Red Hat Enterprise Linux（RHEL） 6

High Availability 外掛程式總覽

RHEL 的 High Availability 外掛程式總覽

版 6

Last Updated: 2017-10-29

Red Hat Enterprise Linux（RHEL） 6 High Availability 外掛程式總覽

RHEL 的 High Availability 外掛程式總覽
版 6

法律聲明

摘要

《High Availability 外掛程式總覽》提供了有關於 Red Hat Enterprise Linux 6 上的 High Availability 外掛程式之總覽。

內容目錄

簡介

本文件提供了有關於 Red Hat Enterprise Linux 6 上的 High Availability 外掛程式之基礎總覽。

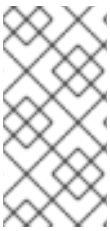
儘管本文件中的資訊僅為總覽，讀者應該要擁有進階的 Red Hat Enterprise Linux 操作知識，並理解伺服器運算的概念，才能充分吸收本文件中的資訊。

欲取得更多有關於 Red Hat Enterprise Linux 使用上的相關資訊，請參閱下列資源：

- 《*Red Hat Enterprise Linux 安裝指南*》— 提供了有關於 Red Hat Enterprise Linux 6 安裝上的相關資訊。
- 《*Red Hat Enterprise Linux 建置指南*》— 提供了有關於 Red Hat Enterprise Linux 6 的建置、配置和管理上的相關資訊。

欲取得更多有關於此以及 Red Hat Enterprise Linux 6 相關產品上的資訊，請參閱以下資源：

- 《*配置和管理 High Availability 外掛程式*》提供了有關於 RHEL 6 上的 High Availability 外掛程式（亦稱為 Red Hat Cluster）的配置與管理資訊。
- 《*邏輯卷冊管理程式管理*》— 提供了邏輯卷冊管理程式（LVM，Logical Volume Manager）的相關資訊，包括在叢集環境中執行 LVM 的相關說明。
- 《*全域檔案系統 2：配置與管理*》— 提供了有關於安裝、配置和維護 Red Hat GFS2（Red Hat 全域檔案系統 2）上的相關資訊，而這項資訊包含在 Resilient Storage 外掛程式中。
- 《*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。
- 《*負載平衡管理*》— 提供了透過 Red Hat Load Balancer 外掛程式（先前名為 Linux 虛擬伺服器〔LVS〕）配置高效能系統與服務的相關資訊。
- 《*發行公告*》— 提供了有關於 Red Hat 最新產品的相關資訊。



注意

欲知有關於透過 High Availability 外掛和 Red Hat Global File System 2（GFS2）來建置和升級 RHEL 叢集上的最佳方式，請參閱 Red Hat 客戶入口網站上的「Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Best Practices」文件（位於 . https://access.redhat.com/kb/docs/DOC-40821）。

本文件以及其它 Red Hat 文件皆擁有 HTML、PDF 以及 RPM 版本於 Red Hat Enterprise Linux 文件光碟，以及 http://access.redhat.com/documentation/docs 網站上。

1. 我們需要您的意見！

若您在本指南中發現了任何錯誤，或是您有任何改善本指南的方式，我們很樂意聽取您的意見！請於 Bugzilla（http://bugzilla.redhat.com/）針對於產品 **Red Hat Enterprise Linux 6**、*doc-High_Availability_Add-On_Overview* 元件以及版本 **6.6** 提交一份報告。

若您希望提供改善文件的建議，請盡可能地詳細描述。若您發現一項錯誤，請包含部份號碼以及其附近的部份文字，如此一來我們便能輕易地找到這項錯誤。

章 1. HIGH AVAILABILITY 外掛程式總覽

High Availability 外掛程式是個為重要生產服務提供高可靠性、延展性和可用性的叢集系統。下列部分提供了有關於 High Availability 外掛程式功能及元件的基本詳述。

1.1. 叢集基礎

叢集代表二或更多台電腦（亦稱為*節點*或是*成員*）合併運作以執行一項任務。叢集類型主要分為四種：

- 儲存裝置（Storage）
- 高可用性（High Availability）
- 負載平衡（Load Balancing）
- 高效能（High Performance）

儲存裝置叢集會在叢集中的伺服器之間，提供一致性的檔案系統映像，以讓伺服器同時讀取和寫入單一共享檔案系統。儲存裝置叢集藉由將應用程式的安裝及升級限制在一個檔案系統，以簡化儲存裝置上的管理。此外，當使用一個叢集全域的檔案系統時，儲存裝置叢集可省略複製應用程式資料，並簡化備份和災害復原。High Availability 外掛程式提供了儲存裝置叢集，並結合了 Red Hat GFS2（Resilient Storage 外掛程式的一部分）。

High availability 叢集藉由除去單一失敗點並在節點失效時，將服務由一個叢集節點容錯移轉至另一節點上，以提供高可用性的服務。一般來講，high availability 叢集中的服務會讀取和寫入資料（透過讀寫掛載的檔案系統）。因此，一個 high availability 叢集必須能在一個叢集節點由另一叢集節點接收服務控制權時，保留資料的完整性。一個 high availability 叢集中的節點失效，將不會被叢集外的客戶端看見。（high availability 叢集有時亦稱為容錯移轉叢集。）High Availability 外掛程式透過了其 High Availability Service Management 元件 **`rgmanager`**，來提供了高可用性的叢集處理。

Load-balancing 叢集會將網路服務請求發送至數個叢集節點上，以平衡叢集節點之間的請求負載。負載平衡機制能提供高效益的延展，因為您能夠根據負載需求來比對節點的數量。若一個 load-balancing 叢集中有個節點失效，load-balancing 軟體將會偵測到此失效情況，並將請求重新導向至其它叢集節點上。Load-balancing 叢集中的節點失效，不會被叢集外部的客戶端看見。負載平衡可藉由 Load Balancer 外掛程式提供。

High-performance 叢集乃用來進行同時運算的的叢集節點。High-performance 叢集能讓應用程式平行運作，以增強應用程式的效能。（High performance 叢集亦稱為「運算叢集」或「資料格運算」。）

注意

概述於前置文字中的叢集類型反映了基本的配置；您的需求可能需要合併使用描述到的叢集。

此外，Red Hat Enterprise Linux High Availability 外掛程式*僅*支援配置和管理 high availability 伺服器。它*不*支援 high-performance 叢集。

1.2. HIGH AVAILABILITY 外掛程式簡介

High Availability 外掛程式是個整合式軟體元件集，它可藉由各種不同的配置方式來建置，以滿足您對於效能、高可用性、負載平衡、延展性、檔案共享與經濟上的需求。

High Availability 外掛程式包含了下列主要元件：

- Cluster infrastructure — 提供了節點作為叢集互相協作的基礎功能：配置檔案管理、成員管理、鎖定管理以及隔離。
- High availability Service Management — 當節點失效時，能將服務由一個叢集節點上，容錯移轉至另一個節點上。
- Cluster administration tools — 用來設定、配置和管理 High Availability 外掛程式的配置和管理工具。此工具可搭配叢集基礎結構（Cluster Infrastructure）元件、high availability 和 Service Management 元件，以及儲存裝置使用。

注意

目前僅完全支援單地區叢集。目前尚未正式支援散佈於多實體位置的叢集。欲取得更多詳情，以及多地區叢集上的相關資訊，請與您的 Red Hat 業務或支援代表進行聯繫。

您可使用下列元件來補充 High Availability 外掛程式：

- Red Hat GFS2（Global File System 2）— Resilient Storage 外掛程式的一部分，它提供了一個叢集檔案系統，以搭配 High Availability 外掛程式使用。GFS2 能在區塊層級中，讓多個節點共享儲存裝置，就如儲存裝置本機連上了各個叢集節點。GFS2 叢集檔案系統需要一個叢集基礎結構。
- Cluster Logical Volume Manager（CLVM）— Resilient Storage 外掛程式的一部分，這提供了叢集儲存裝置的卷冊管理。CLVM 的支援亦需要叢集基礎結構。
- Load Balancer 外掛程式 — 提供 IP 負載平衡的路由軟體。Load Balancer 外掛程式會在一對冗余虛擬伺服器中執行，並將客戶端請求平均發送至虛擬伺服器後的真實伺服器上。

1.3. 叢集架構

High Availability 外掛程式的叢集基礎結構，為一組電腦（亦稱為*節點*或是*成員*）提供了基礎功能，以讓它們作為叢集協作。當叢集透過使用叢集基礎結構形成之後，您便可使用其它元件來滿足您的叢集需求（比方說設定一個叢集，以在一個 GFS2 檔案系統上共享檔案，或是設定服務容錯移轉）。叢集基礎結構能進行下列功能：

- 叢集管理
- 鎖定管理
- 隔離
- 叢集配置管理

章 2. 以 CMAN 進行叢集管理

叢集管理功能可管理叢集仲裁與叢集成員。CMAN（cluster manager 的縮寫）會在 Red Hat Enterprise Linux 的 High Availability 外掛程式中進行叢集管理。CMAN 是個分散式的叢集管理程式，並且會在各個叢集節點中執行；叢集管理散佈於叢集中的所有節點上。

CMAN 會藉由監控來自於其它叢集節點的訊息，以追蹤叢集成員。當叢集成員改變時，叢集管理程式會通知其它基礎結構元件，並進行適當的動作。若叢集節點未在特定的時間內傳送訊息的話，叢集管理程式便會將節點由叢集中移除，並與該節點不屬於其成員的其它叢集基礎結構元件進行通訊。其它的叢集基礎結構元件，會在被通知節點已不屬於叢集成員時，決定要進行哪些動作。比方說，「隔離」會將已不再屬於叢集成員的節點離線。

CMAN 會透過監控叢集節點的計數來追蹤叢集仲裁。若超過一半的節點啓用中，叢集便有仲裁。若一半（或更少）的節點未啓用，則叢集便無仲裁，而所有的叢集活動皆會停下。叢集仲裁可避免叢集分裂（split brain）的狀況發生 — 此狀況代表同一叢集分裂並同時執行。此分裂狀況會造成各個分裂的叢集，在不知有另一叢集存在的情況下存取叢集資源，導致於叢集完整性損毀。

2.1. 叢集仲裁

「仲裁（quorum）」乃 CMAN 所使用的投票機制。

叢集若要能夠正確運作，所屬成員必須接受其狀態。一般叢集採用仲裁機制，即代表大部份節點皆運作中、進行有效通訊，並接受啓用中的叢集成員。比方說，在一個擁有十三個節點的叢集中，若要取得仲裁，必須有七個或更多個節點進行通訊。若第七個節點失效，則叢集便會失去仲裁，並無法再運作。

叢集必須維持仲裁以避免發生*分裂（split-brain）*問題。若無仲裁，當在以上提到的十三個節點的叢集發生了通訊錯誤時，可能會造成六個節點在共享的儲存裝置上運作，而另外六個節點也同時在相同位置上獨立運作。因為通訊錯誤，這兩個部分叢集可能會同時覆寫磁碟上的區域，並使檔案系統損毀。若強制實施仲裁規則，則僅有一個部分叢集能夠使用共享儲存裝置，進而保護資料的完整性。

與其將仲裁視為避免分裂狀況發生的機制，不如將它視為一個能決定哪些節點可在叢集中運作的機制。當分裂情況發生時，仲裁機制會避免超過一個叢集群組進行任何動作。

仲裁機制的存在性，是以叢集節點透過乙太網路所進行的訊息通訊來判定的。仲裁亦可選用性地透過乙太網路，藉由一系列的通訊訊息*以及*透過仲裁磁碟來判定。透過乙太網路的仲裁機制，需包含 50% + 1 個額外的節點。當配置一個仲裁磁碟時，仲裁機制會包含使用者指定的條件。

注意

就預設值，各個節點皆會有一個仲裁投票。您可選用性地為各個節點配置超過一個投票。

2.1.1. 仲裁磁碟

仲裁磁碟或分割區，是個專門設定來讓叢集專案元件所使用的部分。其用途將透過以下範例詳細解說。

假設您擁有節點 A 和 B，節點 A 遺漏了數個來自於節點 B 的叢集管理程式「 heartbeat」封包。節點 A 不知它為何並未收到這些封包，而可能性有幾種：節點 B 可能已失效、網路切換器或是 hub 可能已失效，節點 A 的網路控制卡可能已失效，或是節點 B 可能只是因為過於忙碌以致於封包未送出。當您的叢集非常大、您的系統過於忙碌，或是網路阻塞時，便有可能會發生此情況。

節點 A 不知情況為何，並且不知問題出於自身或是節點 B。這對於擁有兩個節點的叢集來說問題特別大，因為這兩個失聯的節點可能會嘗試互相進行隔離。

因此在隔離一個節點之前，儘管我們似乎無法聯繫另一節點，若能有其它方式檢查該節點是否運作中會較佳。此時，仲裁磁碟便提供了此功能。在隔離一個失聯的節點之前，叢集軟體能藉由檢查節點有無寫入資料到仲裁磁碟中，來判定該節點是否依然運作中。

當有兩個節點系統時，仲裁磁碟也會作為一個 tie-breaker 運作。若一個節點能存取仲裁磁碟和網路，即代表兩個投票。

與網路或是仲裁磁碟失聯的節點將會失去一個投票，並且將能安全地被隔離。

有關於配置仲裁磁碟參數的更多相關資訊，位於《*叢集管理*》指南中的 Conga 和 **ccs** 管理的章節中。

2.1.2. Tie-breaker（仲裁器）

Tie-breaker 是個額外的啓發式機制，以在發生了相等分裂的情況下，讓叢集分割區在隔離之前，決定它的狀態是否為「quorate」。典型的 tie-breaker 建構乃是個 IP tie-breaker，有時亦稱為「ping 節點」。

透過這種 tie-breaker，節點不僅會互相監控，同時也會在叢集進行通訊時，監控位於相同路徑上的上游路由器。若這兩個節點失聯，勝出的將會是依然能夠 ping 上游路由器的一方。當然，在「switch-loop」的情況下，兩個節點皆能偵測到上游路由器，不過卻無法偵測到對方；這將會造成「split brain」狀況發生。這也就是為何儘管使用了 tie-breaker，依然還需要確認隔離是否已正確配置。

其它類型的 tie-breaker（包括通常稱為仲裁磁碟的共享分割區）亦提供了額外資訊。clumanager 1.2.x（Red Hat Cluster Suite 3）含有一個磁碟 tie-breaker，以允許在網路失效時繼續運作，只要這兩個節點在共享分割區上還能夠進行通訊即可。

其它還有一些較複雜的 tie-breaker 配置，例如 QDisk（屬於 linux-cluster 的一部分）。QDisk 能接受任意的啓發式機制。這能讓各個節點判定其健康狀態是否適合參與加入叢集。然而，它一般會被使用來作為基本的 IP tie-breaker。欲取得更多資訊，請參閱 qdisk(5) man page。

基於某些原因，CMAN 沒有內部的 tie-breaker。然而，tie-breaker 可藉由使用 API 來實作。此 API 能允許仲裁裝置註冊和更新。比方說，請查看 QDisk 的原始碼。

若您屬於以下情況，您便可能需要 tie-breaker：

- 擁有兩個節點，並且隔離裝置位於與使用來進行叢集通訊的路徑不同的網路路徑上
- 擁有兩個節點，並且隔離乃網狀架構等級 - 特別是用於 SCSI 預留

然而，若您的叢集中配置了正確的網路和隔離配置，tie-breaker 只會增加複雜性，除了用於發生問題的情況下。

章 3. RGMANAGER

RGManager 可管理並為服務、資源群組或是資源樹之類的叢集資源提供容錯移轉功能。這些資源群組乃樹狀結構，並且在各個子目錄樹中皆有父子相依性和繼承關係。

RGManager 的運作方式允許管理員定義、配置和監控叢集服務。當節點失效時，RGManager 會在盡可能不影響服務的情況下，將叢集服務重定位至其它節點上。您亦可將服務限制僅在特定節點上執行，比方說將 **httpd** 限制僅能在單一群組的節點上執行，而 **mysql** 則能被限制僅能在不同的個別節點上執行。

RGManager 若要正常運作，需要各種程序和代理程式共同協作。以下清單概述了這些項目。

- 容錯移轉區域 - RGManager 容錯移轉區域系統的運作方式
- 服務政策 - Rgmanager 的服務啓用和復原政策
- 資源樹 - RGManager 的資源樹運作方式，包括啓用/停用的順序和繼承
- 服務作業特性 - RGManager 的作業運作方式以及其狀態意義
- 虛擬機器特性 - 在 rgmanager 叢集中執行 VM 時所需記得的特殊事項
- 資源動作 - RGManager 所使用的代理程式動作，以及如何透過 **cluster.conf** 檔案自訂其特性。
- 事件指令碼處理 - 若 rgmanager 的容錯移轉和復原政策不適用於您的環境，您可藉由使用此指令碼處理子系統來自訂您自己的政策。

3.1. 容錯移轉區域

容錯移轉區域是個服務可綁定至、按照順序的成員子集。儘管容錯移轉區域對於叢集自訂來說相當有幫助，不過若要進行作業，容錯移轉功能並非是必要的。

以下是一列用來管理不同配置選項會如何影響容錯移轉區域特性的語法。

- 偏好使用的節點或是偏好使用的成員：偏好使用的節點乃被指定來執行特定服務的成員。我們可藉由為正好一個成員指定無順序、不受限的容錯移轉區域來模擬此特性。
- 受限區域：綁定至特定區域的服務，只能在也屬於該容錯移轉區域成員的叢集上執行。若容錯移轉區域中沒有可用的成員叢集，服務將會進入停止狀態中。在一個擁有數個成員的叢集中，使用受限容錯移轉區域能簡化叢集服務（例如 httpd）的配置，因為此類型服務的配置，需要您在執行該服務的所有成員上，進行相同的配置。與其將整個叢集設定執行該叢集服務，您僅必須設定與叢集服務相聯的受限容錯移轉區域中的成員。
- 不受限的區域：此乃預設特性，綁定至此區域的服務可在所有叢集成員上執行，並且會在區域成員可使用時，在該成員上執行。這代表若一項服務執行於區域外部，而有個區域成員上線時，服務將會遷移至該成員上，除非設定了 **nofailback**。
- 照順序的區域：指定於配置中的順序可支配區域中之成員的偏好順序。階級最高的區域成員將會在上線時，執行服務。這代表若成員 A 的階級比成員 B 高，服務原本在成員 B 上執行而成員 A 上線時，該服務便會遷移至成員 A 上執行。
- 不按照順序的區域：此乃預設的特性，區域的成員沒有偏好的順序；任何成員皆可執行服務。服務會盡可能遷移至其容錯移轉區域的成員上。然而，這會是個不按照順序的區域。

- 容錯回復：在一個按照順序的容錯移轉區域成員上的服務，應容錯回復至其原本執行於（在失效之前）的節點上，這對於時常失效的節點來說相當有幫助，它有助於避免服務經常在失效的節點，以及容錯移轉節點之間進行遷移。

Ordering、restriction 以及 nofailback 屬於旗標，並且能以幾乎任何的方式（例如 ordered+restricted、unordered+unrestricted 等等）組合使用。這些組合會影響服務在初始仲裁形成後，會在哪裡啓用，以及當服務失效的情況下，哪個叢集成員會接管該服務。

3.1.1. 特性範例

一個含有這組成員的叢集：{A, B, C, D, E, F, G}。

按照順序、受限的容錯移轉區域 {A, B, C}

取消 nofailback 設定：一項服務 'S' 總是會在成員 'A' 上線並擁有仲裁時，在 'A' 上執行。若 {A, B, C} 所有的成員皆離線的話，服務便不會執行。若服務原本在 'C' 上執行，而 'A' 上線時，服務便會遷移至 'A' 上。

當設置了 nofailback 時：當仲裁形成後，服務 'S' 將會在優先權最高的叢集成員上執行。若 {A, B, C} 的所有成員皆處於離線狀態，服務將不會執行。若服務原本在 'C' 上執行而 'A' 上線時，服務將會保持在 'C' 上執行，除非 'C' 失效。在此情況下它將會容錯移轉至 'A' 上。

不按照順序、受限的容錯移轉區域 {A, B, C}

一項服務 'S' 僅會在有仲裁並且 {A, B, C} 中至少一個成員上線時才會執行。當區域的另一成員上線時，服務將不會重新定位。

按照順序、不受限的容錯移轉區域 {A, B, C}

取消 nofailback 設置：每當擁有仲裁時，服務 'S' 便會執行。當容錯移轉區域的成員上線時，服務便會在優先權最高的成員上執行，否則將會有個叢集成員被隨機選擇來執行該服務。這代表每當 'A' 上線時，服務便會在 'A' 上執行，其次為 'B'。

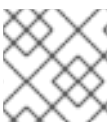
設置 nofailback：每當擁有仲裁時，服務 'S' 便會執行。每當仲裁形成時，若容錯移轉區域的成員之一上線的話，服務便會在擁有最高優先權的容錯移轉區域成員上執行。也就是說，當 'B' 在線上（而 'A' 未上線）時，服務將會在 'B' 上執行。之後當 'A' 加入叢集時，服務也不會重新定位至 'A'。

不按照順序、不受限的容錯移轉區域 {A, B, C}

這亦稱為「偏好使用的成員」。當一或多個容錯移轉區域的成員在線上時，服務將會在非特定的線上容錯移轉成員中執行。當另一容錯移轉成員上線時，服務將不會重新定位。

3.2. 服務政策

RGManager 擁有三項服務復原政策，這能經由管理員根據各項服務進行自訂。



注意

這些政策亦適用於虛擬機器資源。

3.2.1. 啓用政策

RGManager 就預設值會在 rgmanager 啓動並擁有仲裁時啓用所有服務。管理員可修改此特性。

- autostart（預設值）- 在rgmanager 啓動而仲裁形成時啓用服務。若設為「0」的話，叢集將不會啓用服務，而是會將服務更改為「停用」狀態。

3.2.2. 復原政策

復原政策乃服務在特定節點上失效時，rgmanager 所會進行的預設動作。可用選項有三個，其定義如下。

- restart（預設值）- 在相同的節點上重新啓用服務。若未指定其它復原政策，此復原政策便會被使用。若重新啓用失敗，rgmanager 便會返回重新定位該服務。
- relocate - 嘗試在叢集中的其它節點上啓用服務。若沒有其它節點可成功啓用服務，該服務便會被置入停止狀態中。
- disable - 什麼也不做。將服務置入停用狀態中。
- restart-disable - 嘗試就地重新啓用服務。若重新啓用失敗，則將服務置入停用狀態中。

3.2.3. 重新啓用政策延伸

當使用重新啓用復原政策時，您可額外指定一個最大閾值，以制定在一段時間之內，一個相同節點中所允許重新啓用的次數。若要進行這項控制，您可使用服務的 max_restarts 和 restart_expire_time 這兩項可用參數。

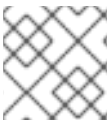
max_restarts 參數是個整數值，用來指定重新啓用服務的最大次數，在這之後，服務便會被重新定位至叢集中的其它主機上。

restart_expire_time 參數會告知 rgmanager 需記得一項重新啓用事件多久。

當同時使用這兩項參數時，您可指定在一段時間內允許重新啓用的次數。比方說：

```
<service name="myservice" max_restarts="3" restart_expire_time="300" ...>
  ...
</service>
```

以上的服務容錯為五分鐘內最多可重新啓用三次。在 300 秒內若發生第四次服務失效的話，rgmanager 將不會重新啓用服務，而是會將服務重新定位至叢集中的其它可用主機上。



注意

您必須同時指定這兩項參數；僅使用單一參數將會造成參數無法定義。

3.3. 資源樹 - 基礎 / 定義

以下內容描述了資源樹的結構，以及定義了各個部分的相應清單。

```
<service name="foo" ...>
    <fs name="myfs" ...>
        <script name="script_child"/>
    </fs>
    <ip address="10.1.1.2" .../>
</service>
```

- 資源樹乃資源、其屬性、父/子與同層級關係的 XML 表示式。資源樹的根部一般會是一種特殊類型的資源稱為「服務」。在此 wiki 上，資源樹、資源群組和服務一般會交互使用。以 rgmanager

的角度來看，資源樹是個原子單位（atomic unit）。一個資源樹的所有元件皆會由相同的叢集節點上作為起始。

- fs:myfs 和 ip:10.1.1.2 屬於同層級的資源
- fs:myfs i乃 script:script_child 的父系資源
- script:script_child 乃 fs:myfs 的子系資源

3.3.1. 父/子關係、相依性和起始順序

資源樹中的父/子關係規則相當單純：

- 父系資源會比子系資源先起始
- 父系資源停止前，所有的子系資源必須先完全停止
- 由這兩項規則看來，您可斷言子系資源依賴其父系資源
- 若資源要被視為健全，所有其相依的子系資源也必須處於健全狀態

3.4. 服務操作與狀態

下列操作適用於服務和虛擬機器，除了遷移上的操作，該操作僅適用於虛擬機器。

3.4.1. 服務操作

服務操作為使用者可調用的可用指令，以套用定義於下列清單中，五種可用動作中的其中一項。

- enable — 啓用服務，可選用性在一個偏好的目標上啓用，以及選用性根據容錯移轉區域規則啓用。當未指定任何一項選項時，執行 clusvcadm 的本地主機將會執行服務。若原始啓用失敗，服務便會如請求了重新定位作業一般地進行（請參閱以下部分）。若作業成功，服務便會被置入「已啓用」狀態中。
- disable — 停止服務，並將其置入「已停用」狀態中。此乃當服務處於失效狀態時，唯一允許進行的操作。
- relocate — 將服務移至另一節點上。管理員可選用性指定偏好使用的節點來接收服務，不過服務無法在該主機上執行（比方說若服務啓用失敗或是主機離線的話）將無法避免重新定位，並且另一個節點將會被選擇。Rgmanager 會嘗試在叢集中，所有允許的節點上啓用服務。若叢集中沒有目標節點成功啓用服務的話，重新定位將會失敗並且服務將會被嘗試重新啓用於原始節點上。若原始節點上無法重新啓用服務的話，服務便會被置入「已停止」狀態中。
- stop — 停止服務並將其置入「已停止」狀態中。
- migrate — 將虛擬機器遷移至另一節點上。管理員必須指定一個目標節點。根據失效原因，遷移失敗可能會造成虛擬機器進入失效狀態，或在原始節點上進入啓用狀態。

3.4.1.1. freeze 作業

RGManager 可將服務凍結。這麼做能讓使用者升級 rgmanager、CMAN 或是系統上任何其它軟體，並同時有效減少 rgmanager 所管理之服務的停擺時間。

它亦可讓使用者進行 rgmanager 服務的維護。比方說，若您在一個單獨的 rgmanager 服務中有個資料庫和網站伺服器，您可凍結 rgmanager 服務、停用資料庫、進行維護、重新啓用資料庫，以及取消服務凍結。

3.4.1.1.1. 凍結時的服務特性

- 狀態檢查將會停用
- 啓用作業將會停用
- 停止作業將會停用
- 容錯移轉不會發生（儘管您將服務擁有者的電源關閉）

重要

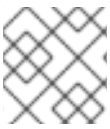
若沒依照這些分針進行，可能會造成資源被分配在多部主機上。

- 您絕對不能在一項服務凍結時中止所有 rgmanager，除非您計劃在重新啓用 rgmanager 之前，重新啓動主機。
- 您絕對不能在一項服務的原始節點加入叢集並重新啓用 rgmanager 之前，將一項服務取消凍結。

3.4.2. 服務狀態

以下清單定義了 RGManager 所管理之服務的狀態。

- disabled — 服務會持續處於停用狀態，直到管理員重新啓用服務，或是叢集失去仲裁（在此情況下，autostart 參數將會被評估）。管理員可在此狀態下啓用服務。
- failed — 服務被假設已失效。每當資源的 stop 作業失效時，便會發生此狀態。管理員在發出一項停用請求之前，必須先驗證是否有已分配的資源（掛載的檔案系統等等）。能在此狀態下進行的動作只有「停用」。
- stopped — 當處於已停止的狀態時，服務將會被評估，並在下次服務或節點轉換後啓用。這是個暫時性的措施。管理員可在此狀態下停用或啓用服務。
- recovering — 叢集正在嘗試復原服務。管理員可視需求停用服務以避免復原。
- started — 若一項服務狀態檢查失敗的話，請根據服務復原政策來將它復原。若執行服務的主機失效的話，請依照容錯移轉區域和獨佔性的服務規則來將它復原。管理員可由此狀態中重新定位、中止、停用，和（虛擬機器）遷移服務。



注意

其它狀態，例如 **`starting`** 和 **`stopping`** 乃 **`started`** 狀態的特殊可轉換狀態。

3.5. 虛擬機器特性

RGManager 處理虛擬機器的方式和其它非 VM 服務的處理方式不太一樣。

3.5.1. 正常作業

rgmanager 所管理的虛擬機器只應透過 clusvcadm 或是另一項叢集感知的工具來進行管理。大部份的特性對於一般正常服務來說皆屬常見的。這些特性包含了：

- 起始（啓用中）
- 中止（停用中）
- 狀態監控
- 重新定位
- 復原

欲取得更多有關於高可用性虛擬服務上的相關資訊，請參閱〈章 7, *虛擬化與 High Availability*〉。

3.5.2. 遷移

除了正常服務作業之外，虛擬機器還支援了一項其它服務所不支援的特性：遷移。遷移作業無需啓用/中止虛擬機器，便能將它移至叢集中的其它位置上，大幅減少了作業停擺的時間。

rgmanager 支援兩種類型的遷移，您可透過以下遷移屬性來針對各個 VM 進行選擇：

- live（預設值）— 虛擬機器會持續執行，而其大部份的記憶體內容皆會被複製至目標主機上。這可大幅減少 VM 的停擺時間（一般約 1 秒以下），所換取的是 VM 在進行遷移時的效能，以及較長的遷移所需時間。
- pause — 虛擬機器將會被凍結在記憶體中，而其記憶體內容將會被複製至目標主機上。這可大幅減少虛擬機器完成遷移的所需時間。

您所使用的遷移形式取決於可用性及效能需求。比方說，一項即時遷移可能代表 29 秒的效能降級以及 1 秒鐘的完全停機時間，而暫停遷移可能代表 8 秒的完全停機時間，但無效能降級的問題。

重要

虛擬機器能夠屬於服務的元件，不過這麼做可能會停用所有形式的遷移，以及下列大部份的便利功能。

此外，搭配 KVM 進行的遷移，需經過精心的 ssh 配置。

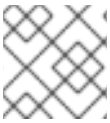
3.5.3. RGManager 虛擬機器功能

下列部分列出了各種 RGManager 用來簡化虛擬機器管理的方式。

3.5.3.1. 虛擬機器追蹤

若在虛擬機器已在運作中的情況下透過 **clusvcadm** 啓用虛擬機器，這會造成 rgmanager 在叢集中搜尋虛擬機器，並且無論在哪裡發現該虛擬機器，皆會將它標記為 **started**。

若管理員不小心透過非叢集的工具（例如 **virsh**）來在叢集節點之間遷移虛擬機器，這會造成 rgmanager 在叢集中搜尋虛擬機器，並且無論在哪裡發現該虛擬機器，皆會將它標記為 **started**。



注意

若虛擬機器在多個位置上執行，RGManager 則不會警告您。

3.5.3.2. 暫時性區域支援

Rgmanager 支援 libvirt 所支援的暫時性虛擬機器。這能讓 rgmanager 輕易地建立和移除虛擬機器，並協助降低因為使用非叢集工具，而意外雙重啓用虛擬機器的機會。

支援暫時性虛擬機器亦可讓您在一個叢集檔案系統上儲存 libvirt XML 描述檔案，因此您便無須手動式在叢集之間同步 **/etc/libvirt/qemu**。

3.5.3.2.1. 管理功能

從 cluster.conf 中移除或新增虛擬機器不會啓用或中止虛擬機器；它僅會使 rgmanager 開始或停止注意虛擬機器

透過使用遷移來進行備援（移至一個偏好使用的節點），以減少停機時間。

3.5.4. 未處理的行為

RGManager 中不支援下列情況和使用者動作。

- 使用一項非叢集感知的工具（例如 virsh 或是 xm）來在叢集管理虛擬機器時，操作虛擬機器的狀態或配置。檢查虛擬機器的狀態是否無誤（例如 virsh list、virsh dumpxml）。
- 將一部由叢集管理的虛擬機器遷移至一個非叢集的節點，或是一個位於未執行 rgmanager 之叢集中的節點上。Rgmanager 會在先前的位置中重新啓動虛擬機器，造成兩個虛擬機器 instance 同時執行，並導致檔案系統損毀。

3.6. 資源動作

RGManager 會預期下列來自於資源代理程式的回傳值：

- start - 啓用資源
- stop - 停用資源
- status - 檢查資源的狀態
- metadata - 回報 OCF RA XML metadata

3.6.1. 回傳值

OCF 會為監控作業提供大量回傳碼，不過因為 rgmanager 調用 status，因此它幾乎只會依賴 SysV-style 回傳碼。

0 - 成功

停用後停止或沒有執行時停止必須回傳成功

啓用後開始或執行時開始必須回傳成功

非零 - 失敗

若 stop 作業回傳了一個非零的值，服務便會進入「失敗」狀態，並且該服務將必須透過手動的方式復原。

章 4. 隔離

「隔離（fencing）」代表將節點由叢集的共享儲存裝置中移除。隔離會切斷所有來自共享儲存裝置的 I/O，以確保資料的完整性。叢集基礎結構會透過隔離系統程式（**fenced**）來進行這項工作。

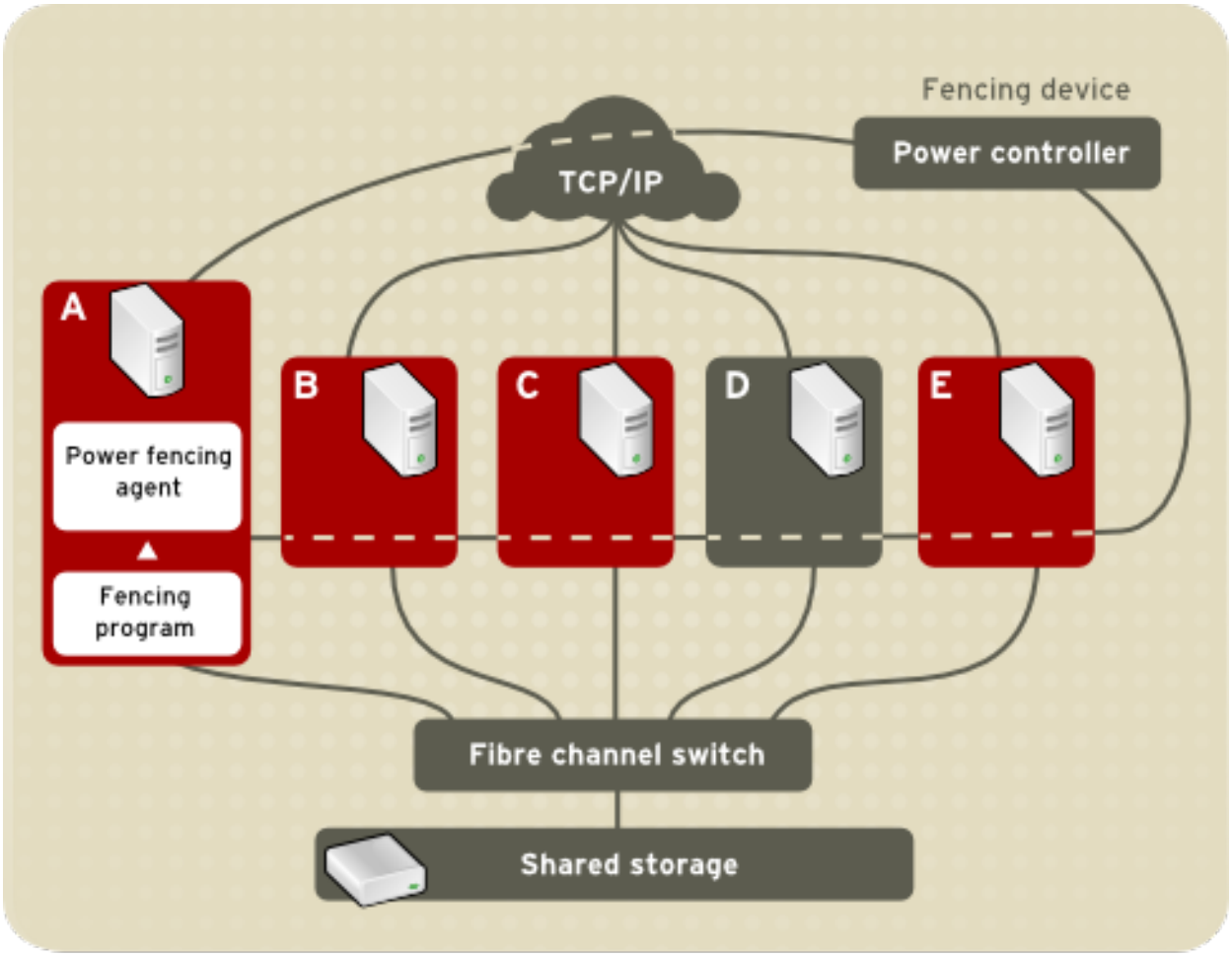
當 CMAN 發現一組節點失效時，CMAN 會與其它叢集基礎結構的元件進行通訊，告知該節點已失效。當通知了 **fenced** 時，**fenced** 會將失效的節點隔離。其它叢集基礎結構的元件會決定該進行哪些動作 — 亦即會進行任何復原動作。舉例來說，DLM 與 GFS2 收到節點失效的通知時，會暫時停止活動，直到 **fenced** 完成隔離失效節點為止。確認失效節點已經隔離後，DLM 與 GFS2 就會開始進行復原。DLM 會解除對於失效節點的鎖定；GFS2 會復原失效節點的日誌檔。

隔離程式會從配置檔案來決定要採取何種隔離措施。配置檔案中有兩個主要因素決定隔離的措施：隔離代理程式和隔離裝置。隔離程式會調用叢集配置檔案中所指定的代理程式。接著，代理程式便會透過隔離裝置將節點隔離。當隔離完成後，隔離程式便會通知叢集管理員。

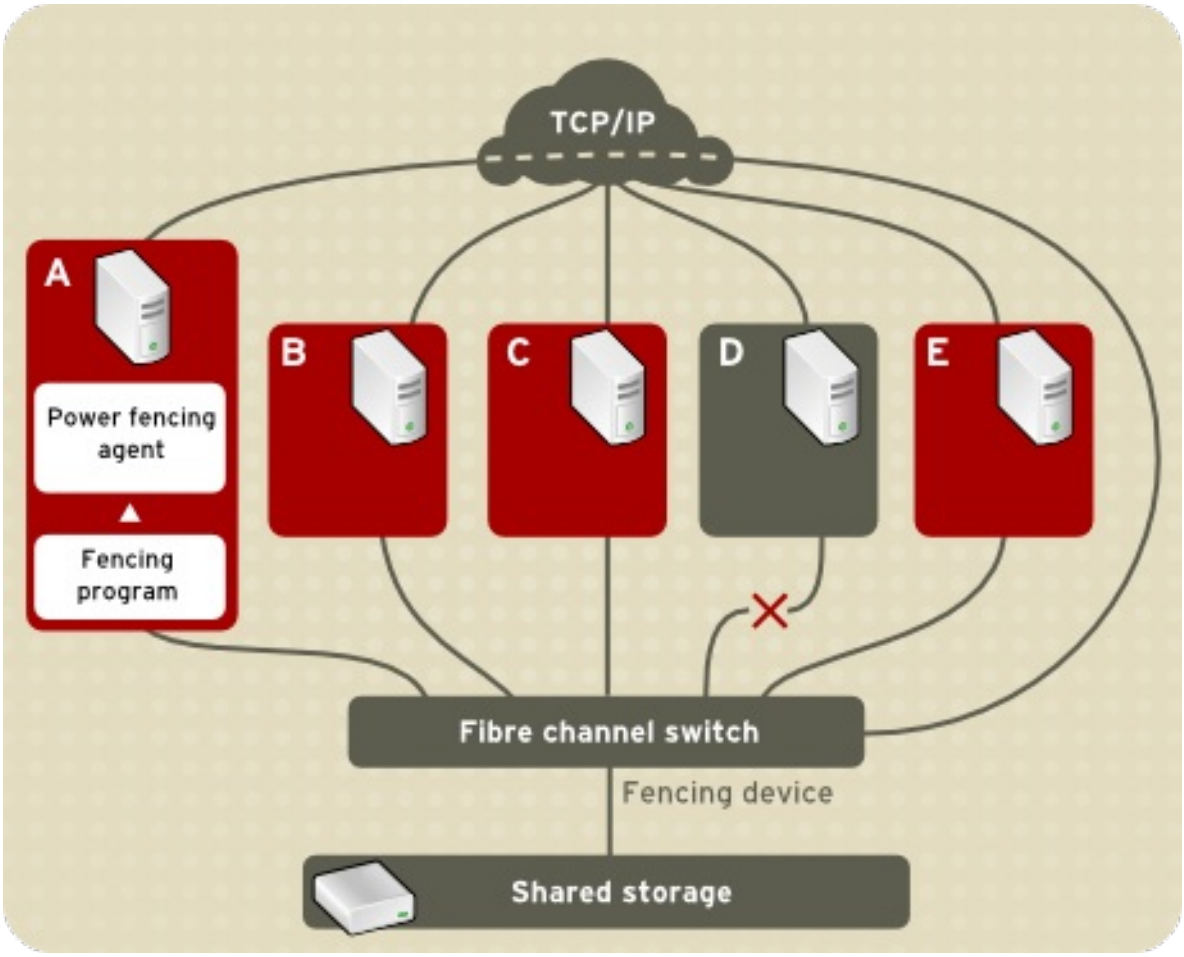
High Availability 外掛程式提供了多項隔離措施：

- 電源隔離 — 一項透過電源控制器，以將無法運作之節點關機的隔離措施。
- 儲存裝置隔離 — 這項隔離措施會停用將儲存裝置連至一個無法操作之節點的光纖頻道連接埠。
- 其它隔離措施 — 其它停止 I/O 或關閉失效節點的方法，包括 IBM Bladecenter、PAP、DRAC/MC、HP ILO、IPMI、IBM RSA II 等等。

〈圖形 4.1, “電源隔離範例”〉顯示了電源隔離的範例。在範例中，節點 A 中的隔離程式會造成電源控制器將節點 D 關閉。〈圖形 4.2, “儲存裝置隔離範例”〉顯示了隔離儲存裝置的範例。在範例中，節點 A 的隔離程式會使光纖頻道切換器將節點 D 的連接埠停用，並切斷節點 D 與儲存裝置之間的連線。


Fencing device
Power controller
TCP/IP
A
B
C
D
E
Power fencing agent
Fencing program
Fibre channel switch
Shared storage

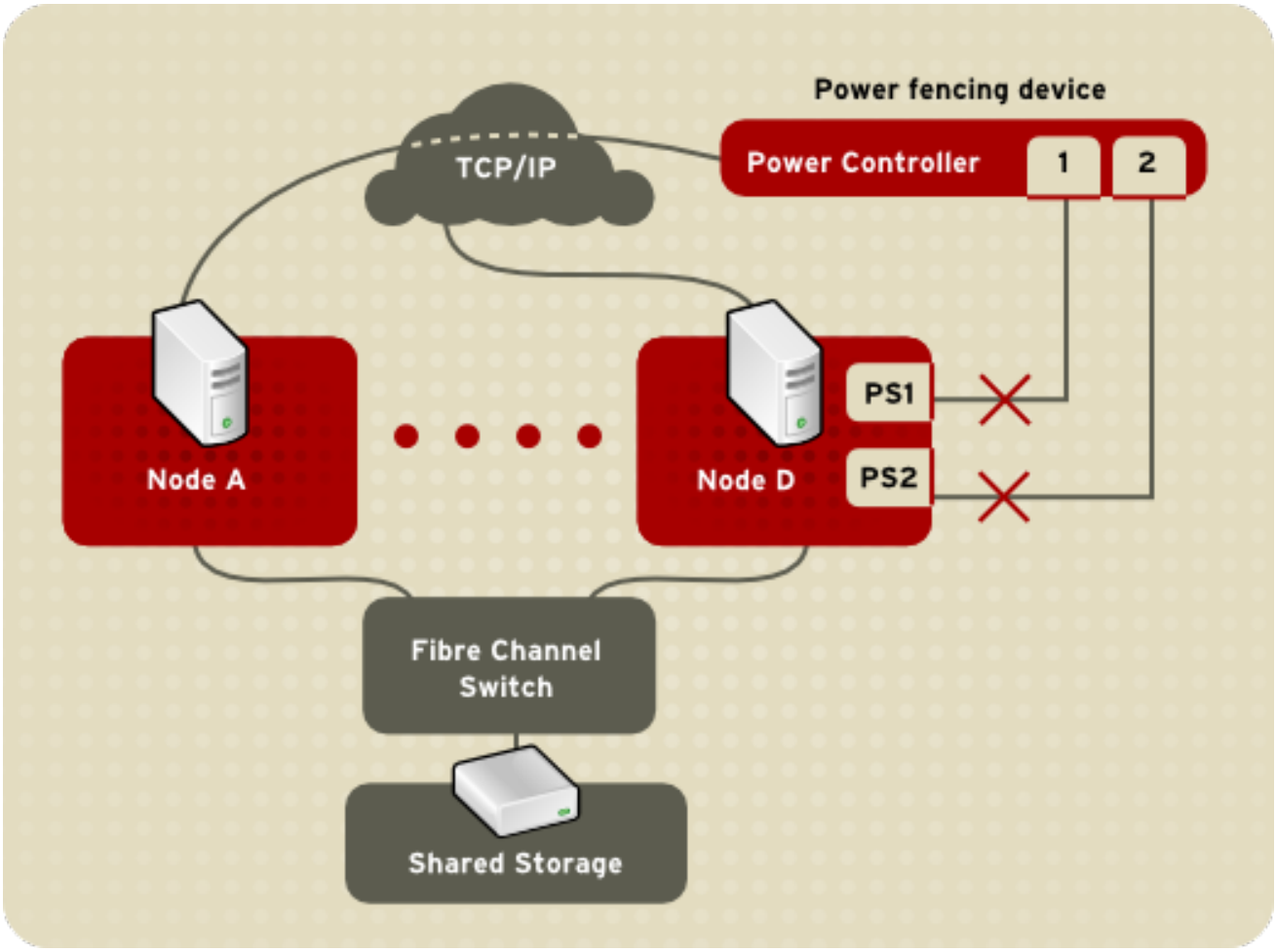
圖形 4.1. 電源隔離範例



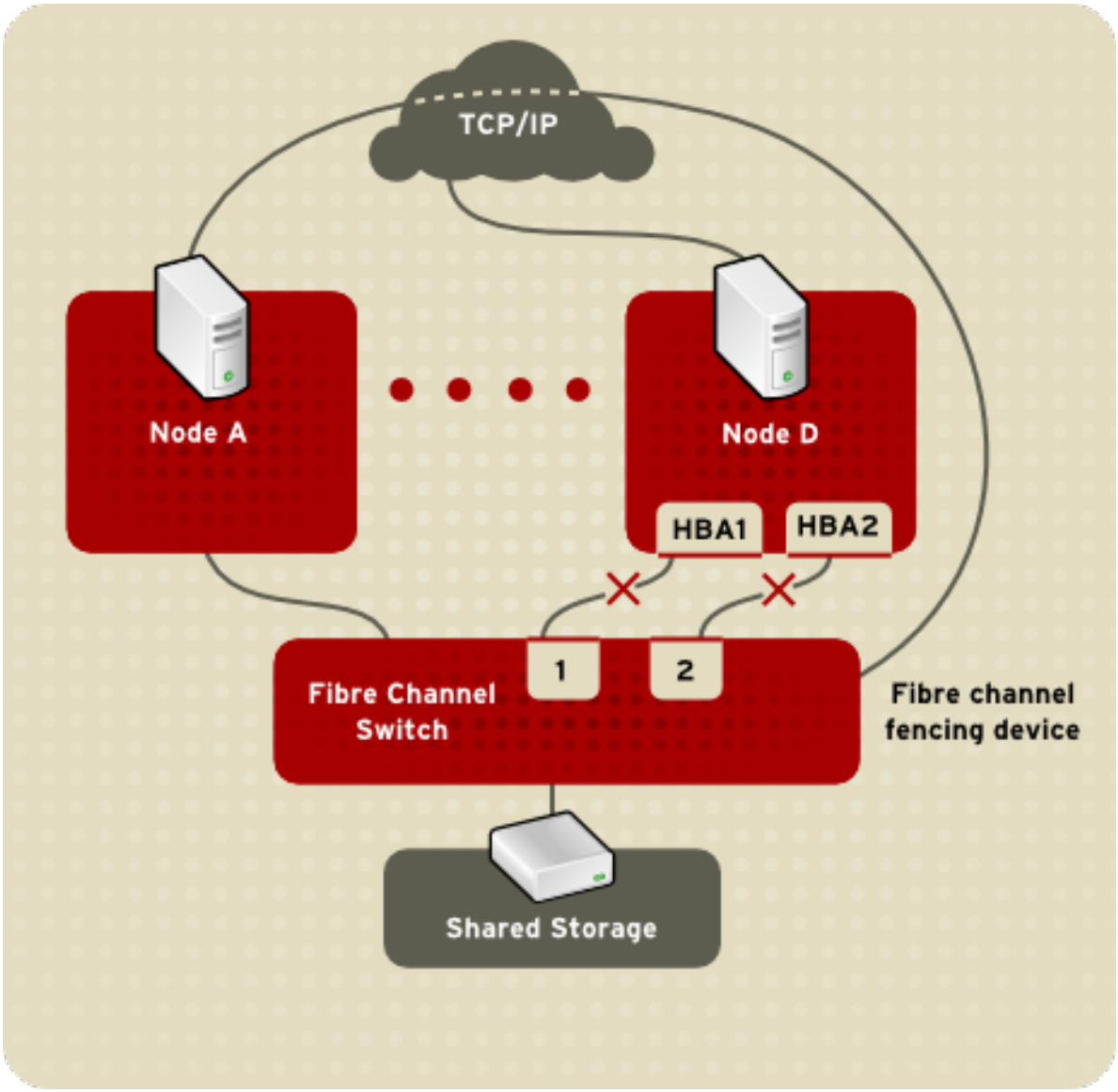
圖形 4.2. 儲存裝置隔離範例

指定一項隔離的措施，其中包含了編輯叢集配置檔案，以指定隔離措施的名稱、隔離代理程式、以及叢集中每個裝置的隔離裝置。

隔離措施的指定取決於結點是否使用雙電源供給以及是否擁有多重儲存裝置路徑。若節點擁有雙電源補給，則該節點的隔離措施便必須指定至少兩個隔離裝置 — 一個電源供給一個隔離裝置（請參閱〈圖形 4.3, “以雙電源供給（Dual Power Supplies）來隔離節點”〉）。相似地，若節點有多個能連至光纖頻道儲存裝置的路徑，則該節點的隔離措施便必須為各個連至光纖頻道儲存裝置的路徑指定一個隔離裝置。比方說，若一個節點有兩個能連至光纖頻道儲存裝置的路徑，隔離措施便應指定兩個隔離裝置 — 各個連至光纖頻道儲存裝置的路徑各一個（請參閱〈圖形 4.4, “以雙光纖頻道連線（Dual Fibre Channel）來隔離節點”〉）。



圖形 4.3. 以雙電源供給（Dual Power Supplies）來隔離節點



圖形 4.4. 以雙光纖頻道連線（Dual Fibre Channel）來隔離節點

一個節點可以設定一或多個隔離措施。當僅為一個節點使用單一隔離措施時，它將會是隔離該節點的唯一措施。當您為一個節點使用多個隔離措施時，這些措施會根據叢集配置檔案中的設定，按順序*串聯*在一起。如果一個節點失效，它會使用配置檔案中的第一個隔離措施。如果第一個措施不成功，便會使用配置檔案中的第二個措施。如果所有措施皆不成功，那麼系統便會返回重新嘗試所指定的第一個隔離措施，並按照叢集配置檔案中所指定的順序嘗試所有隔離措施，直到節點被隔離為止。

欲取得更多有關於配置隔離裝置上的詳細資訊，請參閱《*叢集管理*》指南中的相應章節。

章 5. 鎖定管理

鎖定管理是項常見的叢集基礎結構服務，它提供了一項機制，以讓其它叢集基礎結構元件同步共享資源的存取。在 Red Hat 叢集中，DLM（Distributed Lock Manager）便是鎖定管理程式。

鎖定管理程式負責控制叢集中的資源存取，比方說 GFS 檔案系統的存取。您需要這項機制，因為若缺少鎖定管理程式的話，便沒有能控制您共享儲存裝置存取上的機制，並且叢集中的節點將可能會損毀互相的資料。

顧名思義，DLM 是個散佈式鎖定管理程式，並且執行於各個叢集節點中；鎖定管理會被散佈至叢集中的所有節點上。GFS2 和 CLVM 使用了來自於鎖定管理程式的鎖定。GFS2 使用了來自於鎖定管理程式的鎖定來同步（位於共享儲存裝置上的）檔案系統 metadata 的存取。CLVM 則使用來自於鎖定管理程式的鎖定來同步 LVM 卷冊和卷冊群組的更新（也是在共享儲存裝置上）。此外，**rgmanager** 使用了 DLM 來同步服務狀態。

5.1. DLM 鎖定模組

DLM 鎖定模組提供了大量的鎖定模式，以及同步與非同步的執行。應用程式會在一項鎖定資源上取得鎖定。鎖定資源與鎖定之間有項一對多的關係：單一鎖定資源能夠有多個與它相聯的鎖定。

鎖定資源可與一項實際的物件相對應，比方說一個檔案、資料結構、資料庫，或是一個可執行的外部常式，不過它不一定非要與它們相對應。您用來與鎖定資源相聯的物件，可判定鎖定的規模。將資料庫中的各項項目鎖定，會被視為是精細規模的鎖定。

DLM 鎖定模組支援：

- 六種不同資源存取限制的鎖定模式
- 透過轉換動作來進行鎖定的升級與降級
- 同步完成鎖定請求
- 非同步完成
- 透過鎖定值區塊制訂的全域資料

DLM 自行提供了支援其鎖定功能的機制，比方說管理鎖定流量的節點間通訊，以及在節點失效後重新制訂鎖定，和在節點加入叢集時遷移鎖定的復原協定。然而，DLM 並不會提供實際管理叢集本身的機制。因此，DLM 會被預期在一個叢集中，連同另一個提供了下列最小需求的叢集基礎結構環境聯合運作：

- 節點乃叢集的一部分。
- 所有節點皆接受叢集成員並擁有仲裁。
- 節點上必須有組和 DLM 通訊的 IP 位址。一般來說，DLM 會使用 TCP/IP 來進行節點間的通訊，這會限制一個節點僅能有單一組 IP 位址（儘管您可使用 bonding driver 來進行重複）。DLM 可被配置來使用 SCTP 作為其節點間的傳送工具，這便能允許一個節點擁有多組 IP 位址。

DLM 適用於任何滿足了上述基本需求的任何叢集基礎結構環境。使用開源式或是封閉式的環境取決於使用者。然而，DLM 的主要缺點就是進行測試的不同環境有限。

5.2. 鎖定狀態

鎖定狀態顯示了一項鎖定請求目前的狀態。鎖定總是會處於以下三種狀態中：

- 已授與 — 鎖定請求已成功並且請求的模式已取得。

- 轉換中 — 客戶端常式更改鎖定模式，並且新的模式與既有的鎖定不相容。
- 已阻擋 — 無法授與新鎖定的請求，因為有造成衝突的鎖定存在。

鎖定的狀態是以其請求的模式，和相同資源上的其它鎖定模式來判定的。

章 6. 配置與管理工具

叢集配置檔案 **/etc/cluster/cluster.conf** 可用來指定 High Availability 外掛程式的配置。此配置檔案是個用來詳述下列叢集特性的 XML 檔案：

- 叢集名稱 — 指定叢集的名稱、叢集配置檔案修訂等級，以及基本隔離定時內容，用於當一個節點加入叢集或是由叢集隔離出去時。
- 叢集 — 指定叢集的各個節點，指定節點名稱、節點 ID、仲裁投票數量，以及該節點的隔離措施。
- 隔離裝置 — 指定叢集中的隔離裝置。其參數將會根據隔離裝置的類型而定。比方說一個用來作為隔離裝置的電源控制器，叢集配置將會定義電源控制器的名稱、其 IP 位址、登錄帳號與密碼。
- 管理資源 — 指定建立叢集服務所需的資源。管理資源包含了容錯移轉區域的定義、資源（比方說一組 IP 位址），以及服務。在一起，這些管理資源將能夠定義叢集服務，以及這些服務的容錯移轉特性。

叢集配置會在啓動和重新載入配置時，根據位於 **/usr/share/cluster/cluster.rng** 的叢集結構描述，自動地進行驗證。並且您亦可在任何時候透過使用 **ccs_config_validate** 指令來驗證叢集配置。

您可檢視位於 **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** 的結構描述（例如 **/usr/share/doc/cman-3.0.12/cluster_conf.html**）。

配置驗證機制會檢查下列基本項目：

- XML 的有效性 — 檢查配置檔案是否是個有效的 XML 檔案。
- 配置選項 — 檢查以確認選項（XML 要素與屬性）有效。
- 選項值 — 檢查選項是否包含了有效的資料（有限）。

6.1. 叢集管理工具

Red Hat High Availability 外掛程式軟體的管理，包含使用配置工具來指定叢集元件之間的關係。Red Hat High Availability 外掛程式包含了下列叢集配置工具：

- **Conga** — 這是個用來安裝、配置與管理 Red Hat High Availability 外掛程式的使用者介面。欲取得透過 **Conga** 配置和管理 High Availability 外掛程式上的相關資訊，請參閱《*配置和管理 High Availability 外掛程式*》。
  - Luci — 這是個提供了 Conga 使用者介面的應用程式伺服器。它能讓使用者管理叢集服務，並在需要時提供協助和線上文件的存取。
  - Ricci — 這是一項服務 daemon，負責管理叢集配置的發佈。使用者可透過 Ricci 介面來傳送配置詳細資料，並且配置將會被載入 corosync，以發佈給叢集節點。
- 由 Red Hat Enterprise Linux 6.1 發行版和更新版本起，Red Hat High Availability 外掛程式開始支援 **ccs** 叢集配置指令，這能讓管理員建立、修改和檢視 **cluster.conf** 叢集配置檔案。欲取得更多有關於透過 **ccs** 指令配置和管理 High Availability 外掛程式的相關資訊，請參閱《*叢集管理*》指南。

注意

system-config-cluster 無法使用於 RHEL 6 中。

章 7. 虛擬化與 HIGH AVAILABILITY

多種虛擬化平台已受到支援，可以與結合了 High Availability 與 Resilient Storage 外掛程式的 RHEL 6 一起運作。虛擬化結合 Red Hat Enterprise Linux High Availability 外掛程式有兩種情況，是受到支援的。

這指的是執行於空機電腦上的 RHEL 叢集/High Availability，這些空機電腦亦為虛擬平台。在這模式下，您可以配置叢集資源管理程式（rgmanager）來管理虛擬機器（客座端），使其成為高度可用的資源。

- 將虛擬機器作為高度可用的資源/服務
- 客座端叢集

7.1. 將虛擬機器作為高度可用的資源/服務

RHEL HA 與 RHEV 都有提供 HA 虛擬機器的機制。鑑於兩者在功能上的重疊，請小心選擇正確的產品，以符合您的特定使用情境。以下是要提供高可用性的虛擬機器時，選擇 RHEL HA 或 RHEV 的指引。

對於虛擬機器與實體主機數量：

- 如果您在大量實體主機上建立大量 HA 虛擬機器，那麼使用 RHEV 可能是更好的解決方案，因為 RHEV 管理虛擬機器的演算法則更為複雜，並納入對主機 CPU、記憶體、負載資訊等考量。
- 如果您在少量實體主機上建立少量 HA 虛擬機器，那麼使用 RHEV HA 可能是更好的解決方案，因為不需要額外的架構。RHEL HA 虛擬機器的解決方案至少需要兩台實體主機，以建立雙節點的叢集。RHEV 的解決方案至少需要四個節點：兩台為 RHEVM 伺服器提供 HA 功能，另外兩台作為虛擬主機。
- 並無明文規定多少主機或虛擬機器代表「大量」。不過請記得，單一 RHEL HA 叢集中所允許的最大主機數量為 16，並且任何包含了 8 或更多部主機的叢集，將需要經過 Red Hat 進行架構上的檢測，以判定支援能力。

虛擬機器的使用方法：

- 若您的 HA 虛擬機器正在提供服務並且提供了共享基礎結構的話，您可使用 RHEL HA 或是 RHEV。
- 若您需要為虛擬機器中執行的一組重要服務提供 HA，您可使用 RHEL HA 或是 RHEV。
- 若您計劃提供基礎結構，以允許快速佈建虛擬機器，您應使用 RHEV。
  - RHEV VM HA 應該要是動態式的。新增虛擬機器至 RHEV「叢集」非常容易，並且受到了完整支援。
  - RHEL VM HA 不應該是個高度動態式的環境。應設定擁有固定虛擬機器的叢集，並且在該叢集生命週期的有效期限內，不建議再新增或移除額外的虛擬機器。
- 基於叢集配置的靜態特性，以及實體節點最大數量限制較低（16 個節點），RHEL HA 不應被使用來提供基礎結構，以建立類似雲端的環境。

RHEL 5 支援兩種虛擬化平台。Xen 從 RHEL 5.0 發行版起便受到支援。RHEL 5.4 則發表了 KVM。

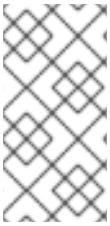
RHEL 6 僅支援 KVM 為虛擬化平台。

RHEL 5 AP 叢集支援使用 KVM 和 Xen 來執行由主機叢集基礎結構管理的虛擬機器。

RHEL 6 HA 支援使用 KVM 來執行由主機叢集基礎結構管理的虛擬機器。

下列清單列出了 Red Hat 目前所支援的建置方案：

- RHEL 5.0+ 支援 Xen 結合 RHEL AP 叢集
- RHEL 5.4 開始支援 KVM 虛擬機器作為 RHEL AP Cluster 中受管理的資源，此乃技術預覽。
- RHEL 5.5+ 將 KVM 虛擬機器的支援等級提升為完整支援。
- RHEL 6.0+ 支援在 RHEL 6 High Availability 外掛程式中，使用 KVM 虛擬機器來作為高可用性資源。
- RHEL 6.0+ 不支援 Xen 虛擬機器搭配 RHEL 6 High Availability 外掛程式，因為 RHEL 6 已不再支援 Xen。



注意

欲取得有關於受支援之建置方案上的更新資訊和特殊備註，請參閱以下 Red Hat 知識庫項目：

https://access.redhat.com/kb/docs/DOC-46375

作為受管理的資源執行的虛擬機器類型並不重要。在 RHEL 中任何 Xen 或 KVM 所支援的客座端，皆可被使用來作為高可用性客座端。這包含了各版本的 RHEL（RHEL3、RHEL4、RHEL5），以及多種版本的微軟 Windows 作業系統。請查看 RHEL 的相關文件，以檢查各個 hypervisor 下，最新支援的客座端作業系統清單。

7.1.1. 一般建議

- 在 RHEL 5.3 和較舊版本中，rgmanager 使用了原生的 Xen 界面來管理 Xen domU（客座端）。在 RHEL 5.4 中，Xen 與 KVM 這兩個 hypervisor 類型的界面，已藉由使用 libvirt，提供了一致性的界面。除了這項架構變更之外，RHEL 5.4 和 5.4z 還發佈了多項錯誤修正，因此建議您在配置 Xen 所管理的服務之前，至少為您的主機叢集升級最新的 RHEL 5.5 套件。
- 若要配置 KVM 所管理的服務，您必須升級至 RHEL 5.5，因為這是完整支援這項功能的第一個 RHEL 版本。
- 在建置叢集前，總是先查看最新的 RHEL 勘誤，以確保您擁有最新的修正檔，以修正已知的問題或錯誤。
- 不支援混合使用主機和不同類型的 hypervisor。主機叢集必須全部基於 Xen 或是 KVM。
- 您應將主機硬體佈建成能夠吸收來自於多個失效主機的重定位客座端，並且不造成主機過量使用記憶體或是嚴重過量使用虛擬 CPU。若失效情況過多並造成了記憶體或是虛擬 CPU 過量使用，這可能會導致於嚴重的效能下降，甚至是叢集失效。
- 不支援或建議直接使用 xm 或 libvirt 工具（virsh、virt-manager）來管理（即時遷移、中止、開始）由 rgmanager 控制的虛擬機器，因為這將會跳過叢集管理堆疊。
- 在叢集中，各組虛擬機器的名稱皆必須要是獨特的，包括唯本機/非叢集的虛擬機器。Libvirtd 僅會根據各個主機強制其使用獨特的名稱。若您手動複製一個虛擬機器，您必須在副本的配置檔案中，更改其名稱。

7.2. 客座端叢集

這代表在各種虛擬化平台上的虛擬客座端中執行的 RHEL Cluster/HA。在使用案例中，RHEL Clustering/HA 主要被用來確保執行於客座端中的應用程式的高可用性。此使用案例類似 RHEL Clustering/HA 如何使用於傳統的空機主機中。不同的地方就是 Clustering 會執行於客座端中。

以下為一列虛擬化平台，以及目前使用 RHEL Cluster/HA 的客座叢集，所擁有的支援等級。在以下清單中，RHEL 6 客座端將圍繞著 High Availability（核心叢集）以及 Resilient Storage 外掛程式（GFS2、clvmd 和 cmirror）。

- RHEL 5.3+ 的 Xen 主機能完全支援執行客座叢集，而客座端作業系統也必須要是 RHEL 5.3 或更新版本：
  - Xen 客座叢集可使用 fence_xvm 或是 fence_scsi 來進行客座端隔離。
  - 若要使用 fence_xvm/fence_xvmd，必須要有一個運作中的主機叢集，以支援 fence_xvmd，並且 fence_xvm 必須被使用來在所有叢集客座端上作為客座端隔離代理程式。
  - 共享儲存裝置能以受到主機區塊儲存裝置，或是檔案型儲存裝置（原生映像檔）支援的 iSCSI 或是 Xen 共享區塊裝置提供。
- RHEL 5.5+ 的 KVM 主機不支援執行客座叢集。
- RHEL 6.1+ 的 KVM 主機支援執行客座叢集，而客座端作業系統必須要是 RHEL 6.1+ 或是 RHEL 5.6+。不支援 RHEL 4 客座端。
  - 允許混合使用空機叢集節點和虛擬化的叢集節點。
  - RHEL 5.6+ 的客座叢集可使用 fence_xvm 或 fence_scsi 來進行客座端隔離。
  - RHEL 6.1+ 的客座叢集可使用 fence_xvm (在 **`fence-virt`** 套件中）或是 fence_scsi 來進行客座端隔離。
  - 若客座端使用了 fence_virt 或是 fence_xvm 來作為隔離代理程式的話，RHEL 6.1+ KVM Host 便必須使用 fence_virtd。若客座叢集使用了 fence_scsi 的話，則主機上的 fence_virtd 便是非必要的。
  - fence_virtd 可以三種模式運作：
    - 獨立模式，主機至客座端的映對乃硬式編碼，並且不允許即時遷移客座端
    - 使用 Openais Checkpoint 服務來追蹤叢集客座端的即時遷移。若要如此，必須有個運作中的主機叢集。
    - 使用 libvirt-qpid 套件所提供的 Qpid Management Framework（QMF）。這將能在沒有完整主機叢集的情況下，使用 QMF 來追蹤客座端的遷移。
  - 共享儲存裝置能以受到主機區塊儲存裝置，或是檔案型儲存裝置（原生映像檔）支援的 iSCSI 或是 KVM 共享區塊裝置提供。
- Red Hat Enterprise Virtualization Management（RHEV-M）版本 2.2+ 和 3.0 目前支援 RHEL 5.6+ 和 RHEL 6.1+ 叢集客座端。
  - 客座叢集必須是同質的（所有的 RHEL 5.6+ 客座端或是所有的 RHEL 6.1+ 客座端）。
  - 允許混合使用空機叢集節點和虛擬化的叢集節點。
  - 在 RHEV-M 2.2+ 中，隔離是由 fence_scsi 所提供，而在 RHEV-M 3.0 中則是由 fence_scsi 和 fence_rhevm 所提供的。隔離乃透過使用 fence_scsi 來支援的，詳述如下：

    - 搭配 iSCSI 儲存裝置使用 fence_scsi，僅限於在支援 SCSI 3 持續保留與 preempt 和 abort 指令的 iSCSI 伺服器上。並非所有 iSCSI 伺服器皆支援這項功能。請與您的儲存裝置廠商確認，以確保您的伺服器與 SCSI 3 持續保留支援相容。請注意，RHEL 所包含的 iSCSI 伺服器目前並不支援 SCSI 3 持續保留，因此它不適合與 fence_scsi 搭配使用。

- VMware vSphere 4.1、VMware vCenter 4.1、VMware ESX 和 ESXi 4.1 支援執行客座叢集，而客座端作業系統必須是 RHEL 5.7+ 或是 RHEL 6.2+。版本 5.0 的 VMware vSphere、vCenter、ESX 和 ESXi 亦受到支援；然而，因為 Vmware vSphere 5.0 的初始版本中包含了非完整的 WDSL 結構描述，因此 fence_vmware_soap 工具程式在預設安裝下無法運作。欲取得更新程序以修正此問題，請參閱 Red Hat 知識庫 https://access.redhat.com/knowledge/。

    - 客座叢集必須是同質的（所有的 RHEL 5.7+ 客座端或是所有的 RHEL 6.1+ 客座端）。

    - 允許混合使用空機叢集節點和虛擬化的叢集節點。

    - fence_vmware_soap 代理程式需要協力廠商的 VMware perl API。此軟體套件必須由 VMware 的網站下載，並安裝在 RHEL 叢集客座端上。

    - 此外，fence_scsi 可如以下部分中所詳述地被使用來提供隔離機制。

    - 共享儲存裝置可由 iSCSI 或是 VMware 原生共享區塊裝置來提供。

    - 您可透過 fence_vmware_so_ap 或是 fence_scsi 來支援使用 VMware ESX 客座叢集。

- 目前不支援使用 Hyper-V 客座叢集。

7.2.1. 使用 **fence_scsi** 和 **iSCSI** 共享儲存裝置

- 在以上的所有虛擬環境中，fence_scsi 和 iSCSI 儲存裝置可被使用來取代原生共享儲存裝置和原生隔離裝置。

- 若 iSCSI 目標可正確支援 SCSI 3 持續保留（persistent reservation）和 preempt 與 abort 指令的話，fence_scsi 便可被使用來為透過 iSCSI 操作的共享儲存裝置，提供 I/O 隔離。請與您的儲存裝置廠商進行確認，以判定您的 iSCSI 解決方案是否支援以上功能。

- RHEL 所包含的 iSCSI 伺服器軟體不支援 SCSI 3 持續保留，因此它無法與 fence_scsi 搭配使用。然而，它適合被使用來與其它像是 fence_vmware 或是 fence_rhevm 之類的隔離裝置結合，作為共享儲存裝置解決方案。

- 若在所有客座端上使用了 fence_scsi，主機叢集便是非必要的（在 RHEL 5 Xen/KVM 和 RHEL 6 KVM Host 的使用案例中）

- 若使用了 fence_scsi 作為隔離代理程式，所有共享儲存裝置皆必須透過 iSCSI 操作。您不允許混合使用 iSCSI 和原生共享儲存裝置。

7.2.2. 一般建議

- 如以上所述，建議您在使用虛擬化功能之前，先將主機和客座端的 RHEL 套件升級為最新的版本，因為在這之後推出了許多補強功能和錯誤修正。

- 目前不支援在客座叢集下混合使用虛擬平台（hypervisor）。所有基礎主機皆必須使用相同的虛擬技術。

- 不支援在單一實體主機上執行客座叢集中的所有客座端，因為當發生單點失效的情況時，這將不會提供 high availability。然而，此配置可被用來作為原型或是開發用途。

- 最佳做法包含：

- 並非所有客座端皆必須擁有單一主機，不過此配置會提供最高等級的可用性，因為主機失效僅會影響叢集中的單一節點。若您擁有 2 比 1 的映對（各個實體主機的單一叢集中有兩個客座端），這代表單一主機失效將會造成兩個客座端失效。因此建議盡可能使用 1 比 1 的映射。

- 目前當使用 fence_xvm/fence_xvmd 或是 fence_virt/fence_virtd 隔離代理程式時，尚不支援在相同的實體主機集上，混合使用多個獨立的客座叢集。

- 若使用 fence_scsi + iSCSI 儲存裝置或是 fence_vmware + VMware（ESX/ESXi 和 vCenter）的話，便能在相同的實體主機集上混合使用多個獨立客座叢集。

- 目前已支援在相同實體主機集上，將非叢集的客座端作為客座叢集執行，不過因為若配置了主機叢集的話，主機便會實際地互相隔離，因此當主機隔離作業進行時，這些其它客座端也會被終止。

- 佈建主機硬體時，應避免過量使用（overcommit）記憶體或虛擬 CPU。過量使用記憶體或虛擬 CPU 會造成效能降低。若效能降低情況過於嚴重，叢集活動訊號（heartbeat）可能會受到影響並造成叢集失效。

附錄 **A.** 修訂記錄

修訂 1-15.1 **Mon Feb 16 2015** **Chester Cheng**
翻譯完成。

修訂 1-15 **Tue Dec 16 2014** **Steven Levine**
更新以符合 RHEL 6 歡迎頁的排序功能。

修訂 1-13 **Wed Oct 8 2014** **Steven Levine**
Red Hat Enterprise Linux 6.6 GA 發行版

修訂 1-12 **Thu Aug 7 2014** **Steven Levine**
Red Hat Enterprise Linux 6.6 Beta 發行版

修訂 1-11 **Fri Aug 1 2014** **Steven Levine**
解決：#852720
小幅修正編輯上的問題

修訂 1-10 **Fri Jun 6 2014** **Steven Levine**
發行 Red Hat Enterprise Linux 6.6 草稿

修訂 1-7 **Wed Nov 20 2013** **John Ha**
Red Hat Enterprise Linux 6.5 GA 發行版

修訂 1-4 **Mon Feb 18 2013** **John Ha**
Red Hat Enterprise Linux 6.4 GA 發行版

修訂 1-3 **Mon Jun 18 2012** **John Ha**
Red Hat Enterprise Linux 6.3 GA 發行版

修訂 1-2 **Fri Aug 26 2011** **John Ha**
6.2 發行版的更新

修訂 1-1 **Wed Nov 10 2010** **Paul Kennedy**
初始發行版