠協定，對 Load Balancer 外掛程式環境來說，它會帶來許多問題。要了解這些問題的本質，我們必須了解 FTP 運作的關鍵因素。

3.5.1. FTP 如何運作

以大部分用戶端/伺服器的關係來說，用戶端電腦會透過特定的連接埠，啓動與伺服器之間的連線，然後伺服器會透過該連接埠與用戶端通訊。當 FTP 用戶端連上 FTP 伺服器時，它會透過連接埠 21 連上FTP 伺服器。然後*用戶端*會告訴 FTP *伺服器*要使用*主動式連線*或是*被動式連線*。用戶端所選擇的連線方式決定了伺服器該如何回應，也決定了接下來的工作將使用哪些連接埠。

資料連線的兩種方式為：

主動式連線

建立主動式連線時，*伺服器*會開啓連往用戶端的資料連線，用戶端的連接埠會從 20 改到高範圍的連接埠。所有從伺服器送出的資料，都會透過這組連線送出去。

被動式連線

建立被動式連線時，*用戶端*會要求 FTP 伺服器開啓一個被動式連接埠，這能夠是高於 10,000 的任何連接埠。伺服器接下來會將該連接埠與此特定連線綁定在一起，並將此連接埠號傳送回給用戶端。用戶端接下來便開啓這個新的連接埠，進行資料傳送。每次用戶端請求資料時，都會開啓獨立的資料連線。現在，大部分新的 FTP 用戶端在向伺服器端請求資料時，都會嘗試建立一條被動式連線。

注意

*用戶端*會決定連線方式，而不是伺服器端。這表示要有效使用叢集 FTP，您必須配置 LVS 路由器，使其處理主動式與被動式連線。
FTP 用戶端/伺服器的關係可能會開啓許多 **Piranha Configuration Tool** 與 IPVS 所不知道的連接埠。

3.5.2. 這會如何影響 Load Balancer 外掛程式的路由轉送

IPVS 封包轉送會根據所能辨識的埠號或防火牆標記，允許連線進出叢集。如果叢集外的用戶端試圖開啓 IPVS 所未配置的連接埠，它會放棄這個連線。同樣地，如果真實伺服器試圖回頭向網際網路上開啓連線，使用了 IPVS 無法辨識的連接埠，它也會放棄此連線。這表示*所有*來自網際網路的 FTP 用戶端，皆*必須*有同樣的防火牆標記，而且所有來自 FTP 伺服器的連線皆*必須*正確設定，使用網路封包篩選規則以正確轉送封包。

3.5.3. 建立網路封包的篩選規則

在為 FTP 服務指定任何 **`iptables`** 規則之前，請參閱 節 3.4.1, “指定 firewall mark” 中，有關多連接埠服務的資訊，以及檢查現有網路封包篩選技術的技巧。

以下規則指定了同樣的 firewall mark 21 給 FTP 交通。要讓這些規則運作無誤，請使用 **Piranha Configuration Tool** 的「**虛擬伺服器**」部份，為虛擬伺服器配置連接埠 21；方法是在「**firewall mark**」欄位裡輸入 **21**。詳情請參 節 4.6.1, “虛擬伺服器子畫面”。

3.5.3.1. 主動式連線的規則

主動式連線的規則會告訴 kernel 接受、轉送來自*內部*浮動 IP 位址、連接埠 20 的連線 — 這是 FTP 資料埠的埠號。

以下 **iptables** 指令允許 LVS 路由器接受來自真實伺服器、IPVS 並不知道的向外連線：

```
/sbin/iptables -t nat -A POSTROUTING -p tcp -s n.n.n.0/24 --sport 20 -j MASQUERADE
```

在 **iptables** 指令裡，請將 ***n.n.n*** 替代為 NAT 介面卡的內部網路浮動 IP 的前三個數字，這定義於 **Piranha Configuration Tool**的「**全域設定**」中。

3.5.3.2. 被動式連線的規則

將 firewall mark 指定給被動式連線、從網際網路傳至浮動 IP 的規則，這數字是大範圍的連接埠 — 10,000 到 20,000。

警告

如果您限制了被動式連線的連接埠範圍，您必須配置 VSFTP 伺服器使用相對應的連接埠範圍。方法是將以下幾行附加至 **/etc/vsftpd.conf** 中：

```
pasv_min_port=10000
pasv_max_port=20000
```

您也必須控制伺服器為被動式 FTP 連線的用戶端所顯示的位址。在 NAT 路由的 Load Balancer 外掛程式系統上，請將底下一行加入 **/etc/vsftpd.conf** 以覆蓋真實伺服器的 IP，使其成為 VIP，也就是用戶端再連線時所看到的位址。例如：

```
pasv_address=n.n.n.n
```

請以 LVS 系統的 VIP 位址取代 ***n.n.n.n***。

欲取得如何配置其它 FTP 伺服器上的相關資訊，請參閱相關文件。

這個範圍應該要涵蓋夠廣，以符合大多數情況；然而，您可以增加這個數字，以包括所有可用的非安全性連接埠，方法是將 **10000:20000** 改為 **1024:65535**，指令為：

以下 **iptables** 指令會擁有將任何通往適當連接埠上的浮動 IP 的流量，指定為 firewall mark 21 的效果。其後，這將會被 IPVS 所辨識，並正確轉送：

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport 21 -j MARK --set-mark 21
```

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport 10000:20000 -j MARK --set-mark 21
```

在 **iptables** 指令中，***n.n.n.n*** 應該以 **Piranha Configuration Tool** 的「**虛擬伺服器**」中定義、FTP 虛擬伺服器的浮動 IP 位址所取代。

警告

以上指令會即刻生效，但系統重新開機之後，此設定將不會存在。要確定網路封包的篩選設定能在重新開機後繼續運作，請參閱 節 3.6, “儲存網路封包篩選設定”。

最後，您需要確定服務會在正確的 runlevel 中執行。欲取得相關資訊，請參閱 節 2.1, “在 LVS 路由器上進行服務配置”。

3.6. 儲存網路封包篩選設定

在配置了網路封包篩選程式後，請儲存設定，這樣在開機之後還能繼續生效。若您使用的是 **iptables**，請輸入以下指令：

```
/sbin/service iptables save
```

這會將設定寫入 **/etc/sysconfig/iptables** 中，重新開機之後便可再次載入。

寫入檔案之後，您便可使用 **/sbin/service** 指令來啓動、停止 **iptables**，或檢查其狀態（使用 status 參數）。**/sbin/service** 會自動載入需要的模組。要知道如何使用 **/sbin/service** 指令，請參閱 節 2.3, “啓用 **Piranha Configuration Tool** 服務”。

最後，請確定服務會在正確的 runlevel 中執行。欲取得相關資訊，請參閱 節 2.1, “在 LVS 路由器上進行服務配置”。

下一章將會解釋如何使用 **Piranha Configuration Tool** 來配置 LVS 路由器，並詳述啓用 Load Balancer 外掛程式的所需步驟。

章 4. 以 Piranha Configuration Tool 配置 Load Balancer 外掛程式

Piranha Configuration Tool 提供了為 Load Balancer 外掛程式建立必要配置檔案的結構方案 — **/etc/sysconfig/ha/lvs.cf**。此章節描述了 **Piranha Configuration Tool** 的基本作業，以及當配置完成後如何啓動 Load Balancer 外掛程式。

重要

Load Balancer 外掛程式的配置檔案遵循了嚴格的格式規則。使用 **Piranha Configuration Tool** 是避免 **lvs.cf** 中發生語法錯誤的最佳方式，並可避免軟體上的錯誤發生。

4.1. 必要的軟體

piranha-gui 服務必須在主要的 LVS 路由器上執行，才可使用 **Piranha Configuration Tool**。若要配置 Load Balancer 外掛程式，您至少需要一個純文字的網站瀏覽器，例如 **links**。若您要從另一部機器存取 LVS 路由器，您還必須要以 root 使用者身份透過 **ssh** 連線連至主 LVS 路由器。

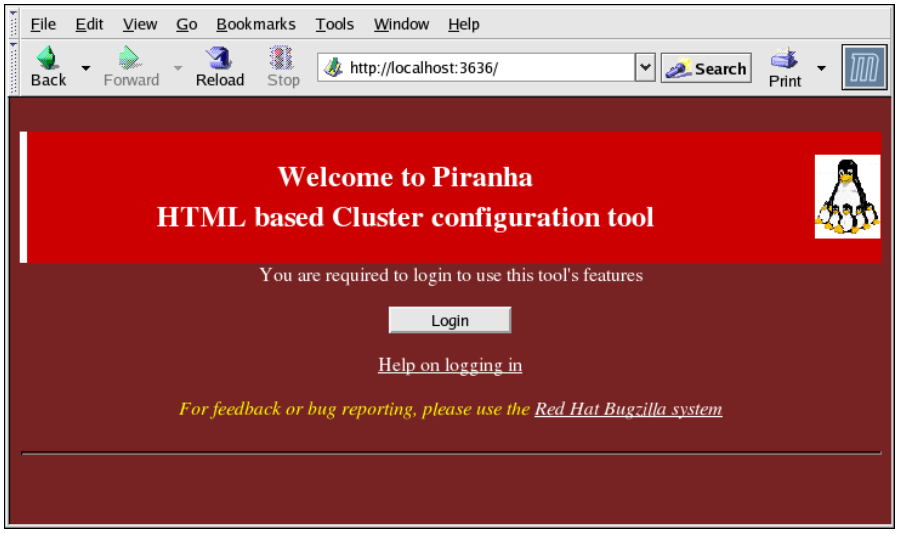
當在配置主 LVS 路由器時，建議您在一個終端機視窗中保留目前的 **ssh** 連線。此連線提供了一項重新啓動 **pulse** 與其它服務、配置網路封包過濾器，以及在進行疑難排解時監控 **/var/log/messages** 的安全方式。

以下四個部份將簡述 **Piranha Configuration Tool** 的各個配置頁，並提供了如何使用它來設定 Load Balancer 外掛程式的指示。

4.2. 登入 Piranha Configuration Tool

當配置 Load Balancer 外掛程式時，您應總是先透過 **Piranha Configuration Tool** 配置主路由器。若要這麼作，請確認 **piranha-gui** 服務有在運作，並且您已設置了一組管理員密碼，如 節 2.2, “為 Piranha Configuration Tool 設定密碼” 中所描述。

若您要由本機存取機器，您可在一個網站瀏覽器中開啓 **http://localhost:3636**，以存取 **Piranha Configuration Tool**。此外，您亦可輸入伺服器的主機名稱或真實 IP 位址，然後加上 **:3636**。當瀏覽器連上了之後，您將會看見顯示在 圖形 4.1, “歡迎面板” 中的畫面。



圖形 4.1. 歡迎面板

請點選「**登入**」按鈕，在「**使用者名稱**」欄位中輸入 **piranha**，並在「**密碼**」欄位中輸入您所建立的管理密碼。

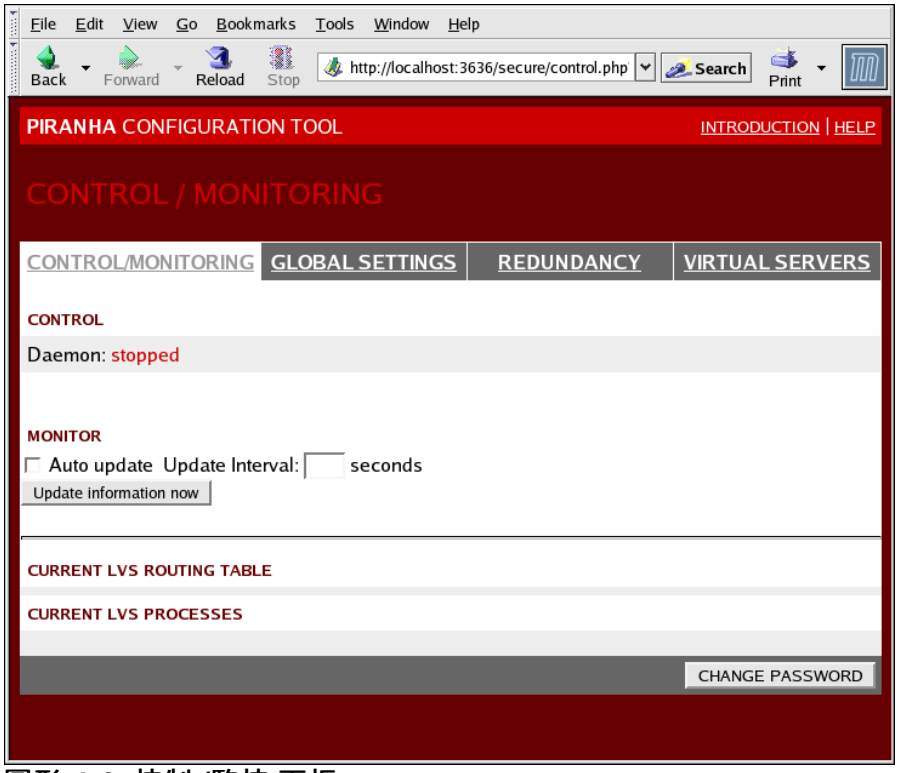
Piranha Configuration Tool 包含了四個主要的畫面（或*面板*）。此外，「**虛擬伺服器**」面板包含了四個*子畫面*。「**控制/監控**」面板會是在登錄畫面後出現的第一個面板。

4.3. 控制/監控

「**控制/監控**」面板會顯示受限制的 Load Balancer 外掛程式 runtime 狀態。它會顯示 **pulse** daemon 的狀態、LVS 的路由表、以及 LVS 衍生的 **nanny** 程序。

請注意

「**目前的 LVS 路由表**」和「**目前的 LVS 程序**」欄位將保留為空白，直到您實際地啓用 Load Balancer 外掛程式，如 節 4.8, “啓用 Load Balancer 外掛程式” 中所示。



圖形 4.2. 控制/監控 面板

自動更新

此頁上的狀態畫面可根據使用者所配置的頻率自動更新。若要啓用這項功能，請點選「**自動更新**」核取方塊，並在「**更新頻率，以秒鐘為單位**」的文字方塊中設置欲使用的更新頻率（預設值為 10 秒鐘）。

我們不建議您將此間隔設定設為低於十秒。如果低於十秒的話，頁面更新的頻率會非常頻繁，那麼要再重新設定「**自動更新**」的間隔便會有困難。如果您遇上了這項問題，只要按下另一個面板，再回到「**控制/監控**」即可。

並非所有瀏覽器皆支援「**自動更新**」功能，例如 **Mozilla**。

立即更新資訊

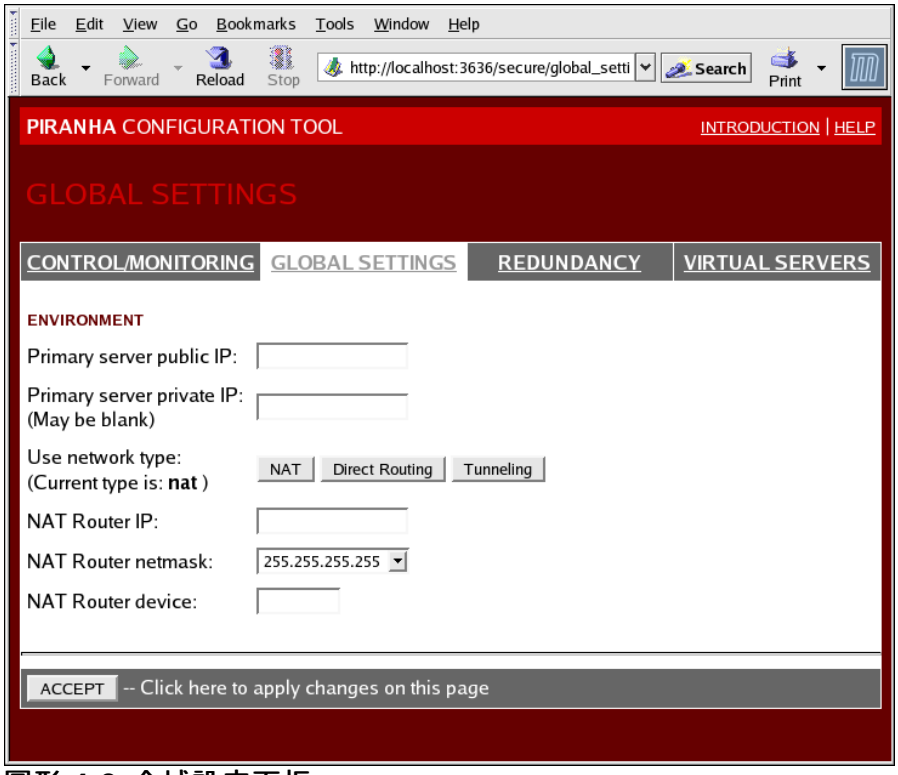
您可藉由點選此按鈕，以手動式更新狀態資訊。

更改密碼

按下此按鈕將會出現一個包含了協助資訊的畫面，以幫助您更改 **Piranha Configuration Tool** 的管理者密碼。

4.4. 全域設定

「**全域設定**」是 LVS 管理者定義主 LVS 路由器的公開與私有網路介面卡的詳情之處。



圖形 4.3. 全域設定面板

此面板的上半部可設定主 LVS 路由器的公開與私有網路介面卡。這些都是已在 節 3.1.1, “配置搭配了 NAT 的 Load Balancer 外掛程式的網路介面卡” 中配置的介面卡。

主伺服器的公用 IP

在此欄位中，請輸入主 LVS 節點的公開、可路由的真實 IP 位址。

主伺服器的私有 IP

請為主 LVS 節點上的另一個網路介面卡輸入一組真實的 IP 位址。這組位址只會被使用來作為備份路由器的額外 heartbeat 頻道，並且無須與指定於 節 3.1.1, “配置搭配了 NAT 的 Load Balancer 外掛程式的網路介面卡” 中的真實私密 IP 位址互相關聯。您可將此欄位保留空白，不過這麼作即代表備份 LVS 路由器沒有可用的額外 heartbeat 頻道，並且將會產生單點失效（single point of

failure）。

請注意

進行「**直接路由**」不需使用私密 IP 位址，因為所有的真實伺服器與 LVS director 皆共享相同的虛擬 IP 位址，並且應該擁有相同的 IP 路由配置。

請注意

主 LVS 路由器的私密 IP 可配置於任何接受 TCP/IP 的介面上，無論是乙太網路卡或是序列埠。

使用網路類型

點選「**NAT**」按鈕來選擇 NAT 路由。

按下「**直接路由**」按鈕來選擇直接路由。

接下來的三個欄位專注於連結私密網路與真實伺服器的 NAT 路由器的虛擬網路介面卡。這些欄位「*不適用*」於直接路由的網路類型。

NAT 路由器 IP

請在此文字欄位中輸入私密浮動 IP。此浮動 IP 應使用來作為真實伺服器的閘道器。

NAT 路由器子網路遮罩

如果 NAT 路由器的浮動 IP 需要特定的子網路遮罩，請從下拉式視窗中選擇。

NAT 路由器裝置

使用此文字欄位來定義網路介面卡的裝置名稱，以取得浮動 IP 位址，例如 **eth1:1**。

請注意

您應將 NAT 浮動 IP 位址與連至了私密網路的乙太網路卡相聯。在此範例中，私密網路位於 **eth1** 卡上，因此 **eth1:1** 為浮動 IP 位址。

警告

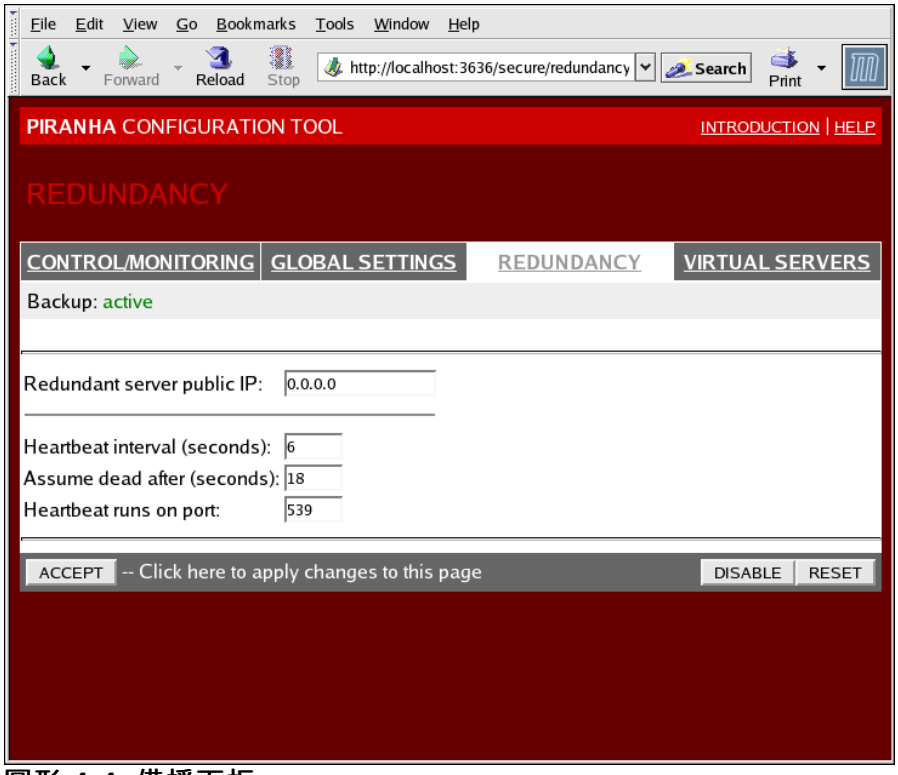
在完成了此階段後，請點選「**確定**」按鈕以確保當您選擇了新面板時，不會遺失任何變更。

4.5. 備援

「**備援**」面板能讓您配置備用 LVS 路由器節點，並設置不同的 heartbeat 監控選項。

請注意

您第一次進入此畫面時，它會顯示一個「互動式」的「**備份**」狀態以及一個「**啓用**」按鈕。若要配置備份 LVS 路由器，請點選「**啓用**」按鈕，以使畫面與 圖形 4.4, “備援面板” 相符。



圖形 4.4. 備援面板

備援伺服器的公共 IP

輸入備用 LVS 路由器節點的公共真實 IP 位址。

備援伺服器的私有 IP

在此文字欄位中輸入備用節點的私有真實 IP 位址。

若您沒有看到一個名為「**備援伺服器私密 IP**」的欄位，請退回至「**全域設定**」面板，然後輸入一組「**主伺服器私密 IP**」位址然後按下「**確定**」。

面板的其它部份主要用來配置 heartbeat 頻道，以供備用節點監控主節點是否失效。

Heartbeat 間隔（秒）

此欄位可用來設定 heartbeat 間隔之間的秒數 — 備份節點檢查主 LVS 節點運作作狀態的間隔。

再此間隔（秒）後，判斷伺服器已無法運作

如果主 LVS 節點在此秒數後沒有回應，那麼備用 LVS 路由器節點便會開始啓動備援。

Heartbeat 的連接埠

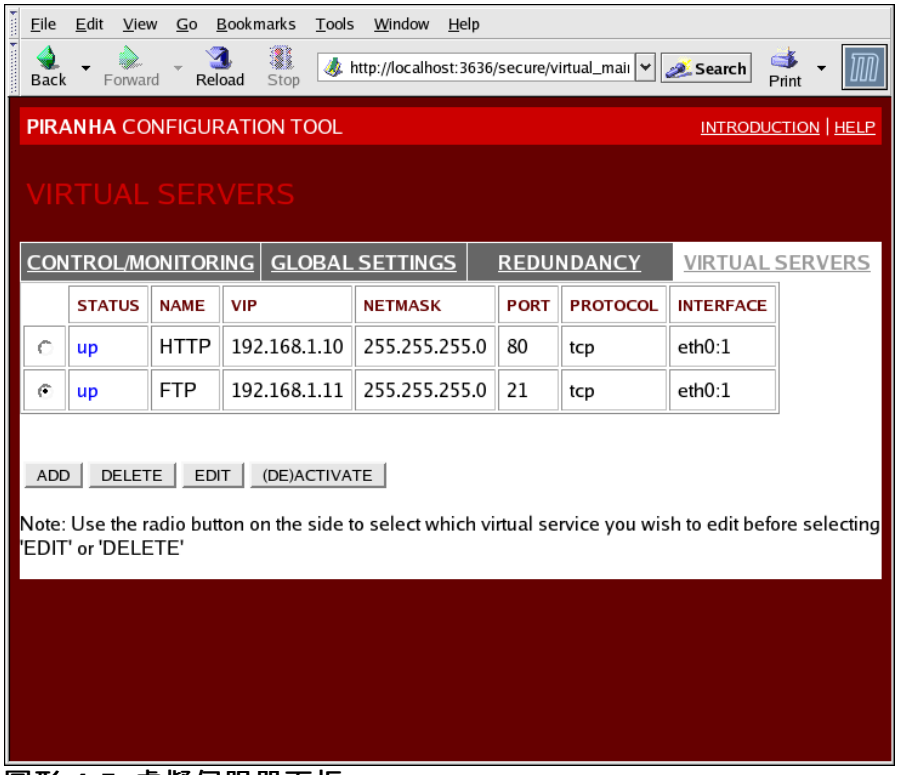
此欄位可設置 heartbeat 與主 LVS 節點通訊的連接埠。如果此欄位空白的話，預設值便會設為 539。

警告

在此面板中進行了變更後，請記得按下「**確定**」按鈕，以確保您在選擇新面板時不會失去任何變更。

4.6. 虛擬伺服器

「**虛擬伺服器**」面板顯示了目前所有已定義的虛擬伺服器。每個表格顯示的是虛擬伺服器的狀態、伺服器名稱、伺服器的虛擬 IP、虛擬 IP 的子網路遮罩、服務所使用的連接埠、通訊協定、以及虛擬裝置的介面卡。



圖形 4.5. 虛擬伺服器面板

「**虛擬伺服器**」面板所顯示的每台伺服器皆可在以下畫面或*子畫面*中進行配置。

要新增服務，請按下「**新增**」按鈕。若要移除一項服務，請勾選虛擬伺服器，然後按下「**刪除**」按鈕。

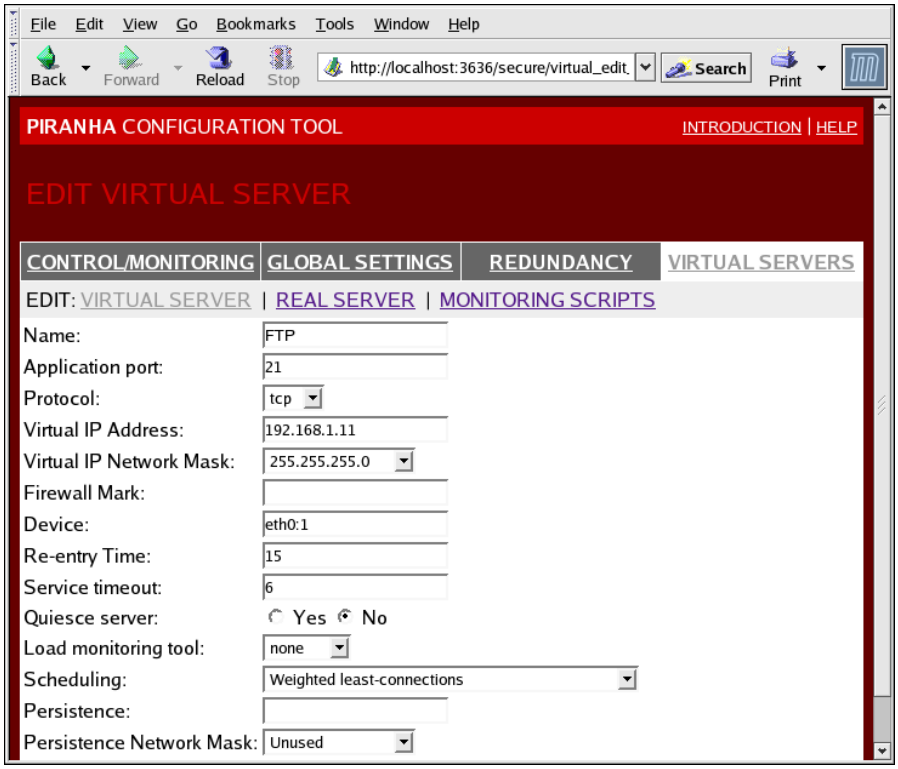
要啓用或停用表格中的虛擬伺服器，點選它然後按下「**停用/啓用**」按鈕。

新增了虛擬伺服器之後，若要進行配置，您可按下左方的勾選按鈕，然後按下「**編輯**」按鈕，以顯示「**虛擬伺服器**」子畫面。

4.6.1. 虛擬伺服器子畫面

「**虛擬伺服器**」子畫面（如 圖形 4.6, “「虛擬伺服器」子畫面” 中所示）能讓您配置各別的虛擬伺服器。特

別與這台虛擬伺服器關聯的子畫面連結，位於網頁上方。但在配置任何與此虛擬伺服器關聯的子畫面之前，請先完成此畫面，然後按下「**確定**」按鈕。



圖形 4.6.「虛擬伺服器」子畫面

名稱

請輸入一組描述此虛擬伺服器的名稱。這組名稱「*並非*」機器的主機名稱，請盡可能地詳盡描述這台機器，使其易於辨識。您亦可參照此虛擬伺服器所使用的通訊協定，例如 HTTP。

應用程式連接埠

輸入服務應用程式將監聽的連接埠號。因為此範例專注於 HTTP 服務，因此使用了連接埠號 80。

通訊協定

在下拉式選單中選取 UDP 或是 TCP。網站伺服器一般會透過 TCP 協定來進行通訊，因此上述範例中選擇了它。

虛擬 IP 位址

在此文字欄位中輸入虛擬伺服器的浮動 IP 位址。

虛擬 IP 的子網路遮罩

以下拉式選單設置此虛擬伺服器的子網路遮罩。

Firewall Mark

除非您要為各別、不過卻相聯的協定綁定多連接埠的協定，或是建立一個多連接埠的虛擬伺服器，否則「*請勿*」在此欄位中輸入 firewall mark 的數值。在此範例中，上述的虛擬伺服器含有一個「**Firewall Mark**」80，因為我們要使用數值為 80 的 firewall mark 來將連線綁定至連接埠 80 上的 HTTP，以及連接埠 443 上的 HTTPS。當與 persistence 結合之後，此技巧可確保存取安全

與非安全網頁的使用者，皆會被路由轉送至相同的真實伺服器上，並保留狀態。

警告

在此欄位中輸入 firewall mark 能讓 IPVS 辨識使用此 firewall mark 的封包會被視為相同的，不過您必須在 **Piranha Configuration Tool** 之外進行額外的配置，以實際地指定 firewall mark。欲取得如何建立多連接埠服務上的相關資訊，請參閱 節 3.4, “多連接埠服務與 Load Balancer 外掛程式”，若要建立高可用性的 FTP 虛擬伺服器，請參閱 節 3.5, “配置 FTP”。

裝置

請輸入您希望定義了「**虛擬 IP 位址**」的浮動 IP 位址所要綁定的網路裝置之名稱。

您應使公用浮動 IP 位址與連至了公用網路的乙太網路介面卡相聯。在此範例中，公用網路位於 **eth0** 介面卡上，因此 **eth0:1** 應作為裝置名稱。

再次進入的時間

請輸入一組數值，以設置啓用中的 LVS 路由器在失效後，嘗試將真實伺服器停用之前的所需時間長度（秒）。

服務逾時

請輸入一組數值，以定義當真實伺服器被視為無法運作，並由伺服器群中移除前的所需時間長度（秒）。

Quiesce 伺服器

當勾選了「**Queisce 伺服器**」（靜默伺服器）後，當新的真實伺服器節點上線時，最少連線表會更新為零，如此一來啓用中的 LVS 路由器便會轉送請求，就如所有真實伺服器皆為新加入叢集一般。此選項會避免新伺服器在進入叢集時，因為高連線數而負載過重。

負載監控工具

LVS 路由器可以使用 **rup** 或 **ruptime**，以監控多個真實伺服器的負載。如果您從下拉式選單選擇 **rup**，每一台伺服器皆必須執行 **rstatd** 服務。如果您選擇了 **ruptime**，每台伺服器皆必須執行 **rwhod** 服務。

警告

負載監控與負載平衡「*不同*」，並且當與加權排程演算法合併時，可能會產生難以預測的排程事件。此外，若您使用負載監控，真實伺服器皆必須為 Linux 機器。

排程

請從下拉式選單中選擇您偏好使用的排程演算法。預設值為**加權最少連線**。欲取得更多有關於排程演算法上的相關資訊，請參閱 節 1.3.1, “排程演算法”。

Persistence

如果系統管理員需要在用戶端連上虛擬伺服器時使用 persistence 連線，請在此文字欄位中指定連線逾時之前，所經過的閒置秒數。

重要

若您在以上的「**Firewall Mark**」欄位中輸入了一組數值，您也應為 persistence 輸入一組數值。此外，請確認若您一起同時使用了 firewall mark 和 persistence，那麼 persistence 的數量會與各個含有 firewall mark 的虛擬伺服器相同。欲取得更多有關於 persistence 和 firewall mark 上的相關資訊，請參閱 節 1.5, “Persistence 與 Firewall Mark”。

Persistence 子網路遮罩

若要限制 persistence 使用在特定的子網路上，請從下拉式選單中選擇適當的子網路遮罩。

請注意

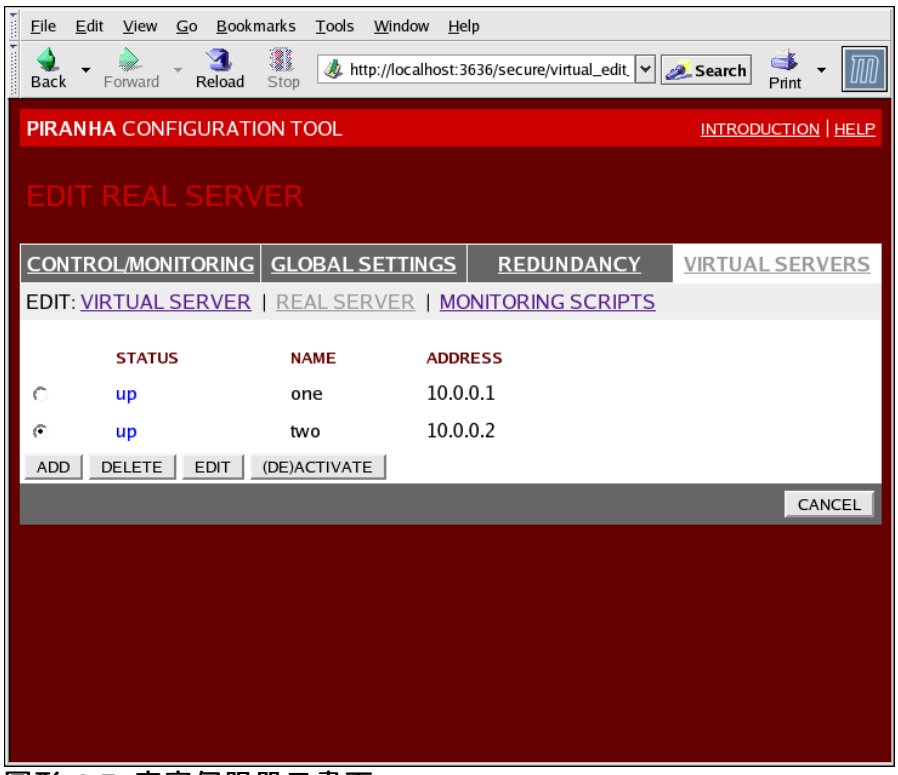
在 firewall mark 出現之前，受限於子網路的 persistence 原本是項未成熟的連線綁定方式。不過，現在最佳的方式就是相互使用 persistence 與 firewall mark 以達到相同的效果。

警告

在此面板中進行了變更後，請記得按下「**確定**」按鈕，以確保您不會在選擇新面板時，失去您所作的變更。

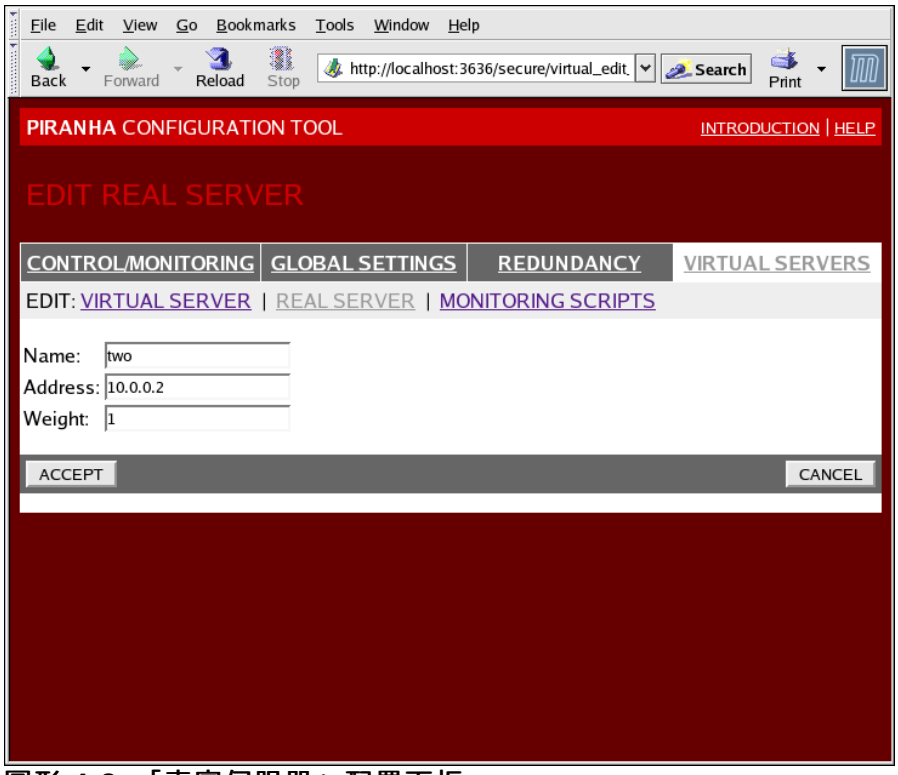
4.6.2. 真實伺服器子畫面

按下面板上方的「**真實伺服器**」子畫面連結將會顯示「**編輯真實伺服器**」的子畫面。它會顯示某項特定虛擬服務的實體伺服器主機狀態。



圖形 4.7. 真實伺服器子畫面

請按下「**新增**」按鈕以新增伺服器。若要刪除既有的伺服器，請點選相對應的伺服器，然後按下「**刪除**」按鈕。按下「**編輯**」按鈕將會載入「**編輯真實伺服器**」面板，如 圖形 4.8, “「真實伺服器」配置面板” 中所示。



圖形 4.8. 「真實伺服器」配置面板

此面板包含了三個可輸入欄位：

名稱

真實伺服器的描述性名稱。

請注意

這個名稱「*不是*」電腦的主機名稱，因此更詳盡的說明，可讓它易於被辨識。

「位址」

真實伺服器的 IP 位址。由於已為相關的虛擬伺服器指定了監聽連接埠，因此請勿在此加入連接埠號。

「權重」

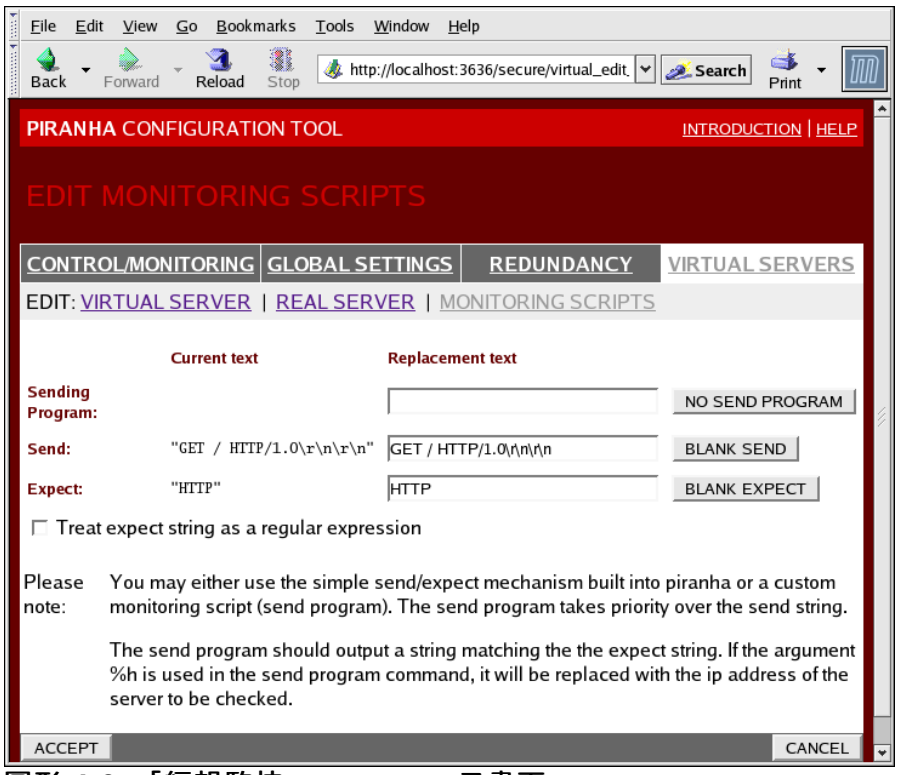
一個表示了此主機相對於叢集中其它主機的能力的數值。該數值為任意的，不過請將它視為與叢集中其它真實伺服器的比例。欲取得更多有關於伺服器權重上的相關資訊，請參閱 節 1.3.2, “伺服器加權與排程”。

警告

在此面板中進行了變更後，請記得按下「**確定**」按鈕，以確保您在選擇新面板時不會失去任何變更。

4.6.3. 編輯監控 SCRIPTS 子畫面

按下網頁上方的「**監控 SCRIPTS**」連結。「**編輯監控 SCRIPTS**」子畫面能讓管理者指定發出（或預期收到）字串序列，以檢查虛擬伺服器的服務在每個真實伺服器上皆執行無誤。此部份也是管理者能夠自訂 script，以檢查需要動態變動資料的服務的地方。



圖形 4.9.「編輯監控 SCRIPTS」子畫面

「傳送程式」

若要進一步驗證服務，您可使用此欄位來指定檢查服務的 script 之路徑。這項功能對於需要動態變更資料的服務（例如 HTTPS 或 SSL）來說，特別有幫助。

若要使用這項服務，您必須編寫一個 script，該 script 必須要能夠回傳文字回應，將其設定為可執行，同時在「**傳送程式**」欄位中輸入路徑。

請注意

若要確保真實伺服器群中的所有伺服器皆有被檢查到，請在「**傳送程式**」欄位中的 script 路徑後附加特殊的標記符號 **%h**。當 script 被 **nanny** daemon 調用時，此標記符號便會被取代為各個真實伺服器的 IP 位址。

當建置一個外部服務檢查 script 時，您可使用下列 script 範本：

```
#!/bin/sh

TEST=`dig -t soa example.com @$1 | grep -c dns.example.com

if [ $TEST != "1" ]; then
 echo "OK
else
 echo "FAIL"
fi
```

請注意

如果您在「**傳送程式**」欄位中輸入了一個外部程式，那麼「**傳送**」欄位便會被忽略。

「傳送」

在此欄位中輸入一組透過 **nanny** daemon，以發送給所有真實伺服器的字串。就預設值， HTTP 的傳送欄位已完成。您可根據需求修改這個值。若此欄位保留為空白的話，**nanny** daemon 便會嘗試開啓連接埠，並且若成功開啓的話，便會假定該服務正在運作中。

這欄位只允許一個發送，也只能包含可列印的 ASCII 字元，以及以下逸出字元：

- \n 是換行。
- \r 是回到第一個字元。
- \t 相當於按下「Tab」鍵。
- \ 是逸出下一個緊跟著的字元。

預期

輸入伺服器應該會傳回的文字（如果伺服器正常運作的話）。如果您自行撰寫了發送程式，請輸入您自訂的成功回應。

請注意

要決定某個服務要發送什麼內容，您可以開啓一組 **telnet** 連線到真實伺服器的連接埠上，然後看看會傳回什麼。舉例來說，FTP 會在連線時傳回 220，如此一來就可以在「**發送**」欄位裡輸入 **quit**，「**預期**」欄位裡輸入 **220**。

警告

在此面板中進行了變更後，請記得按下「**確定**」按鈕，以確保您在選擇新面板時不會失去任何變更。

一旦您使用 **Piranha Configuration Tool** 配置了虛擬伺服器，您必須將特定的配置檔案複製到備用 LVS 路由器上。詳情請參閱 節 4.7, “同步配置檔案”。

4.7. 同步配置檔案

在配置了主 LVS 路由器之後，在您啓動 Load Balancer 外掛程式之前，有幾個配置檔案必須複製到備用的 LVS 路由器上。

這些欄位包括：

- **/etc/sysconfig/ha/lvs.cf** — LVS 路由器的配置檔案。
- **/etc/sysctl** — 開啓 kernel 封包轉送功能的配置檔案。
- **/etc/sysconfig/iptables** — 如果您使用了 firewall mark，就必須根據所使用的網路封包篩選成是，同步這些檔案的其中之一。

重要

當您使用 **Piranha Configuration Tool** 配置 Load Balancer 外掛程式時，**/etc/sysctl.conf** 與 **/etc/sysconfig/iptables** 檔案「*不會*」改變。

4.7.1. 同步 lvs.cf

當 LVS 的配置檔案 **/etc/sysconfig/ha/lvs.cf** 建立或更新時，您必須將其複製到備用 LVS 路由器上。

警告

啓用中的 LVS 與備用 LVS 路由器必須擁有一致的 **lvs.cf** 檔案。兩者的 LVS 配置檔案若不一致，會無法提供備援功能。

同步的最好方法，是使用 **scp** 指令。

重要

要使用 **scp** 指令，備用伺服器上的 **sshd** 服務必須開啓。欲知如何再 LVS 路由器上正確配置所需的服務，請參閱 節 2.1, “在 LVS 路由器上進行服務配置”。

請在 LVS 路由器上以 root 身份執行以下指令，好將 **lvs.cf** 檔案同步到路由器節點上：

```
scp /etc/sysconfig/ha/lvs.cf n.n.n.n:/etc/sysconfig/ha/lvs.cf
```

在這指令裡，請以備用 LVS 路由器的真實 IP 位址取代 ***n.n.n.n***。

4.7.2. 同步 sysctl

在大部分情形下，**sysctl** 檔案只會被更改一次。這檔案會在開機時讀取，告訴 kernel 開啓封包轉送功能。

重要

如果您不確定 kernel 是否啓用了封包轉送功能，請參閱 節 2.5, “啓動封包轉送（Packet Forwarding）” 以了解檢查並視需要啓用此關鍵功能的相關資訊。

4.7.3. 同步網路封包篩選規則

如果您使用了 **iptables**，請將配置檔案同步到備用的 LVS 路由器。

如果您改變任何網路封包篩選成是，請從主 LVS 路由器上，以 root 身份輸入以下指令：

```
scp /etc/sysconfig/iptables n.n.n.n:/etc/sysconfig/
```

在這指令裡，請以備用 LVS 路由器的真實 IP 位址取代 ***n.n.n.n***。

接下來，請在備用路由器上開啓 **ssh** session，或使用 root 登入並執行以下指令：

```
/sbin/service iptables restart
```

一旦您將這些檔案複製到備用路由器，並啓動相關服務之後（詳情請參閱 節 2.1, “在 LVS 路由器上進行服務配置”），您便可啓動 Load Balancer 外掛程式。

4.8. 啓用 Load Balancer 外掛程式

若要啓動 Load Balancer 外掛程式，最好開啓兩組以 root 身份登入的終端機視窗，或兩組以 root 身份、藉由 **ssh** 登入主 LVS 路由器的視窗。

在其中一個終端機視窗中，利用以下指令觀察 kernel 的日誌紀錄：

```
tail -f /var/log/messages
```

然後在另一個終端機中，輸入以下指令以啓動 Load Balancer 外掛程式：

```
/sbin/service pulse start
```

請看著 **pulse** 服務啓動時的進度，參照 kernel 的日誌訊息。當您看到以下訊息時，表示 pulse daemon 已經正確啓動：

```
gratuitous lvs arps finished
```

要停止觀察 **/var/log/messages**，請輸入 **Ctrl**+**c**。

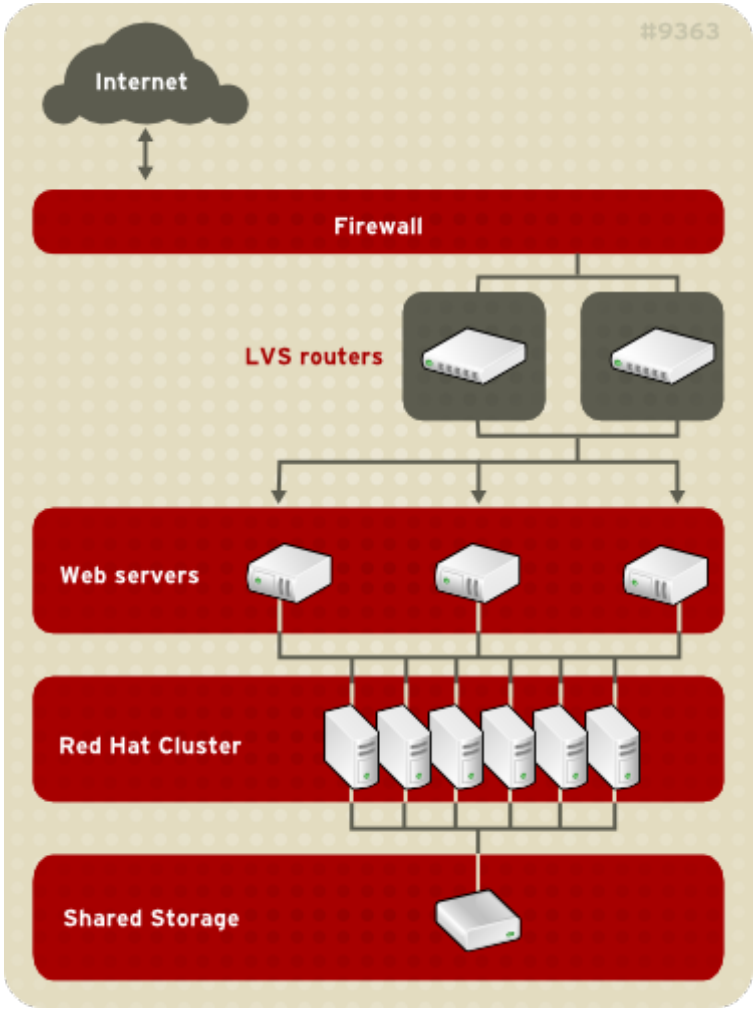
從現在開始，主 LVS 路由器也是啓用中的 LVS 路由器。當您在此向 Load Balancer 外掛程式提出請求時，請在讓 Load Balancer 外掛程式提供服務之前，啓動備用 LVS 路由器。若要啓動備用路由器，只要在備用 LVS 路由器節點上重複以上動作即可。

完成此最後步驟後，Load Balancer 外掛程式便會啓動、執行。

搭配使用 Load Balancer 外掛程式與 High Availability 外掛程式

您可以搭配使用 Load Balancer 外掛程式與 High Availability 外掛程式，以建置高可用性的電子商務網站，以提供負載平衡、資料完整性、以及應用程式的可用性。

圖形 A.1, “搭配使用 High Availability 外掛程式與 Load Balancer 外掛程式” 中的配置代表一個電子商務網站，透過網址在網路上進行銷售。用戶的請求會透過此網址傳到防火牆，送達啓用中的 LVS 負載平衡路由器，然後轉送到其中一台網站伺服器上。High Availability 外掛程式節點會為網站伺服器提供動態資料，並將這些資料回傳給用戶端。



圖形 A.1. 搭配使用 High Availability 外掛程式與 Load Balancer 外掛程式

利用 Load Balancer 外掛程式提供動態網頁內容需要三階段式的配置（如 圖形 A.1, “搭配使用 High Availability 外掛程式與 Load Balancer 外掛程式” 中所示）。Load Balancer 外掛程式與 High Availability 外掛程式的結合，提供了高整合性、無單點失效問題的電子商務網站配置。High Availability 外掛程式可執行高可用性的單一資料庫或多個資料庫，並讓網站伺服器透過網路存取。

要提供動態內容需要三階段式的配置。若網站伺服器只提供動態網頁內容（小量、不常變動的資料），那麼兩階段的 Load Balancer 外掛程式配置非常適用；但兩階段的配置不適用於提供動態內容的網站伺服器。動態內容可能包括產品的存貨狀態、購貨訂單、或客戶資料，所有網站伺服器上的資料都必須一致，確保客戶所獲得的是最新、最正確的資訊。

每一階段皆提供以下功能：

- 第一階段 — LVS 路由器藉由負載平衡功能，將網站請求分配出去。

- 第二階段 — 多台網站伺服器，以回應這些請求。
- 第三階段 — High Availability 外掛程式，為網站伺服器提供資料。

在 Load Balancer 外掛程式配置（如 圖形 A.1, “搭配使用 High Availability 外掛程式與 Load Balancer 外掛程式”）中，用戶端系統會在網際網路上發出請求。基於安全理由，這些請求會透過防火牆進入網站，防火牆可以是 Linux 系統，也可以是專門的防火牆裝置。為了提供冗餘功能，您可用備援配置來設定防火牆裝置。防火牆之後是提供覆載平衡的 LVS 路由器，並可配置為主動/備援模式。啓用中的負載平衡路由器會將請求轉送至網站伺服器群上。

每台網站伺服器皆可獨立處理 HTTP 請求，並將結果傳回給用戶端。Load Balancer 外掛程式能讓使用者擴充網站的處理能力，方法是在 LVS 路由器的後方新增網站伺服器。LVS 路由器會在多台網站伺服器之間，進行負載平行。除此之外，若有台網站伺服器當機，該伺服器可被移除；而 Load Balancer 外掛程式會繼續在剩下的網站伺服器群之間進行負載平衡。

修訂紀錄

修訂 6-14.400	**2013-10-31**	**Rüdiger Landmann**

Rebuild with publican 4.0.0

修訂 6-14	**2012-07-18**	**Anthony Towns**

Rebuild for Publican 3.0

修訂 1.0-0	**Wed Nov 10 2010**	**Paul Kennedy**

Red Hat Enterprise Linux 6 的初始版本。

索引

符號

N

P

S