



# Red Hat Enterprise Linux (RHEL) 6

## 透過 Pacemaker 來配置 Red Hat High Availability Add-On

Red Hat Enterprise Linux 6 的 High Availability Add-On 參照文件

版 1



# Red Hat Enterprise Linux (RHEL) 6 透過 Pacemaker 來配置 Red Hat High Availability Add-On

---

Red Hat Enterprise Linux 6 的 High Availability Add-On 參照文件  
版 1

## 法律聲明

Copyright © 2015 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 摘要

《透過 Pacemaker 來配置 Red Hat High Availability Add-On》提供了使用 Pacemaker 來配置 Red Hat High Availability Add-On 的相關資訊。

## 內容目錄

<b>簡介</b> .....	<b>4</b>
1. 意見	4
<b>章 1. RED HAT HIGH AVAILABILITY ADD-ON 配置與管理參照總覽</b> .....	<b>6</b>
1.1. 安裝 PACEMAKER 配置工具	6
1.2. 配置 IPTABLES 防火牆，以允許使用叢集元件	6
1.3. 叢集與 PACEMAKER 配置檔案	6
<b>章 2. PCS 指令列介面</b> .....	<b>8</b>
2.1. PCS 指令	8
2.2. PCS 使用方法說明畫面	8
2.3. 檢視原生叢集配置	9
2.4. 將配置變更儲存至一個檔案中	9
2.5. 顯示狀態	9
2.6. 顯示完整的叢集配置	10
2.7. 顯示目前的 PCS 版本	10
<b>章 3. 叢集建立和管理</b> .....	<b>11</b>
3.1. 建立叢集	11
3.1.1. 啟用 pcsd daemon	11
3.1.2. 認證叢集節點	11
3.1.3. 配置和啟用叢集節點	11
3.2. 管理叢集節點	12
3.2.1. 停止叢集服務	12
3.2.2. 啟用和停用叢集服務	12
3.2.3. 新增叢集節點	13
3.2.4. 移除叢集節點	14
3.2.5. 預備模式	14
3.3. 設定使用者權限	14
3.4. 移除叢集配置	16
3.5. 顯示叢集狀態	16
<b>章 4. 隔離：配置 STONITH</b> .....	<b>17</b>
4.1. 可用的 STONITH（隔離）代理程式	17
4.2. 隔離裝置的一般內容	17
4.3. 顯示所有裝置特屬的隔離選項	18
4.4. 建立隔離裝置	18
4.5. 為基於儲存裝置的隔離裝置配置反隔離	19
4.6. 顯示隔離裝置	19
4.7. 修改和刪除隔離裝置	19
4.8. 透過隔離裝置管理節點	19
4.9. 額外的隔離配置選項	20
4.10. 配置隔離等級	22
<b>章 5. 配置叢集資源</b> .....	<b>24</b>
5.1. 資源建立	24
5.2. 資源內容	24
5.3. 資源特屬的參數	25
5.4. 資源 META 選項	25
5.5. 資源作業	28
5.6. 顯示已配置的資源	30
5.7. 修改資源參數	31
5.8. 多重監控作業	31

5.9. 啓用和停用叢集資源	31
5.10. 叢集資源清除	32
<b>章 6. 資源限制式</b>	<b>33</b>
6.1. 位置限制式	33
6.1.1. 配置一個「Opt-In」 Cluster	34
6.1.2. 配置一個「Opt-Out」 Cluster	34
6.2. 順序限制	34
6.2.1. 強制順序	35
6.2.2. 諮詢順序	35
6.2.3. 已排序的資源集	36
6.2.4. 將資源從順序限制中移除	36
6.3. 資源的 COLOCATION	36
6.3.1. 強制安置 (Mandatory Placement)	37
6.3.2. 諮詢安置 (Advisory Placement)	37
6.3.3. Colocate 一組資源	37
6.3.4. 移除 Colocation 限制	37
6.4. 顯示限制	37
6.5. 資源群組	38
6.5.1. 群組選項	39
6.5.2. 群組綁定	39
<b>章 7. 管理叢集資源</b>	<b>40</b>
7.1. 在叢集之間手動移動資源	40
7.2. 在發生錯誤時移動資源	40
7.3. 失去連線時移動資源	41
7.4. 啓用、停用和禁止叢集資源	42
7.5. 停用監控作業	42
7.6. 資源管理	43
<b>章 8. 進階資源類型</b>	<b>44</b>
8.1. 資源複製	44
8.1.1. 建立和移除一項複製的資源	44
8.1.2. 複製限制式	45
8.1.3. 複本綁定	46
8.2. 多狀態資源：含有多重模式的資源	46
8.2.1. 監控多狀態資源	47
8.2.2. 多狀態限制式	47
8.2.3. 多狀態綁定	47
8.3. 監控資源的事件通知	48
8.4. PACEMAKER_REMOTE 服務	48
8.4.1. 容器遠端節點資源選項	49
8.4.2. 主機與客座認證	50
8.4.3. 更改預設的 pacemaker_remote 選項	50
8.4.4. 配置總覽：KVM 遠端節點	50
<b>章 9. PACEMAKER 規則</b>	<b>52</b>
9.1. 節點屬性表示式	52
9.2. 基於時間/日期的表示式	53
9.3. 日期規格	53
9.4. 期間	54
9.5. 透過 PCS 配置規則	54
9.6. 基於時間的表示式 (範本)	54
9.7. 使用規則來判斷資源位置	55

---

<b>章 10. PACEMAKER 叢集內容</b> .....	<b>56</b>
10.1. 叢集內容與選項的概要	56
10.2. 設定與移除叢集內容	58
10.3. 查詢叢集內容設定	58
<b>附錄 A. RED HAT ENTERPRISE LINUX RELEASE 6.5 與 RED HAT ENTERPRISE LINUX RELEASE 6.6 (和更新版本) 的叢集建立</b> .....	<b>60</b>
A.1. 透過 RGMANAGER 和 PACEMAKER 建立叢集	60
A.2. 在 RED HAT ENTERPRISE LINUX RELEASE 6.5 與 RED HAT ENTERPRISE LINUX RELEASE 6.6 (和更新版本) 中透過 PACEMAKER 建立叢集	63
<b>附錄 B. 透過 PCS 指令進行配置的範例</b> .....	<b>65</b>
B.1. 初始系統設定	65
B.1.1. 安裝叢集軟體	65
B.1.2. 建立和啓用叢集	65
B.2. 隔離配置	66
B.3. 透過 PCS 指令來在一個 RED HAT HIGH AVAILABILITY CLUSTER 中配置 APACHE WEB SERVER	67
B.3.1. 為一個 LVM 卷冊配置 ext4 檔案系統	68
B.3.2. 網站伺服器配置	69
B.3.3. 獨佔啓用 (exclusive activation) 叢集中的卷冊群組	70
B.3.4. 透過 pcs 指令來建立資源和資源群組	71
B.3.5. 測試資源配置	73
<b>附錄 C. 修訂記錄</b> .....	<b>74</b>

## 簡介

本文件提供了有關於安裝、配置，和管理 Red Hat High Availability 外掛程式元件上的相關資訊。Red Hat High Availability 外掛程式元件能讓您連至電腦群組（亦稱為節點或成員），以作為叢集共同運作。在本文件中，我們使用了 *cluster*（叢集）和 *clusters* 來表示執行 Red Hat High Availability 外掛程式的電腦群組。

本文件的目標讀者應擁有 Red Hat Enterprise Linux 的進階知識，以及叢集、儲存裝置，和伺服器運算上的概念理解。

欲取得更多有關於 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、配置與管理上的相關資訊。

欲取得更多有關於 Red Hat Enterprise Linux 6 的 High Availability 外掛程式以及相關產品上的資訊，請參閱下列資源：

- *High Availability 外掛程式總覽* — 提供了 Red Hat High Availability 外掛程式的高層總覽。
- *叢集管理* — 提供了關於安裝、配置、與管理 High Availability 外掛程式的相關資訊。
- 《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。
- 【Global File System 2 : 配置與管理】— 提供了有關於安裝、配置和維護 Red Hat GFS2（Red Hat Global File System 2）的相關資訊，它包含於 Resilient Storage 外掛程式中。
- 【DM Multipath】— 提供了有關於 Red Hat Enterprise Linux 6 的 Device-Mapper Multipath 功能使用上的相關資訊。
- *負載平衡器管理* — 提供了有關於透過 Red Hat Load Balancer 外掛程式配置高效能系統和服務上的相關資訊，一組提供了 Linux 虛擬伺服器（LVS），以在多台真實伺服器之間，平衡 IP 負載的整合軟體元件。
- 《發行公告》— 提供了有關於最新 Red Hat 產品的相關資訊。

Red Hat 文件擁有 HTML、PDF 以及 Red Hat Enterprise Linux 文件光碟上的 RPM 版本，以及位於 <https://access.redhat.com/site/documentation/> 的網站版本。

## 1. 意見

若您發現了錯字，或有改善本指南的意見，我們很樂意與您取得聯繫。請在 Bugzilla（<http://bugzilla.redhat.com/bugzilla/>）中提交相關意見。請針對於「Red Hat Enterprise Linux 6」產品和「doc-Cluster\_General」元件提交錯誤報告。

請確認您提供了指南識別子：

Configuring\_High\_Availability\_With\_Pacemaker(EN)-6 (2015-7-8T16:26)



■

透過指南的識別子，我們將能清楚地知道您所擁有的版本為何。

若您有任何改善文件的意見，請盡可能地進行詳述。若您發現了錯誤，請提供章節號碼以及附近文字，以供我們更容易地找出錯誤所在。

# 章 1. RED HAT HIGH AVAILABILITY ADD-ON 配置與管理參照總覽

本文件提供了使用 Pacemaker 的 Red Hat High Availability Add-On 之選項與功能的描述。

本指南記載了 Red Hat Enterprise Linux Release 6.6 與更新版本的 **pcs** 配置介面的使用方法。



## 注意

欲知如何透過 High Availability 外掛和 Red Hat GFS2 來建置及升級 RHEL 叢集上的最佳方式，請參閱 Red Hat 客戶端入口網站上的《Red Hat Enterprise Linux Cluster, High Availability, and GFS Deployment Best Practices》文章（位於 <https://access.redhat.com/kb/docs/DOC-40821>）。

## 1.1. 安裝 PACEMAKER 配置工具

您可使用下列 **yum install** 指令來安裝 Red Hat High Availability Add-On 軟體套件，以及來自於 High Availability 頻道的所有可用隔離代理程式。

```
# yum install pcs fence-agents
```

**lvm2-cluster** 和 **gfs2-utils** 套件屬於 ResilientStorage 頻道的一部分。您可視需求透過下列指令安裝它們。

```
# yum install lvm2-cluster gfs2-utils
```



## 警告

在您安裝了 Red Hat High Availability 外掛套件後，您應確認您已完成您的軟體更新偏好設定，以確保不會有軟體被自動安裝。在運作中的叢集上進行安裝可能會造成非預期的行為發生。

## 1.2. 配置 IPTABLES 防火牆，以允許使用叢集元件

Red Hat High Availability Add-On 需要啓用下列連接埠：

- TCP：連接埠 2224、3121、21064
- UDP：連接埠 5405

## 1.3. 叢集與 PACEMAKER 配置檔案

Red Hat High Availability 外掛的配置檔案為 **cluster.conf** 與 **cib.xml**。請勿直接編輯這些檔案；請使用 **pcs** 或 **pcsd** 介面來代替。

**cluster.conf** 檔案提供了 **corosync**（Pacemaker 所建置於的叢集管理程式）所使用的叢集參數。

**cib.xml** 檔案乃一個代表叢集配置與目前叢集中所有資源狀態的 XML 檔案。此檔案會被 Pacemaker 的叢集資訊基礎 (Cluster Information Base, CIB) 所使用。CIB 的內容會自動在所有叢集之間同步化。

## 章 2. PCS 指令列介面

**pcs** 指令列介面提供了用來控制與配置 **corosync** 和 **pacemaker** 的能力。

**pcs** 指令的一般格式如下。

```
pcs [-f file] [-h] [commands]...
```

### 2.1. PCS 指令

**pcs** 指令如下。

- **cluster**

配置叢集選項與節點。欲取得 **pcs cluster** 指令上的相關資訊，請參閱 [〈章 3, 叢集建立和管理〉](#)。

- **resource**

建立與管理叢集資源。欲取得 **pcs cluster** 指令上的相關資訊，請參閱 [〈章 5, 配置叢集資源〉](#)、[〈章 7, 管理叢集資源〉](#) 和 [〈章 8, 進階資源類型〉](#)。

- **stonith**

配置隔離裝置以搭配 Pacemaker 使用。欲取得 **pcs stonith** 指令上的相關資訊，請參閱 [〈章 4, 隔離：配置 STONITH〉](#)。

- **constraint**

管理資源限制。欲取得 **pcs constraint** 指令上的相關資訊，請參閱 [〈章 6, 資源限制式〉](#)。

- **property**

設置 Pacemaker 內容。欲取得透過 **pcs property** 指令來設定內容的相關資訊，請參閱 [〈章 10, Pacemaker 叢集內容〉](#)。

- **status**

檢視目前的叢集與資源狀態。欲取得 **pcs status** 指令上的相關資訊，請參閱 [〈節 2.5, “顯示狀態”〉](#)。

- **config**

以易於閱讀的格式顯示完整的叢集配置。欲取得 **pcs config** 指令上的相關資訊，請參閱 [〈節 2.6, “顯示完整的叢集配置”〉](#)。

### 2.2. PCS 使用方法說明畫面

您可使用 **pcs** 的 **-h** 選項來顯示 **pcs** 指令的參數，以及這些參數的詳述。比方說，下列指令會顯示 **pcs resource** 指令的參數。在此僅顯示了一部分輸出。

```
# pcs resource -h
Usage: pcs resource [commands]...
Manage pacemaker resources
```

```

Commands:
  show [resource id] [--all]
    Show all currently configured resources or if a resource is
  specified
    show the options for the configured resource.  If --all is
  specified
    resource options will be displayed

  start <resource id>
    Start resource specified by resource_id
  ...

```

## 2.3. 檢視原生叢集配置

儘管您不應直接編輯叢集配置檔案，您可透過 `pcs cluster cib` 指令來檢視原生叢集配置。

您能透過 `pcs cluster cib filename cib` 來將原生叢集配置儲存至一個指定的檔案中，如〈節 2.4, “將配置變更儲存至一個檔案中”〉中所述。

## 2.4. 將配置變更儲存至一個檔案中

當使用 `pcs` 指令時，您可使用 `-f` 選項以在不影響運作中的 CIB 的情況下，將配置變更儲存至一個檔案中。

若您先前配置了一項叢集，並且早已有一項運作中的 CIB，請使用下列指令來將原生的 xml 儲存至一個檔案。

```
pcs cluster cib filename
```

比方說，下列指令會將來自於 CIB 的原生 xml 儲存入一個名為 `testfile` 的檔案中。

```
pcs cluster cib testfile
```

下列指令會在 `testfile1` 這個檔案中建立資源，但卻不會將該資源加入目前執行中的叢集配置。

```
# pcs -f testfile1 resource create VirtualIP ocf:heartbeat:IPaddr2
ip=192.168.0.120 cidr_netmask=24 op monitor interval=30s
```

您可透過以下指令將 `testfile` 推送至 CIB 中。

```
pcs cluster cib-push filename
```

## 2.5. 顯示狀態

您能以下列指令顯示叢集與叢集資源的狀態。

```
pcs status commands
```

若您不指定一項 `commands` 參數，這項指令便會顯示有關於叢集與資源的所有資源。您可藉由指定 `resources`、`groups`、`cluster`、`nodes` 或 `pcsd` 來只顯示特定的叢集元件。

## 2.6. 顯示完整的叢集配置

使用下列指令來顯示目前的完整叢集配置。

```
pcs config
```

## 2.7. 顯示目前的 PCS 版本

下列指令能顯示目前正在執行中的 **pcs** 版本。

```
pcs --version
```

## 章 3. 叢集建立和管理

本章詳述了如何透過 Pacemaker 來進行基本的叢集管理，包括建立叢集、管理叢集元件和顯示叢集狀態。

### 3.1. 建立叢集

若要建立一個叢集，請執行下列步驟：

1. 在叢集中的各個節點上啟用 **pcsd**。
2. 為構成叢集的節點進行認證。
3. 配置和同步叢集節點。
4. 在叢集節點上啟用叢集服務。

下列部分詳述了您使用來執行這些步驟的指令。

#### 3.1.1. 啟用 pcsd daemon

下列指令會啟用 **pcsd** 服務並在系統啟動時啟用 **pcsd**。這些指令應在叢集中的各個節點上執行。

```
# service pcsd start  
# chkconfig pcsd on
```

#### 3.1.2. 認證叢集節點

下列指令會在叢集中的節點上向 **pcs daemon** 認證 **pcs**。

- 所有節點上的 **pcs** 管理員使用者名稱皆必須是 **hacluster**。建議在各個節點上皆使用相同的 **hacluster** 使用者密碼。
- 若您不指定使用者名稱或密碼，系統將會在您執行指令時，提示您為各個節點輸入這些參數。
- 若您不指定任何節點的話，這項指令便會在透過 **pcs cluster setup** 指令指定的節點上認證 **pcs**（若您先前已執行了這項指令）。

```
pcs cluster auth [node] [...] [-u username] [-p password]
```

比方說，下列指令會在叢集中包含了 **z1.example.com** 和 **z2.example.com** 的兩個節點認證 **z1.example.com** 上的使用者 **hacluster**。這項指令會提示叢集節點上的使用者 **hacluster** 的密碼。

```
root@z1 ~]# pcs cluster auth z1.example.com z2.example.com  
Username: hacluster  
Password:  
z1.example.com: Authorized  
z2.example.com: Authorized
```

認證權杖處存在 **~/.pcs/tokens** 檔案（或 **/var/lib/pcsd/tokens**）中。

#### 3.1.3. 配置和啟用叢集節點

下列指令能配置叢集配置檔案並將配置同步至指定的節點上。

- 若您指定了 **--start** 選項，這項指令也能在指定的節點上啓用叢集服務，您亦可透過獨立的 **pcs cluster start** 指令來啓用叢集服務。
- 若您指定了 **--local** 選項，這項指令便僅會在本機節點上進行變更。

```
pcs cluster setup [--start] [--local] --name cluster_name node1 [node2]
[...]
```

下列指令會在指定的節點上啓用叢集服務。

- 若您指定了 **--all** 選項，這項指令便會在所有節點上啓用叢集服務。
- 若您不指定任何節點，叢集服務便僅會在本機節點上啓用。

```
pcs cluster start [--all] [node] [...]
```

## 3.2. 管理叢集節點

下列部分詳述了您可使用來管理叢集節點的指令，包括用來啓用和停用叢集服務以及用來新增和移除叢集節點的指令。

### 3.2.1. 停止叢集服務

下列指令能停用指定節點上的叢集服務。如同 **pcs cluster start**，**--all** 選項能停止所有節點上的叢集服務，而若您不指定任何節點的話，叢集服務僅會在本機節點上停用。

```
pcs cluster stop [--all] [node] [...]
```

您可透過以下指令來強制停用本機節點上的叢集服務，它會進行一項 **kill -9** 指令。

```
pcs cluster kill
```

### 3.2.2. 啓用和停用叢集服務

使用下列指令來配置叢集服務，使其能在指定節點啓動時執行。

- 若您指定了 **--all** 選項，這項指令便會在所有節點上啓用叢集服務。
- 若您不指定任何節點，叢集服務便僅會在本機節點上啓用。

```
pcs cluster enable [--all] [node] [...]
```

使用下列指令來配置叢集服務，使其在指定節點啓動時不執行。

- 若您指定了 **--all** 選項，這項指令便會在所有節點上停用叢集服務。
- 若您不指定任何節點，叢集服務便僅會在本機節點上停用。

```
pcs cluster disable [--all] [node] [...]
```



### 3.2.3. 新增叢集節點

使用下列程序來新增節點至既有的叢集中。在此範例中，既有的叢集節點為 **clusternode-01.example.com**、**clusternode-02.example.com** 以及 **clusternode-03.example.com**。新的節點為 **newnode.example.com**。

在需加入至叢集的新節點上執行下列任務。

1. 安裝叢集套件；

```
[root@newnode ~]# yum install -y pacemaker cman pcs
```

2. 若要避免 **corosync** 在無 **cman** 的情況下啓用，請執行以下指令：

```
[root@newnode ~]# chkconfig corosync off
```

3. 為使用者 ID **hacluster** 設置一組密碼。建議您使用與叢集中各個節點相同的密碼。

```
[root@newnode ~]# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

4. 在新節點上啓用 **pcsd**：

```
[root@newnode ~]# service pcsd start
[root@newnode ~]# chkconfig pcsd on
```

在既有叢集中的節點上執行下列任務。

1. 在新的叢集節點上認證使用者 **hacluster**。

```
[root@clusternode-01 ~]# pcs cluster auth newnode.example.com
Username: hacluster
Password:
newnode.example.com: Authorized
```

2. 將新節點加入至既有的叢集中。這項指令亦能使叢集配置檔案 **cluster.conf** 與叢集中的所有節點同步，包括您要新增的節點。

```
[root@clusternode-01 ~]# pcs cluster node add newnode.example.com
clusternode-01.example.com: Corosync updated
clusternode-02.example.com: Corosync updated
clusternode-03.example.com: Corosync updated
newnode.example.com: Updated cluster.conf...
newnode.example.com: Starting Cluster...
```

在需加入至叢集的新節點上執行下列任務。

1. 在新的節點上，為叢集中所有節點認證使用者 **hacluster**。

```
[root@newnode ~]# pcs cluster auth
Username: hacluster
Password:
clusternode-01.example.com: Authorized
clusternode-02.example.com: Authorized
clusternode-03.example.com: Authorized
newnode.example.com: Already authorized
```

2. 在新節點上啓用叢集服務。

```
[root@newnode ~]# pcs cluster start
Starting Cluster...
[root@newnode ~]# pcs cluster enable
```

3. 確認您為新的叢集節點配置了隔離裝置。欲取得配置隔離裝置上的相關資訊，請參閱 [〈章 4, 隔離：配置 STONITH〉](#)。

### 3.2.4. 移除叢集節點

下列指令會關閉指定的節點並將其由叢集中所有節點上的叢集配置檔案 `cluster.conf` 中移除。欲取得更多有關於將叢集節點上所有叢集相關資料完全移除，並永久銷毀叢集的相關資訊，請參閱 [〈節 3.4, “移除叢集配置”〉](#)。

```
pcs cluster node remove node
```

### 3.2.5. 預備模式

下列指令會使指定的節點進入預備模式。指定的節點將不再能提供資源。任何在該節點上使用中的資源將會被移至另一個節點上。若您指定了 `--all`，這項指令便會使所有節點進入預備模式。

您可在更新資源的套件時使用這項指令。您亦可在測試配置時使用這項指令，以在無需實際關閉節點的情況下模擬復原。

```
pcs cluster standby node | --all
```

以下指令會將指定的節點從預備模式中移出。在執行了這項指令後，指定的節點便能繼續提供資源。若您指定了 `--all`，此選項便會將所有節點從預備模式中移出。

```
pcs cluster unstandby node | --all
```

請注意，當您執行 `pcs cluster standby` 指令時，這會將限制式加入至資源中以防它們在顯示的節點上執行。當您執行 `pcs cluster unstandby` 指令時，這將會移除限制式。這並不一定會將資源移回顯示的節點上；在此時資源能在哪裡執行取決於您一開始如何配置了您的資源。欲取得更多有關於資源限制式上的相關資訊，請參閱 [〈章 6, 資源限制式〉](#)。

## 3.3. 設定使用者權限

就預設值，`root` 使用者與任何屬於 `haclient` 成員的使用者，皆擁有叢集配置的完整讀取/寫入權限。由 Red Hat Enterprise Linux 6.6 起，您可使用 `pcs acl` 指令來為本機使用者設置權限，以透過存取控制清單 (ACL) 來將叢集配置權限設置為唯讀或可讀寫。

為本機使用者設定權限包含兩個步驟：

1. 執行 `pcs acl role create...` 指令以建立一個角色，並為該角色定義權限。
2. 透過 `pcs acl user create` 指令來為使用者指定您所建立的角色。

下列範例程序將為名為 **rouser** 的本機使用者提供叢集配置的唯讀權限。

1. 這項程序需要在本機系統上有一位 **rouser** 使用者存在，並且 **rouser** 使用者必須屬於 **haclient** 群組的成員。

```
# adduser rouser
# usermod -a -G haclient rouser
```

2. 以 **enable-acl** 叢集內容來啟用 Pacemaker ACL。

```
# pcs property set enable-acl=true --force
```

3. 建立一個名為 **read-only** 的角色，並持有 `cib` 的唯讀權限。

```
# pcs acl role create read-only description="Read access to cluster"
read xpath /cib
```

4. 在 pcs ACL 系統中建立 **rouser** 使用者，並為該使用者指定 **read-only** 角色。

```
# pcs acl user create rouser read-only
```

5. 檢視目前的 ACL。

```
# pcs acl
User: rouser
Roles: read-only
Role: read-only
Description: Read access to cluster
Permission: read xpath /cib (read-only-read)
```

下列範例程序將為名為 **wuser** 的本機使用者提供叢集配置的寫入權限。

1. 這項程序需要使用者 **wuser** 存在本機系統上，而 **wuser** 使用者必須是 **haclient** 群組的成員。

```
# adduser wuser
# usermod -a -G haclient wuser
```

2. 以 **enable-acl** 叢集內容來啟用 Pacemaker ACL。

```
# pcs property set enable-acl=true --force
```

3. 建立一個名為 **write-access** 的角色，並且該角色持有 `cib` 的寫入權限。

```
# pcs acl role create write-access description="Full access" write
xpath /cib
```

4. 在 pcs ACL 系統中建立 **wuser** 使用者，並為該使用者指定 **write-access** 角色。

```
# pcs acl user create wuser write-access
```

5. 檢視目前的 ACL。

```
# pcs acl
User: rouser
  Roles: read-only
User: wuser
  Roles: write-access
Role: read-only
  Description: Read access to cluster
  Permission: read xpath /cib (read-only-read)
Role: write-access
  Description: Full Access
  Permission: write xpath /cib (write-access-write)
```

欲取得更多有關於叢集 ACL 的相關資訊，請參閱 **pcs acl** 指令的說明畫面。

### 3.4. 移除叢集配置

若要移除所有叢集配置檔案並停用所有叢集服務（而永久性銷毀叢集），請使用下列指令。



#### 警告

這項指令會永久性移除任何已建立的叢集配置。建議您在銷毀叢集之前，先執行 **pcs cluster stop**。

```
pcs cluster destroy
```

### 3.5. 顯示叢集狀態

下列指令將顯示叢集與叢集資源的目前狀態。

```
pcs status
```

您可透過下列指令來顯示有關於叢集目前狀態的其中一部分資訊。

下列指令將顯示叢集的狀態，而非叢集資源的狀態。

```
pcs cluster status
```

下列指令將顯示叢集資源的狀態。

```
pcs status resources
```

## 章 4. 隔離：配置 STONITH

STONITH 乃 Shoot-The-Other-Node-In-The-Head 的簡稱，並且它能保護您的資料使其不會因為有害的節點或是同時存取而遭到損毀。

節點無反應並不代表它沒有在存取您的資料。若要 100% 確認您資料安全的話，您需要使用 STONITH 來隔離節點，以確認資料在被其它節點存取之前，目前的節點已確實離線。

STONITH 也能使用於叢集服務無法停下時。在此情況下，叢集能使用 STONITH 來強制整個節點離線，並讓服務能在其它位置上安全啓用。

### 4.1. 可用的 STONITH（隔離）代理程式

使用下列指令來檢視一列所有可用的 STONITH 代理程式。您需指定一項篩選器，然後這項指令便僅會顯示符合該篩選器的 STONITH 代理程式。

```
pcs stonith list [filter]
```

### 4.2. 隔離裝置的一般內容



#### 注意

若要停用隔離裝置/資源，您可如正常資源一般設置 **target-role**。



#### 注意

若要避免節點使用一項隔離裝置，您能為隔離資源配置位置限制式。

表格 4.1, “隔離裝置的一般內容” 詳述了您能為隔離裝置設置的一般內容。欲取得有關您能為特定隔離裝置設置的隔離內容的相關資訊，請參閱〈節 4.3, “顯示所有裝置特屬的隔離選項”〉。



#### 注意

欲取得進階的隔離配置內容，請參閱〈節 4.9, “額外的隔離配置選項”〉

表格 4.1. 隔離裝置的一般內容

欄位	類型	預設	詳述
<b>priority</b>	整數	0	Stonith 資源的優先順序。裝置會以優先順序最高到最低來被測試。
<b>pcmk_host_map</b>	string		若裝置不支援主機名稱，提與主機名稱對映的連接埠。比方說： <b>node1:1;node2:2,3</b> 會告知叢集使用 port 1 來對映 node1，以及 ports 2 和 3 來對映 node2
<b>pcmk_host_list</b>	string		一列受到此裝置控制的機器（選用性，除非 <b>pcmk_host_check=static-list</b> ）。

欄位	類型	預設	詳述
<code>pcmk_host_check</code>	string	dynamic-list	如何判斷哪些機器是由特定裝置所控制。允許的值有： <b>dynamic-list</b> （查詢裝置）、 <b>static-list</b> （檢查 <code>pcmk_host_list</code> 屬性）、 <b>none</b> （假設所有裝置皆能隔離所有機器）

### 4.3. 顯示所有裝置特屬的隔離選項

使用下列指令來檢視指定的 STONITH 代理程式的選項。

```
pcs stonith describe stonith_agent
```

比方說，下列指令會顯示 telnet/SSH 上的 APC 之選項的隔離代理程式。

```
# pcs stonith describe fence_apc
Stonith options for: fence_apc
  ipaddr (required): IP Address or Hostname
  login (required): Login Name
  passwd: Login password or passphrase
  passwd_script: Script to retrieve password
  cmd_prompt: Force command prompt
  secure: SSH connection
  port (required): Physical plug number or name of virtual machine
  identity_file: Identity file for ssh
  switch: Physical switch number on device
  inet4_only: Forces agent to use IPv4 addresses only
  inet6_only: Forces agent to use IPv6 addresses only
  ipport: TCP port to use for connection with device
  action (required): Fencing Action
  verbose: Verbose mode
  debug: Write debug information to given file
  version: Display version information and exit
  help: Display help and exit
  separator: Separator for CSV created by operation list
  power_timeout: Test X seconds for status change after ON/OFF
  shell_timeout: Wait X seconds for cmd prompt after issuing command
  login_timeout: Wait X seconds for cmd prompt after login
  power_wait: Wait X seconds after issuing ON/OFF
  delay: Wait X seconds before fencing is started
  retry_on: Count of attempts to retry power on
```

### 4.4. 建立隔離裝置

下列指令會建立一項 stonith 裝置。

```
pcs stonith create stonith_id stonith_device_type [stonith_device_options]
```

```
# pcs stonith create MyStonith fence_virt pcmk_host_list=f1 op monitor
interval=30s
```

若您在多個節點之間使用單一隔離裝置，並且各個節點皆使用不同連接埠的話，您無須為各個節點個別建立一項裝置。您可使用 `pcmk_host_map` 選項來定義哪個連接埠連接哪個節點來代替。比方說，下列指令會建立單一隔離裝置名為 `myapc-west-13`，該隔離裝置使用了一個名為 `west-apc` 的 APC powerswitch，並且使用了連接埠 15 來連至 `west-13`。

```
# pcs stonith create myapc-west-13 fence_apc pcmk_host_list="west-13"
ipaddr="west-apc" login="apc" passwd="apc" port="15"
```

然而下列範例使用了名為 `west-apc` 的 APC powerswitch 來隔離使用連接埠 15 的 `west-13` 節點、使用連接埠 17 的 `west-14` 節點、使用連接埠 18 的 `west-15` 節點，以及使用連接埠 19 的 `west-16` 節點。

```
# pcs stonith create myapc fence_apc pcmk_host_list="west-13,west-14,west-
15,west-16" pcmk_host_map="west-13:15;west-14:17;west-15:18;west-16:19"
ipaddr="west-apc" login="apc" passwd="apc"
```

## 4.5. 為基於儲存裝置的隔離裝置配置反隔離

當建立 SAN/儲存隔離裝置（也就是使用非基於電源隔離代理程式的隔離裝置）時，您必須在建立 `stonith` 裝置時設置 `meta` 選項 `provides=unfencing`。這能確保已隔離的節點在重新啟動前能成為非隔離狀態，並且叢集服務能在該節點上啟用。

當配置一項基於電源的隔離裝置時，設置 `provides=unfencing meta` 選項為非必要的，因為裝置本身會為節點提供電源以讓它啟動（並嘗試重新加入叢集）。在此情況下的啟動動作代表反隔離已發生。

下列指令會配置一項名為 `my-scsi-shooter` 的 `stonith` 裝置，該裝置使用 `fence_scsi` 隔離代理程式來將裝置反隔離。

```
pcs stonith create my-scsi-shooter fence_scsi devices=/dev/sda meta
provides=unfencing
```

## 4.6. 顯示隔離裝置

下列指令會顯示所有目前已配置的隔離裝置。若指定 `stonith_id`，這項指令僅會顯示該 `stonith` 裝置的選項。若指定 `--full` 選項，所有已配置的 `stonith` 選項皆會顯示。

```
pcs stonith show [stonith_id] [--full]
```

## 4.7. 修改和刪除隔離裝置

使用下列指令來修改或新增選項至目前已配置的隔離裝置。

```
pcs stonith update stonith_id [stonith_device_options]
```

使用下列指令來將一項隔離裝置由目前配置中移除。

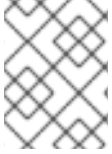
```
pcs stonith delete stonith_id
```

## 4.8. 透過隔離裝置管理節點

您能以下列指令手動隔離節點。若您指定 `--off` 選項的話，這項指令將會使用 `off` API 調用來執行 `stonith`，這會將節點關閉，而不會將它重新啓動。

```
pcs stonith fence node [--off]
```

您可透過以下指令來確認指定節點的電源目前是否已關閉。



### 注意

若您所指定的節點依然在執行叢集軟體或一般由叢集所控制的服務，資料損毀/叢集錯誤的情況可能會發生。

```
pcs stonith confirm node
```

## 4.9. 額外的隔離配置選項

表格 4.2, “隔離裝置的進階內容” 概述了您能為隔離裝置設置的額外內容。請注意，這些內容僅供進階使用。

表格 4.2. 隔離裝置的進階內容

欄位	類型	預設	詳述
<code>pcmk_host_argument</code>	string	port	一個能代替 port 的額外參數。部分裝置不支援標準的 port 參數，或可能會提供額外的參數。請使用它來指定額外、裝置特屬的參數，以顯示需隔離的機器。 <code>none</code> 這個值能被使用來命令叢集不提供任何額外參數。
<code>pcmk_reboot_action</code>	string	reboot	一項可替代 <code>reboot</code> 的指令。部分裝置不支援標準指令或可能會提供額外的指令。請使用它來指定一項額外、裝置特屬的指令，以實作 <code>reboot</code> 動作。
<code>pcmk_reboot_timeout</code>	time	60s	為重新啓動動作指定一項額外的逾時值，以代替 <code>stonith-timeout</code> 。部分裝置需要比正常高出更多/更少時間來完成。請使用它來指定額外、裝置特屬的逾時值，以用於 <code>reboot</code> 動作。
<code>pcmk_reboot_retries</code>	整數	2	在逾時期間之內，可重新嘗試 <code>reboot</code> 指令的最多次數。部分裝置不支援多重連線。若裝置忙於進行另一項任務，作業可能會失敗，因此若還有剩餘時間，Pacemaker 將會自動重新嘗試進行作業。請使用此選項來更改 Pacemaker 在放棄之前，所會重新嘗試 <code>reboot</code> 動作的次數。



欄位	類型	預設	詳述
<b>pcmk_off_action</b>	string	off	一項可替代 <b>off</b> 的指令。部分裝置不支援標準指令或可能會提供額外的指令。請使用它來指定一項額外、裝置特屬的指令，以實作 off 動作。
<b>pcmk_off_timeout</b>	time	60s	為關閉動作指定一項額外的逾時值，以代替 <b>stonith-timeout</b> 。部分裝置需要比正常高出更多或更少的時間來完成。請使用它來指定額外、裝置特屬的逾時值，以用於 off 動作。
<b>pcmk_off_retries</b>	整數	2	在逾時期間之內，可重新嘗試 off 指令的最多次數。部分裝置不支援多重連線。若裝置忙於進行另一項任務，作業可能會失敗，因此若還有剩餘時間，Pacemaker 將會自動重新嘗試進行作業。請使用此選項來更改 Pacemaker 在放棄之前，所會重新嘗試 off 動作的次數。
<b>pcmk_list_action</b>	string	list	一項可替代 <b>list</b> 的指令。部分裝置不支援標準指令或可能會提供額外的指令。請使用它來指定一項額外、裝置特屬的指令，以實作 list 動作。
<b>pcmk_list_timeout</b>	time	60s	為 list 動作指定一項額外的逾時值，以代替 <b>stonith-timeout</b> 。部分裝置需要比正常高出更多或更少的時間來完成。請使用它來指定額外、裝置特屬的逾時值，以用於 list 動作。
<b>pcmk_list_retries</b>	整數	2	在逾時期間之內，可重新嘗試 <b>list</b> 指令的最多次數。部分裝置不支援多重連線。若裝置忙於進行另一項任務，作業可能會失敗，因此若還有剩餘時間，Pacemaker 將會自動重新嘗試進行作業。請使用此選項來更改 Pacemaker 在放棄之前，所會重新嘗試 list 動作的次數。
<b>pcmk_monitor_action</b>	string	monitor	一項可替代 <b>monitor</b> 的指令。部分裝置不支援標準指令或可能會提供額外的指令。請使用它來指定一項額外、裝置特屬的指令，以實作 monitor 動作。
<b>pcmk_monitor_timeou t</b>	time	60s	為 monitor 動作指定一項額外的逾時值，以代替 <b>stonith-timeout</b> 。部分裝置需要比正常高出更多或更少的時間來完成。請使用它來指定額外、裝置特屬的逾時值，以用於 monitor 動作。

欄位	類型	預設	詳述
<code>pcmk_monitor_retries</code>	整數	2	在逾時期間之內，可重新嘗試 <code>monitor</code> 指令的最多次數。部分裝置不支援多重連線。若裝置忙於進行另一項任務，作業可能會失敗，因此若還有剩餘時間，Pacemaker 將會自動重新嘗試進行作業。請使用此選項來更改 Pacemaker 在放棄之前，所會重新嘗試 <code>monitor</code> 動作的次數。
<code>pcmk_status_action</code>	string	<code>status</code>	一項可替代 <code>status</code> 的指令。部分裝置不支援標準指令或可能會提供額外的指令。請使用它來指定一項額外、裝置特屬的指令，以實作 <code>status</code> 動作。
<code>pcmk_status_timeout</code>	time	60s	為 <code>status</code> 動作指定一項額外的逾時值，以代替 <code>stonith-timeout</code> 。部分裝置需要比正常高出更多或更少的時間來完成。請使用它來指定額外、裝置特屬的逾時值，以用於 <code>status</code> 動作。
<code>pcmk_status_retries</code>	整數	2	在逾時期間之內，可重新嘗試 <code>status</code> 指令的最多次數。部分裝置不支援多重連線。若裝置忙於進行另一項任務，作業可能會失敗，因此若還有剩餘時間，Pacemaker 將會自動重新嘗試進行作業。請使用此選項來更改 Pacemaker 在放棄之前，所會重新嘗試 <code>status</code> 動作的次數。

## 4.10. 配置隔離等級

Pacemaker 可透過一項名為隔離拓撲 (fencing topology) 的功能來支援隔離含有多重裝置的節點。若要實作拓撲，請照常建立個別裝置，然後在配置中的 `fencing-topology` 部分裡定義一或更多個隔離等級。

- 各等級皆會被以由小到大的數字順序嘗試，從 1 開始。
- 若某項裝置失敗的話，目前等級的處理便會終止。該等級中的其它裝置皆不會被處理，而下個等級將會被嘗試。
- 若成功隔離了所有裝置，則該等級便已成功而無需嘗試其它等級。
- 當通過等級 (成功) 或嘗試了所有等級 (失敗) 後，作業便會完成。

請使用下列指令來新增隔離等級至節點上。裝置會以逗號區隔的 `stonith id` 提供，並在該等級的節點上被嘗試使用。

```
pcs stonith level add level node devices
```

下列指令能列出目前已配置的所有隔離等級。

```
pcs stonith level
```

在下列範例中有兩個為 **rh7-2** 節點所配置的隔離裝置：一個名為 **my\_ilo** 的 ilo 隔離裝置，以及一個名為 **my\_apc** 的 apc 隔離裝置。這些指令可設置隔離等級，因此當裝置 **my\_ilo** 失效並無法隔離節點時，Pacemaker 便會嘗試使用 **my\_apc** 裝置。此範例也顯示了等級經過配置後，**pcs stonith level** 指令的輸出。

```
# pcs stonith level add 1 rh7-2 my_ilo
# pcs stonith level add 2 rh7-2 my_apc
# pcs stonith level
Node: rh7-2
  Level 1 - my_ilo
  Level 2 - my_apc
```

下列指令會移除指定節點與裝置的隔離等級。若未指定節點或裝置，則您所指定的隔離等級將會由所有節點上移除。

```
pcs stonith level remove level [node_id] [stonith_id] ... [stonith_id]
```

下列指令會清除指定節點上，或指定 stonith id 的隔離等級。若您不指定節點或 stonith id，所有隔離等級皆會被清除。

```
pcs stonith level clear [node|stonith_id(s)]
```

若您指定了超過一組 stonith id，它們必須以逗號區隔開並且不使用任何空格，如以下範例所示。

```
# pcs stonith level clear dev_a,dev_b
```

下列指令能驗證所有指定於隔離等級中的隔離裝置與節點是否皆存在。

```
pcs stonith level verify
```

## 章 5. 配置叢集資源

本章提供了有關於叢集中配置資源上的相關資訊。

### 5.1. 資源建立

使用下列指令來建立一項叢集資源。

```
pcs resource create resource_id standard:provider:type|type [resource options]
```

例如，下列指令會建立一項名為 `VirtualIP`、標準 `ocf`、供應方為 `heartbeat`，以及類型為 `IPAddr2` 的資源。這項資源的浮動位址為 `192.168.0.120`，系統會檢查資源是否每 30 秒會運作一次。

```
# pcs resource create VirtualIP ocf:heartbeat:IPAddr2 ip=192.168.0.120
cidr_netmask=24 op monitor interval=30s
```

此外，您亦可忽略 `standard` 與 `provider` 欄位，並使用下列指令。這將會預設使用 `ocf` 為標準，`heartbeat` 為供應方。

```
# pcs resource create VirtualIP IPAddr2 ip=192.168.0.120 cidr_netmask=24
op monitor interval=30s
```

使用下列指令來刪除一項已配置的資源。

```
pcs resource delete resource_id
```

比方說，下列指令會刪除一項資源 ID 為 `VirtualIP` 的既有資源

```
# pcs resource delete VirtualIP
```

- 欲取得更多有關於 `pcs resource create` 指令的 `resource_id`、`standard`、`provider` 以及 `type` 欄位上的相關資訊，請參閱 [〈節 5.2, “資源內容”](#)。
- 欲取得更多有關於為個別資源定義資源參數上的相關資訊，請參閱 [〈節 5.3, “資源特屬的參數”](#)。
- 欲取得更多有關於定義資源 meta 選項上的相關資訊，請參閱 [〈節 5.4, “資源 Meta 選項”](#)，這些選項會被叢集使用來決定資源的運作行為。
- 欲取得定義資源上的作業的相關資訊，請參閱 [〈節 5.5, “資源作業”](#)。

### 5.2. 資源內容

您為資源所定義的內容能讓叢集知道資源應對應哪個 script、哪裏可以找到該 script，以及它符合哪項標準。 [〈表格 5.1, “資源內容”](#) 詳述了這些內容。

表格 5.1. 資源內容

欄位	詳述
----	----

欄位	詳述
resource_id	您的資源名稱
standard	Script 符合的標準。允許的值有： <b>ocf</b> 、 <b>service</b> 、 <b>upstart</b> 、 <b>systemd</b> 、 <b>lsb</b> 、 <b>stonith</b>
type	您希望使用的資源代理程式之名稱，例如 <b>IPaddr</b> 或 <b>Filesystem</b>
provider	OCF 規格能讓多個協力廠商供應相同的 ResourceAgent。Red Hat 所供應的大部份代理程式皆使用了 <b>heartbeat</b> 來作為供應方。

〈表格 5.2, “顯示資源內容的指令”〉概述了顯示可用資源內容的指令。

表格 5.2. 顯示資源內容的指令

pcs 顯示指令	輸出
<b>pcs resource list</b>	顯示一列所有可用的資源。
<b>pcs resource standard</b>	顯示一列可用的資源代理程式標準。
<b>pcs resource providers</b>	顯示一列可用的資源代理程式供應方。
<b>pcs resource list <i>string</i></b>	顯示一列以指定字串篩選的可用資源。您可使用這項指令來顯示以標準名稱、供應方或類型所篩選的資源。

### 5.3. 資源特屬的參數

您可針對任何個別資源使用下列指令來顯示您能為該資源所設置的參數。

```
# pcs resource describe standard:provider:type|type
```

比方說，下列指令顯示了您能為 **LVM** 類型的資源設置的參數。

```
# pcs resource describe LVM
Resource options for: LVM
volgrpname (required): The name of volume group.
exclusive: If set, the volume group will be activated exclusively.
partial_activation: If set, the volume group will be activated even
only partial of the physicalvolumes available. It helps to set to
true, when you are using mirroring logical volumes.
```

### 5.4. 資源 META 選項

除了資源特屬的參數以外，您亦可為任何資源配置額外的資源選項。這些選項會被參數所使用，以決定您資源的運作行為。〈表格 5.3, “資源 Meta 選項”〉詳述了此選項。

表格 5.3. 資源 Meta 選項

欄位	預設	詳述
<b>priority</b>	<b>0</b>	若並非所有資源皆能啓用，叢集將會停下優先順序較低的資源，以讓優先順序較高的資源繼續運作。
<b>target-role</b>	<b>Started</b>	叢集應該將這項資源維持在什麼狀態下？允許的值為： <ul style="list-style-type: none"> <li>* <i>Stopped</i> - 強制資源停下</li> <li>* <i>Started</i> - 允許資源開始運作（在使用多狀態資源的情況下，它們將不會被升階為 <i>master</i>）</li> <li>* <i>Master</i> - 允許資源被啓用，若適當的話，升階</li> </ul>
<b>is-managed</b>	<b>true</b>	叢集是否允許啓用或停用資源？允許的值有： <b>true</b> 、 <b>false</b>
<b>resource-stickiness</b>	<b>0</b>	顯示資源有多偏好維持在其位置上的值。
<b>requires</b>	Calculated	顯示資源能在什麼條件下啓用。 預設為 <b>fencing</b> ，除了在下列情況下。可用的值有： <ul style="list-style-type: none"> <li>* <b>nothing</b> - 叢集能隨意啓用資源。</li> <li>* <b>quorum</b> - 叢集僅能在大部份已配置節點運作的情況下啓用這項資源。若 <b>stonith-enabled</b> 為 <b>false</b> 或是資源的 <b>standard</b> 為 <b>stonith</b> 時，這便會是預設值。</li> <li>* <b>fencing</b> - 叢集僅能在大部份已配置節點運作 <i>和</i>任何失效或不明節點已關閉的情況下啓用這項資源。</li> <li>* <b>unfencing</b> - 叢集僅能在大部份已配置節點運作 <i>和</i>任何失效或不明節點已關閉的情況下，<i>並</i>只在已被 <i>unfenced</i> 的節點上啓用這項資源。若已為一項隔離裝置設置了 <b>provides=unfencing stonith meta</b> 選項，這便會是預設值。欲取得有關於 <b>provides=unfencing stonith meta</b> 選項上的相關資訊，請參閱 <a href="#">〈節 4.5, “為基於儲存裝置的隔離裝置配置反隔離”</a>。</li> </ul>
<b>migration-threshold</b>	<b>INFINITY</b> (已停用)	這項資源在一個節點上所允許失效的次數，直到該節點被標記為不適合提供這項資源。欲取得配置 <b>migration-threshold</b> 選項上的相關資訊，請參閱 <a href="#">〈節 7.2, “在發生錯誤時移動資源”</a> 。
<b>failure-timeout</b>	<b>0</b> (已停用)	與 <b>migration-threshold</b> 選項搭配使用，可顯示在將失效狀況視為未發生前，需等待幾秒鐘，並嘗試允許資源重新在它所失效的節點上運作。欲取得有關於配置 <b>failure-timeout</b> 選項上的相關資訊，請參閱 <a href="#">〈節 7.2, “在發生錯誤時移動資源”</a> 。

欄位	預設	詳述
<b>multiple-active</b>	<b>stop_start</b>	<p>若叢集發現資源不只在一個節點上運作的話，該進行什麼動作。允許的值有：</p> <ul style="list-style-type: none"> <li>* <b>block</b> - 將資源標記為不受管理</li> <li>* <b>stop_only</b> - 停下所有啓用中的事例，並將它們維持在該狀態下</li> <li>* <b>stop_start</b> - 停止所有啓用中的事例，並僅在一個位置上啓用資源</li> </ul>

若要更改一項資源選項的預設值，請使用下列指令。

```
pcs resource defaults options
```

比方說，下列指令會將 **resource-stickiness** 的預設值設為 100。

```
# pcs resource defaults resource-stickiness=100
```

從 **pcs resource defaults** 中省略 *options* 參數會顯示一列目前已為資源選項配置的預設值。下列範例顯示了這項指令在 **resource-stickiness** 預設值被設為 100 後的輸出。

```
# pcs resource defaults
resource-stickiness:100
```

無論您是否已重設了一項資源 meta 選項的預設值，您皆能在建立資源時，將特定資源的資源選項設為一個預設值以外的值。以下顯示了當您為一項資源 meta 選項指定值時所使用的 **pcs resource create** 指令格式。

```
pcs resource create resource_id standard:provider:type|type [resource options] [meta meta_options...]
```

例如，下列指令會建立一項 **resource-stickiness** 值為 50 的資源。

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120
cidr_netmask=24 meta resource-stickiness=50
```

您亦可透過下列指令來為既有的資源群組、複本資源或 master 資源設置資源 meta 選項的值。

```
pcs resource meta resource_id | group_id | clone_id | master_id
meta_options
```

在下列範例中有個名為 **dummy\_resource** 的既有資源。這項指令會將 **failure-timeout** meta 選項設為 20 秒，如此一來資源能嘗試在經過 20 秒後在相同的節點上重新啓用。

```
# pcs resource meta dummy_resource failure-timeout=20s
```

在執行了這項指令後，您便能顯示資源的值，以驗證 **failure-timeout=20s** 是否已設置。

```
# pcs resource show dummy_resource
Resource: dummy_resource (class=ocf provider=heartbeat type=Dummy)
Meta Attrs: failure-timeout=20s
Operations: start interval=0s timeout=20 (dummy_resource-start-timeout-
20)
              stop interval=0s timeout=20 (dummy_resource-stop-timeout-
20)
              monitor interval=10 timeout=20 (dummy_resource-monitor-
interval-10)
```

欲取得複本 meta 選項上的相關資訊，請參閱〈節 8.1, “資源複製”〉欲取得資源 master meta 選項上的相關資訊，請參閱〈節 8.2, “多狀態資源：含有多重模式的資源”〉。

## 5.5. 資源作業

若要確保資源健全，您可加入一項監控作業至資源定義中。若您不為資源指定一項監控作業，就預設值，**pcs** 指令會建立一項監控作業，其間隔會由資源代理程式來判斷。若資源代理程式不提供預設的監控間隔，**pcs** 指令將會建立一項間隔為 60 秒的監控作業。

表格 5.4, “作業內容” 概述了一項資源監控作業的內容。

表格 5.4. 作業內容

欄位	詳述
<b>id</b>	動作的獨特名稱。系統會在您配置一項作業時指定此名稱。
<b>name</b>	需進行的動作。常用的值為： <b>monitor</b> 、 <b>start</b> 、 <b>stop</b>
<b>interval</b>	進行作業的頻率（秒）。預設值為： <b>0</b> ，代表永不。
<b>timeout</b>	在宣告動作失敗前所需等待的時間。若您發現您的系統包含了一項資源花費許多時間啟用或停用，或是在啟用時會執行一項非遞迴的監控動作，並需要比系統所允許之時間（在宣告啟用動作失敗之前）還要久的時間，您可將預設值 20 更改為更長的時間，或更改「op defaults」中的 <b>timeout</b> 值。
<b>on-fail</b>	此動作失敗時所進行的動作。允許的值有： <ul style="list-style-type: none"> <li>* <b>ignore</b> - 假設資源並未失效</li> <li>* <b>block</b> - 不在資源上進行任何其它作業</li> <li>* <b>stop</b> - 停下資源，並且不在其它地方啟用它</li> <li>* <b>restart</b> - 停下資源並將它重新啟用（可能在不同的節點上）</li> <li>* <b>fence</b> - 針對於資源失效的節點進行 STONITH</li> <li>* <b>standby</b> - 將資源失效於的節點上的所有資源移掉</li> </ul> <p>當 STONITH 啟用時，<b>stop</b> 作業的預設值為 <b>fence</b>，停用時則為 <b>block</b>。所有其它作業的預設值皆為 <b>restart</b>。</p>
<b>enabled</b>	若是 <b>false</b> 的話，作業便會被視為不存在。允許的值為： <b>true</b> 、 <b>false</b>



您可透過下列指令來在您建立資源時配置監控作業。

```
pcs resource create resource_id standard:provider:type|type
[resource_options] [op operation_action operation_options [operation_type
operation_options]....
```

比方說，下列指令會建立一項 **IPAddr2** 資源搭配監控作業。新的資源稱為 **VirtualIP**，其 IP 位址在 **eth2** 上為 192.168.0.99 而子網路遮罩為 24。監控作業會每 30 秒執行一次。

```
# pcs resource create VirtualIP ocf:heartbeat:IPAddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2 op monitor interval=30s
```

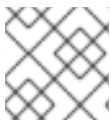
```
# pcs resource create my_address IPAddr2 ip=10.20.30.40 cidr_netmask=24 op
monitor
```

此外，您可透過下列指令來新增一項監控作業至既有的資源中。

```
pcs resource op add resource_id operation_action [operation_properties]
```

使用下列指令來刪除一項已配置的資源作業。

```
pcs resource op remove resource_id operation_name operation_properties
```



### 注意

您必須指定確切的作業內容以正確移除一項既有的作業。

若要更改監控選項的值，您必須移除既有的作業，然後新增作業。比方說，您可透過下列指令來建立一項 **VirtualIP**。

```
# pcs resource create VirtualIP ocf:heartbeat:IPAddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2
```

就預設值，這項指令會建立這些作業。

```
Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
             stop interval=0s timeout=20s (VirtualIP-stop-timeout-20s)
             monitor interval=10s timeout=20s (VirtualIP-monitor-interval-
10s)
```

若要更改停止逾時作業，請執行下列指令。

```
# pcs resource op remove VirtualIP stop interval=0s timeout=20s
# pcs resource op add VirtualIP stop interval=0s timeout=40s

# pcs resource show VirtualIP
Resource: VirtualIP (class=ocf provider=heartbeat type=IPAddr2)
Attributes: ip=192.168.0.99 cidr_netmask=24 nic=eth2
Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
             monitor interval=10s timeout=20s (VirtualIP-monitor-
interval-10s)
             stop interval=0s timeout=40s (VirtualIP-name-stop-interval-
```

```
0s-timeout-40s)
```

若要為監控作業設置全域預設值，請使用下列指令。

```
pcs resource op defaults [options]
```

比方說，下列指令會為所有監控作業設置一項 **timeout** 值為 240s 的全域預設值。

```
# pcs resource op defaults timeout=240s
```

若要顯示監控作業目前已配置的預設值，當您執行 **pcs resource op defaults** 指令時，請勿指定任何選項。

比方說，下列指令會顯示一個已被配置了 **timeout** 值為 240s 的叢集的預設監控作業值。

```
# pcs resource op defaults
timeout: 240s
```

## 5.6. 顯示已配置的資源

若要顯示一列所有已配置的資源，請使用下列指令。

```
pcs resource show
```

比方說，若您的系統已配置了一項名為 **VirtualIP** 的資源，以及一項名為 **WebSite** 的資源，**pcs resource show** 指令將會顯示以下輸出。

```
# pcs resource show
VirtualIP (ocf::heartbeat:IPaddr2): Started
WebSite (ocf::heartbeat:apache): Started
```

若要顯示一列所有已配置的資源，以及為這些資源所配置的參數，請使用 **pcs resource show** 指令的 **--full** 選項，如同以下範例。

```
# pcs resource show --full
Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
Attributes: ip=192.168.0.120 cidr_netmask=24
Operations: monitor interval=30s
Resource: WebSite (type=apache class=ocf provider=heartbeat)
Attributes: statusurl=http://localhost/server-status
configfile=/etc/httpd/conf/httpd.conf
Operations: monitor interval=1min
```

若要顯示一項資源的已配置參數，請使用下列指令。

```
pcs resource show resource_id
```

比方說，下列指令會顯示目前已為 **VirtualIP** 資源配置的參數。

```
# pcs resource show VirtualIP
Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
Attributes: ip=192.168.0.120 cidr_netmask=24
```

```
Operations: monitor interval=30s
```

## 5.7. 修改資源參數

若要修改一項已配置資源的參數，請使用下列指令。

```
pcs resource update resource_id [resource_options]
```

下列指令循序顯示了 **VirtualIP** 資源已配置參數的初始值、更改 **ip** 參數值的指令，以及搭配更新指令的值。

```
# pcs resource show VirtualIP
Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
Attributes: ip=192.168.0.120 cidr_netmask=24
Operations: monitor interval=30s
# pcs resource update VirtualIP ip=192.169.0.120
# pcs resource show VirtualIP
Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
Attributes: ip=192.169.0.120 cidr_netmask=24
Operations: monitor interval=30s
```

## 5.8. 多重監控作業

您可為單一資源配置資源代理程式所支援的任意監控作業數量。如此一來，您便能每分鐘進行一項粗略的健康狀態檢查，並以較長的時間間隔進行更為高度的檢查。



### 注意

當配置多項監控作業時，您必須確認沒有兩項作業以相同的時間間隔執行。

若要為一項支援在不同層級上進行更加詳細檢查的資源配置額外的監控作業，您需要加入一項 **OCF\_CHECK\_LEVEL=*n*** 選項。

比方說，若您配置了以下的 **IPaddr2** 資源，就預設值這會建立一項間隔為 10 秒、逾時值為 20 秒的監控作業。

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99
cidr_netmask=24 nic=eth2
```

若 Virtual IP 支援一項詳細度為 10 的檢查，下列指令便會使 Packemaker 每 60 秒執行一次更加進階的監控檢查（不包括每 10 秒鐘一次的正常 Virtual IP 檢查）。（如同先前所述，您不應該也以 10 秒鐘間隔配置額外監控作業。）

```
# pcs resource op add VirtualIP monitor interval=60s OCF_CHECK_LEVEL=10
```

## 5.9. 啟用和停用叢集資源

下列指令會啟用 **resource\_id** 所指定的資源。

```
pcs resource enable resource_id
```

下列指令會停用 **resource\_id** 所指定的資源。

```
pcs resource disable resource_id
```

## 5.10. 叢集資源清除

若一項資源失效的話，當您顯示叢集狀態時，有一則錯誤訊息將會出現。若您解析了該資源，您可透過 **pcs resource cleanup** 指令來清除該錯誤狀態。這項指令會重設資源狀態與錯誤計數，要求叢集忘掉一項資源的作業歷史，並重新偵測其目前狀態。

下列指令會清除由 **resource\_id** 所指定的資源。

```
pcs resource cleanup resource_id
```

## 章 6. 資源限制式

您可藉由配置叢集中資源限制式以決定該資源的行為。您可配置下列種類的限制式：

- **location** 限制式 — 位置限制式可決定資源能在哪個節點上執行。位置限制式詳述於〈[節 6.1, “位置限制式”](#)〉中。
- **order** 限制式 — 順序限制式可決定資源的執行順序。順序限制式詳述於〈[節 6.2, “順序限制”](#)〉中。
- **colocation** 限制式 — colocation 限制式可決定資源與其它資源相對之下應放置在什麼位置上。Colocation 限制式詳述於〈[節 6.3, “資源的 Colocation”](#)〉中。

Pacemaker 支援資源群組的概念，以簡化限制的配置，它會將一組資源併在一起並確認資源會按照順序啟用，並以反向順序停下。欲取得更多有關於資源群組上的相關資訊，請參閱〈[節 6.5, “資源群組”](#)〉。

### 6.1. 位置限制式

位置限制式可決定資源能在哪個節點上執行。您可配置位置限制式以決定資源是否會偏好或避免在特定節點上執行。

[表格 6.1, “位置限制式選項”](#) 概述了配置位置限制式的選項。

**表格 6.1. 位置限制式選項**

欄位	詳述
<b>id</b>	限制式的獨特名稱。這是當您透過 <b>pcs</b> 配置位置限制式時，由系統所設置的名稱。
<b>rsc</b>	資源名稱
<b>node</b>	節點的名稱
<b>score</b>	用來判斷資源是否應在某個節點上執行，或避免在某個節點上執行的值。 <b>INFINITY</b> 這個值會將「應該」更改為「必須」； <b>INFINITY</b> 乃資源位置限制的預設分數值。

下列指令會為一項資源建立位置限制，以偏好在特定節點上執行。

```
pcs constraint location rsc prefers node[=score] ...
```

下列指令會為一項資源建立位置限制，以避免在特定節點上執行。

```
pcs constraint location rsc avoids node[=score] ...
```

還有兩項能指定資源可在哪個節點上運作的策略：

- **Opt-In Clusters** — 配置一個就預設值無法讓任何資源在任何位置運作於的叢集，並選擇性為指定的資源啟用允許的節點。配置一個 opt-in 叢集的程序詳述於〈[節 6.1.1, “配置一個「Opt-In」 Cluster”](#)〉中。

- Opt-Out Clusters — 配置一個就預設值能讓所有資源執行於任何位置的叢集，然後為不允許執行於指定節點上的資源建立位置限制。配置一個 opt-out cluster 的程序詳述於〈節 6.1.2, “配置一個「Opt-Out」 Cluster”〉中。

您應配置一個 opt-in 或 opt-out cluster 的決定取決於您的個人偏好與您叢集本身的架構。若您大部份的資源皆可在多數節點上執行，那麼 opt-out 可能會是較容易的配置方式。另一方面，若大部份的資源僅能在少數節點上執行，opt-in 配置可能會較為容易。

### 6.1.1. 配置一個「Opt-In」 Cluster

若要建立一個 opt-in cluster，請將 `symmetric-cluster` 叢集內容設為 `false`，以預設避免資源在任何位置上執行。

```
# pcs property set symmetric-cluster=false
```

為個別資源啟用節點。下列指令能配置位置限制，如此一來 `Webserver` 資源將會偏好使用 `example-1` 節點，`Database` 資源將偏好使用 `example-2` 節點，而當偏好的節點失效時，這兩項資源皆可容錯移轉至 `example-3` 節點。

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver prefers example-3=0
# pcs constraint location Database prefers example-2=200
# pcs constraint location Database prefers example-3=0
```

### 6.1.2. 配置一個「Opt-Out」 Cluster

若要建立一個 opt-out cluster，請將 `symmetric-cluster` 叢集內容設為 `true` 以允許資源就預設值能在任何位置上執行。

```
# pcs property set symmetric-cluster=true
```

下列指令接著將會產生一項與〈節 6.1.1, “配置一個「Opt-In」 Cluster”〉中的範例相等的配置。若這兩項資源偏好的節點失效的話，兩項資源皆可容錯移轉至 `example-3` 節點，因為每個節點皆有個為零的隱含分數。

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver avoids example-2=INFINITY
# pcs constraint location Database avoids example-1=INFINITY
# pcs constraint location Database prefers example-2=200
```

請注意，您無須在這些指令中指定一個值為 `INFINITY` 的分數，因為那是分數的預設值。

## 6.2. 順序限制

順序限制能決定資源執行的順序。您可配置一項順序限制來決定資源啟用與停用的順序。

使用以下指令來配置一項順序限制。

```
pcs constraint order [action] resource_id then [action] resource_id
[options]
```

表格 6.2, “順序限制的內容” 概述了用來配置順序限制的內容與選項。

表格 6.2. 順序限制的內容

欄位	詳述
resource_id	動作所會執行於的資源之名稱。
action	<p>欲在資源上執行的動作。 <i>action</i> 內容可用的值如下：</p> <ul style="list-style-type: none"> <li>* <b>start</b> - 啓用資源。</li> <li>* <b>stop</b> - 停用資源。</li> <li>* <b>promote</b> - 將資源從 slave 資源升階為 master 資源。</li> <li>* <b>demote</b> - 將資源從 master 資源降階為 slave 資源。</li> </ul> <p>若未指定動作，預設動作將會是 <b>start</b>。欲取得有關於 master 和 slave 資源上的相關資訊，請參閱 <a href="#">〈節 8.2, “多狀態資源：含有多重模式的資源”</a>。</p>
kind 選項	<p>如何強制限制。 <b>kind</b> 選項可用的值如下：</p> <ul style="list-style-type: none"> <li>* <b>Optional</b> - 僅適用於兩項資源皆啓用與/或停用的情況下。欲取得選用性順序的相關資訊，請參閱 <a href="#">〈節 6.2.2, “諮詢順序”</a>。</li> <li>* <b>Mandatory</b> - 總是（預設值）。若您所指定的第一項資源正在停用或是無法啓用，您所指定的第二項資源就必須停止。欲取得強制順序上的相關資訊，請參閱 <a href="#">〈節 6.2.1, “強制順序”</a>。</li> <li>* <b>Serialize</b> - 確保一組資源不會有同時停止/開始的動作發生。</li> </ul>
symmetrical 選項	若是「true」的話（這是預設值），會將資源以反向的順序停下。預設值： <b>true</b>

### 6.2.1. 強制順序

強制限制代表您制定的第二項資源無法再您所指定的第一項資源啓用前執行。此乃 **kind** 選項的預設值。保留為預設值能確保您指定的第二項資源會在您指定的第一項資源的狀態更改時做出反應。

- 若您所指定的第一項資源原本正在執行並且被停下的話，您所指定的第二項資源（若它原本正在執行的話）也會被停下。
- 若您所指定的第一項資源原本並未在執行並且無法開始的話，您所指定的第二項資源（若它原本正在執行的話）將會被停下。
- 若您所指定的第一項資源在您指定的第二項資源正在執行時（重新）開始的話，您所指定的第二項資源將會停下並重新開始。

### 6.2.2. 諮詢順序

當為順序限制指定了 **kind=Optional** 選項時，這項限制會被視為選用性，並且只有在兩項資訊皆在停下與/或開始時才會有效用。任何對於您所指定的第一項資源的狀態變更皆對您所指定的第二項資源無效。

下列指令能為名為 **VirtualIP** 和 **dummy\_resource** 的資源配置一項諮詢順序限制。

```
# pcs constraint order VirtualIP then dummy_resource kind=Optional
```

### 6.2.3. 已排序的資源集

管理員建立序列排序的資源乃常見的情況，在此情況下，資源 A 可能會在資源 B 之前開始，並且資源 B 則可能會在資源 C 之前開始。您可透過下列指令來配置序列排序的資源。這些資源將會以指定的順序開始。

```
pcs constraint order set resource1 resource2 [resourceN]... [options] [set resource1 resource2 ...]
```

若您有三項名為 **D1**、**D2** 以及 **D3** 的資源，下列指令能將它們配置為已排序的資源集。

```
# pcs constraint order set D1 D2 D3
```

### 6.2.4. 將資源從順序限制中移除

使用下列指令來將資源從任何順序限制中移除。

```
pcs constraint order remove resource1 [resourceN]...
```

## 6.3. 資源的 COLOCATION

Colocation 限制能決定一項資源的位置是否取決於另一項資源。

在兩項資源之間建立 colocation 需考慮到一項重要的副作用：這會影響資源被指派至節點的順序。這是因為您在不知道資源 B 的位置在哪的情況下，無法使資源 A 與資源 B 相聯。因此當您建立 colocation 限制時，您必須考慮應讓資源 A 與資源 B 相聯，或是讓資源 B 與資源 A 相聯。

建立 colocation 限制時需考慮到的另一件事就是，假設資源 A 已與資源 B collocate，當叢集決定資源 B 應使用哪個節點的同時，也會考量資源 A 的偏好設定。

下列指令能建立一項 colocation 限制。

```
pcs constraint colocation add [master|slave] source_resource with [master|slave] target_resource [score] [options]
```

欲取得 master 與 slave 資源上的相關資訊，請參閱〈節 8.2, “多狀態資源：含有多重模式的資源”〉。

表格 6.3, “Colocation 限制的內容” 概述了配置 colocation 限制的內容與選項。

表格 6.3. Colocation 限制的內容

欄位	詳述
source_resource	Colocation 來源。若無法滿足限制，叢集可能會決定根本不允許資源執行。
target_resource	Colocation 目標。叢集會決定先將這項資源放在哪裡，然後才決定將來源資源放在哪裡。
score	正數值代表資源應在相同的節點上執行。負數值則代表資源不應在相同的節點上執行。若值為 + <b>INFINITY</b> (預設值) 則代表 source_resource 必須在與 target_resource 相同的節點上執行。若值為 - <b>INFINITY</b> 則代表 source_resource 必須在與 target_resource 不同的節點上執行。



### 6.3.1. 強制安置 (Mandatory Placement)

任何時候當限制的分數為 **+INFINITY** 或 **-INFINITY** 時，強制安置就會發生。在此情況下，若無法滿足限制，那麼 *source\_resource* 將不被允許執行。若 **score=INFINITY**，這可能代表 *target\_resource* 未啓用。

若您需要 **myresource1** 總是在與 **myresource2** 相同的機器上執行，您需加入下列限制：

```
# pcs constraint colocation add myresource1 with myresource2
score=INFINITY
```

因為使用了 **INFINITY**，若 **myresource2**（基於某些原因）無法在任何叢集節點上執行，則 **myresource1** 將不會被允許執行。

此外，您亦可進行相反配置，將叢集中的 **myresource1** 配置為不可在與 **myresource2** 相同的機器上執行。在此情況下請使用 **score=-INFINITY**

```
# pcs constraint colocation add myresource1 with myresource2 score=-
INFINITY
```

又，透過指定 **-INFINITY**，限制將會綁定。因此若唯一能執行的位置為 **myresource2** 早已在執行的位置，那麼 **myresource1** 將無法在任何位置上執行。

### 6.3.2. 諮詢安置 (Advisory Placement)

若強制安置代表「必須」或「必須不」，則諮詢安置就是「我會比較偏好」的替代選項。若限制的分數大於 **-INFINITY** 而小於 **INFINITY**，叢集將會嘗試符合您的希望，不過若替代方案乃停下某些叢集資源的話，叢集就可能忽略您的希望。諮詢 *colocation* 限制能與配置的其它要素結合，並使其行為被視為強制性。

### 6.3.3. Colocate 一組資源

使用以下指令來在一組資源上建立一項 *colocation* 限制。您可將 **sequential** 選項設為 **true** 或是 **false** 以表示這組已 *colocate* 的資源是否是一組已排序的資源。

```
colocation set resource1 resource2 [resourceN]... [setoptions name=value]
... [set resourceX resourceY ...] [setoptions name=value...]
```

您能為一個設為了 **master** 或 **slave** 的 *colocation* 設置 **role** 選項。欲取得有關於多狀態資源上的相關資訊，請參閱 [〈節 8.2, “多狀態資源：含有多重模式的資源”〉](#)。

### 6.3.4. 移除 Colocation 限制

使用下列指令來移除含有 *source\_resource* 的 *colocation* 限制。

```
pcs constraint colocation remove source_resource target_resource
```

## 6.4. 顯示限制

您可使用數項指令來顯示已配置的限制。

下列指令能列出所有目前的位置、順序和 *colocation* 限制。

```
pcs constraint list|show
```

下列指令能列出所有目前的位置限制。

- 若指定了 **resources**，各項資源的位置限制皆會顯示。這是預設行為。
- 若指定了 **nodes**，各節點的位置限制皆會顯示。
- 若指定了資源或節點，那麼僅有有關於這些資源或節點的資訊會顯示。

```
pcs constraint location [show resources|nodes [specific nodes|resources]]
[--full]
```

以下指令會列出所有目前的順序限制。若指定了 **--full** 選項，內部限制 ID 將會顯示。

```
pcs constraint order show [--full]
```

以下指令會列出所有目前的 colocation 限制。若指定了 **--full** 選項，內部限制 ID 將會顯示。

```
pcs constraint colocation show [--full]
```

以下指令會列出參照特定資源的限制。

```
pcs constraint ref resource ...
```

## 6.5. 資源群組

叢集最常見的要素之一就是一組需要位於相同位置、循序啓用，和相反順序停用的資源。為了簡化這項配置，Pacemaker 提供了群組概念上的支援。

您可透過以下指令來建立資源群組，並指定欲包含在群組中的資源。若群組不存在的話，這項指令便會建立群組。若群組存在的話，這項指令便會將額外資源加入群組中。資源將會以您透過這項指令所指定的順序開始，並以其開始順序的相反順序停止。

```
pcs resource group add group_name resource_id...
```

您亦可透過下列指令，在您的建立資源時新增資源至既有群組中。您所建立的資源會被加入至名為 *group\_name* 的群組中。

```
pcs resource create resource_id standard:provider:type|type
[resource_options] [op operation_action operation_options] --group
group_name
```

您能透過以下指令將資源從群組中移除。若群組中沒有資源的話，這項指令便會將群組本身移除。

```
pcs resource group remove group_name resource_id...
```

下列指令會列出所有目前已配置的資源群組。

```
pcs resource group list
```

下列範例會建立一個名為 **shortcut** 並包含了既有資源 **IPaddr** 與 **Email** 的資源群組。

```
# pcs resource group add shortcut IPaddr Email
```

群組可包含的資源數量是無限的。群組的基礎內容如下。

- 資源會以您所指定的順序開始（在此範例中，**IPaddr** 會先開始，然後才是 **Email**）。
- 資源會以您所指定的反向順序停止（先是 **Email** 然後才是 **IPaddr**）。

若群組中的一項資源無法在任何位置上執行，那麼在該資源之後所指定的資源皆不允許執行。

- 若 **IPaddr** 無法在任何位置上執行，**Email** 也將無法執行。
- 儘管 **Email** 無法在任何位置上執行，它卻完全不會影響到 **IPaddr**。

顯然地，當群組逐漸擴展時，建立資源群組時所需進行的配置便可能會大幅減少。

### 6.5.1. 群組選項

資源群組會從它所包含的資源繼承下列選項：**priority**、**target-role**、**is-managed**。欲取得資源選項上的相關資訊，請參閱〈[表格 5.3, “資源 Meta 選項”](#)〉。

### 6.5.2. 群組綁定

綁定（Stickiness）代表一項資源有多希望留在其目前的位置上，以及對於群組的加總量值。群組中每項啓用中的資源的綁定值皆會加總成為該群組的綁定總值。因此若預設的 **resource-stickiness** 為 100，某個群組擁有七個成員，而其中五個成員正啓用中的話，則此群組對於目前位置的偏好分數就會是 500。

## 章 7. 管理叢集資源

本章詳述了各項您能使用來管理叢集資源的指令。它提供了下列程序的資訊。

- [節 7.1, “在叢集之間手動移動資源”](#)
- [節 7.2, “在發生錯誤時移動資源”](#)
- [節 7.4, “啓用、停用和禁止叢集資源”](#)
- [節 7.5, “停用監控作業”](#)

### 7.1. 在叢集之間手動移動資源

您可置換叢集並強制資源由其目前位置上進行移動。您會這樣做的情況有兩種：

- 當節點進行維護，並且您需要將該節點上所有資源移至另一節點時
- 當需要移動單項資源時

若要將節點上的所有資源移至另一個節點上時，您需要使節點進入預備模式中。欲取得有關於如何將節點設為預備模式的相關資訊，請參閱 [〈節 3.2.5, “預備模式”](#)〉。

若要將資源由一個運作中的節點上移掉，請使用下列指令並如定義一般指定節點的 `resource_id`。

```
pcs resource move resource_id
```

若您希望指定您要移動的資源將在哪個節點上執行，請使用下列指令來指定 `destination_node`。

```
pcs resource move resource_id destination_node
```

是請使用下列指令來將資源移回至它原本執行於的節點上，並讓叢集恢復正常作業。這將移除 `move resource_id` 指令所定義的限制式。

```
pcs resource clear resource_id [node]
```

請注意，當您執行 `pcs resource move` 指令時，這會將限制式加入至資源中以防它在顯示的節點上執行。當您執行 `pcs resource clear` 指令時，這將會移除限制式。這並不一定會將資源移回顯示的節點上；在此時資源能在哪裡執行取決於您一開始如何配置了您的資源。欲取得更多有關於資源限制式上的相關資訊，請參閱 [〈章 6, 資源限制式](#)〉。

### 7.2. 在發生錯誤時移動資源

當您建立一項資源時，您可配置這項資源的 `migration-threshold` 選項，使其在定義的失效次數經過後移至一個新的節點上。當到達了門檻值時，此節點將不再允許執行失效的資源，直到：

- 管理員可透過 `pcs resource failcount` 指令手動重設資源的失效計數。
- 資源的 `failure-timeout` 值已到達。

就預設值未定義門檻值。



## 注意

為資源設定 **migration-threshold** 與配置資源來進行遷移不同，在此情況下資源會移至另一個位置上而不失去其狀態。

下列範例會新增遷移門檻值 10 至名為 **dummy\_resource** 的資源，這顯示資源將在失效 10 次之後移至新的節點上。

```
# pcs resource meta dummy_resource migration-threshold=10
```

您能以下列指令來為整個叢集加入預設的遷移門檻值。

```
# pcs resource defaults migration-threshold=10
```

若要判斷資源目前的失效狀態與限制為何，請使用 **pcs resource failcount** 指令。

遷移門檻的概念包含了兩項例外情況；它們會在資源無法啟動或停止時發生。啟動失敗會使失效計數被設為 **INFINITY**，並且總是會造成資源即刻被移動。

停止失敗稍微不同且較嚴重。若一項資源停止失敗而啓用了 STONITH 的話，叢集將會隔離該節點以在其它地方啟動這項服務。若 STONITH 未啓用的話，則叢集將無法繼續並且將不會嘗試在其它地方啓用這項服務，而是會在失效逾時之後再次嘗試停止它。

## 7.3. 失去連線時移動資源

設定叢集使其在失去外部連線時移動資源的這項程序包含了兩個步驟。

1. 新增 **ping** 資源至叢集。**ping** 資源會使用相同名稱的系統工具程式來測試是否能連上一列（由 DNS 主機名稱或是 IPv4/IPv6 位址所指定）機器，並使用此結果來維護一個名為 **pingd** 的節點屬性。
2. 為會在失去連線時將資源移至一個不同節點的資源配置位置限制式。

表格 5.1, “資源內容” 詳述了您能為 **ping** 資源設置的內容。

表格 7.1. ping 資源的內容

欄位	詳述
<b>dampen</b>	等待進一步變更發生的所需時間 (dampen)。這會避免一項資源在叢集節點於不同時間點發現失去連線時，在叢集之間跳來跳去。
<b>multiplier</b>	連接的 ping 節點會乘以這個值以取得一項分數。當配置了多重的 ping 節點時會相當有幫助。
<b>host_list</b>	要判斷目前連線狀態為何所需聯絡的機器。允許的值包括可解析的 DNS 主機名稱、IPv4 與 IPv6 位址。

下列範例指令會建立一項 **ping** 資源，用來驗證至 **www.example.com** 的連線。一般來講您會驗證您網路閘道器/路由器的連線。您將配置 **ping** 資源作為副本，如此一來資源便會在所有叢集節點上執行。

```
# pcs resource create ping ocf:pacemaker:ping dampen=5s multiplier=1000
host_list=www.example.com --clone
```

下列範例會為既有的資源 **Webserver** 配置位置限制式規則。這會使 **Webserver** 資源移至一部能夠 ping **www.example.com** 的主機（若該資源目前執行於的主機無法 ping **www.example.com** 的話）。

```
# pcs constraint location Webserver rule score=-INFINITY pingd lt 1 or
not_defined pingd
```

## 7.4. 啓用、停用和禁止叢集資源

除了詳述於 [節 7.1, “在叢集之間手動移動資源”](#) 中的 **pcs resource move** 指令之外，您亦可使用其它各種指令來控制叢集資源的運作行為。

您可透過下列指令來停止一項執行中的資源，並避免叢集再次啓動它。取決於剩餘的配置（限制式、選項、錯誤等等），資源可能保持啓動。若您指定了 **--wait** 選項，**pcs** 將會等待達 30 秒鐘（或指定的 'n' 秒）來讓資源停止，並且若資源停止的話回傳 0，若資源未停止的話則回傳 1。

```
pcs resource disable resource_id [--wait[=n]]
```

您可使用下列指令來允許叢集啓動一項資源。取決於剩餘的配置，資源可能會保持停止。若您指定了 **--wait** 選項，**pcs** 將會等待達 30 秒鐘（或指定的 'n' 秒）以讓資源啓動，並且若資源有啓動的話回傳 0，若未啓動的話則回傳 1。

```
pcs resource enable resource_id [--wait[=n]]
```

請使用下列指令來避免資源在指定的節點（或若未指定節點的話在目前的節點）上運作。

```
pcs resource ban resource_id [node]
```

請注意，當您執行 **pcs resource ban** 指令時，這會將限制式加入到資源中以防它在顯示的節點上執行。您可執行 **pcs resource clear** 指令來移除限制。這並不一定會將資源移回顯示的節點上；在此時資源能在哪裡執行取決於您一開始如何配置了您的資源。欲取得更多有關於資源限制式上的相關資訊，請參閱 [〈章 6, 資源限制式〉](#)。

```
pcs resource clear resource_id [node]
```

您可使用 **pcs resource** 指令的 **debug-start** 參數來強制一項指定的資源啓動於目前的節點上，忽略叢集建議並印出啓動資源時的輸出。這主要用於資源除錯；在叢集上啓動一項資源（幾乎）總是透過 Pacemaker 而非直接透過 **pcs** 指令。若您的資源沒啓動的話，這一般是因為資源配置錯誤（您可透過系統日誌除錯）、資源的啓動或是停止受到了限制。您可使用這項指令來測試資源配置，不過它一般不該被使用來在叢集中啓動資源。

**debug-start** 指令的格式如下。

```
pcs resource debug-start resource_id
```

## 7.5. 停用監控作業

欲停止一項遞迴的監控，最簡單的方式就是將它刪除。然而，有時您可能僅希望將它暫時停用。在此情況下，請將 `enabled="false"` 附加至作業的定義中。當您希望復原監控作業時，請在作業定義中設定 `enabled="true"`。

## 7.6. 資源管理

您可將資源設為不受管理（unmanaged）模式，這會顯示資源還位於配置中，不過 Pacemaker 並不負責管理該資源。

下列指令會將顯示的資源設為不受管理的模式。

```
pcs resource unmanage resource1 [resource2] ...
```

下列指令會將資源設為管理模式，也就是預設狀態。

```
pcs resource manage resource1 [resource2] ...
```

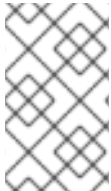
您可透過 `pcs resource manage` 或是 `pcs resource unmanage` 指令來指定資源群組的名稱。這項指令會針對於群組中的所有資源進行動作，如此一來您便能透過單一指令來管理或解除管理群組中的所有資源，並對於資源進行個別的管理。

## 章 8. 進階資源類型

本章節詳述了 Pacemaker 所支援的進階資源類型。

### 8.1. 資源複製

您可複製一項資源，以讓這項資源能啓用於多個節點上。比方說，您可使用複製的資源來配置多重 IP 資源，以在叢集之間提供節點平衡。您可複製任何資源代理程式所支援的資源。一項複製資源包含了一個資源或是一個資源群組。



#### 注意

僅有能同時啓用於多個節點上的資源適合複製。比方說，一項由共享記憶體裝置掛載了非叢集檔案系統（例如 **ext4**）的 **Filesystem** 資源不應被複製。因為 **ext4** 分割區無法對應叢集，此檔案系統不適合同時在多重節點上進行讀/寫作業。

#### 8.1.1. 建立和移除一項複製的資源

您可藉由以下指令同時建立並複製一項資源。

```
pcs resource create resource_id standard:provider:type|type [resource
options] \
--clone [meta clone_options]
```

複本的名稱會是 **resource\_id-clone**。

您無法以單一指令建立資源群組以及該資源群組的複本。

此外，您可透過下列指令來建立一個先前所建立之複本或是資源群組的複本。

```
pcs resource clone resource_id | group_name [clone_options]...
```

複本的名稱會是 **resource\_id-clone** 或 **group\_name-clone**。



#### 注意

您僅需要在一個節點上配置資源配置變更。



#### 注意

當配置限制式時，請總是使用群組或複本的名稱。

當您建立一項資源的複本時，該複本會使用資源的名稱並附加 **-clone** 至名稱。下列指令會建立一項 **apache** 類型，名為 **webfarm** 的資源，以及這項資源的複本，名為 **webfarm-clone**。

```
# pcs resource create webfarm apache clone
```

使用下列指令來移除一項資源或是資源群組的複本。這不會移除資源或資源群組本身。

```
pcs resource unclone resource_id | group_name
```



欲取得更多有關於資源選項上的資訊，請參閱〈節 5.1, “資源建立”〉。

表格 8.1, “資源複製選項” 詳述了您可為複製的資源指定的選項。

表格 8.1. 資源複製選項

欄位	詳述
<b>priority, target-role, is-managed</b>	由被複製的資源所繼承的選項，如 表格 5.3, “資源 Meta 選項” 中所述。
<b>clone-max</b>	需要多少資源複本方可啓用。預設值為叢集中的節點數量。
<b>clone-node-max</b>	一個節點上能啓用多少資源複本；預設值為 <b>1</b> 。
<b>notify</b>	當停用或啓用複本時，先前通知所有其它複本，並在動作成功完成後再次通知。允許的值包含： <b>false</b> 、 <b>true</b> 。預設值為 <b>false</b> 。
<b>globally-unique</b>	各個複本是否會進行不同的任務？允許的值包含： <b>false</b> 、 <b>true</b>  若此選項的值為 <b>false</b> ，這些資源無論在哪裡執行皆會有相同的行為，因此每一部機器上皆只能有一個啓用的複本。  若此選項的值為 <b>true</b> ，一個在一部機器上執行的複本將不會與另一個事例相等，無論該事例是在另一個節點還是相同節點上執行。若 <b>clone-node-max</b> 的值大於一，預設值將會是 <b>true</b> ；否則預設值便會是 <b>false</b> 。
<b>ordered</b>	複本是否該連續啓用（而非平行啓用）。允許的值為： <b>false</b> 、 <b>true</b> 。預設值為 <b>false</b> 。
<b>interleave</b>	更改順序限制式的行為（clones/masters 之間）如此一來第一份 clone 的複本便能在相同節點上第二份 clone 的複本啓用或停用時，即刻啓用或停用（無須等待第二份 clone 的每個複本皆啓用或停用）。允許的值為： <b>false</b> 、 <b>true</b> 。預設值為 <b>false</b> 。

### 8.1.2. 複製限制式

大部分情況下，複本在各個啓用中的叢集節點上皆有一項事例。然而，您亦可將資源複本的 **clone-max** 設為一個比叢集中節點總數還要小的值。在此情況下，您可根據叢集根據資源位置限制式，決定叢集應將複本指定給哪些節點。除了必須使用複本的 ID 之外，這些限制式和使用於一般資源的限制式編寫上並無差異。

以下指令會為叢集建立一項位置限制式，以透過偏好設定將資源複本 **webfarm-clone** 指定給 **node1**。

```
# pcs constraint location webfarm-clone prefers node1
```

順序限制式的行為與複本不太一樣。在以下範例中，**webfarm-stats** 會等待所有需要啓用的 **webfarm-clone** 複本皆啓用後，它才會自行啓用。僅有在無 **webfarm-clone** 複本可啓用時，才可防止 **webfarm-stats** 啓用。此外，**webfarm-clone** 會等待 **webfarm-stats** 停用後，才會自行停用。

```
# pcs constraint order start webfarm-clone then webfarm-stats
```

正規（或群組）資源與複本的 colocation 代表資源可在任何持有一份該複本的機器上運作。叢集將會視複本執行的所在位置，和資源自己的位置偏好設定來選擇一個複本。

您亦可進行複本之間的 colocation。在此情況下，允許的複本位置將會被限制在複本運作（或將會運作）於的節點上。之後分配動作將會正常進行。

下列指令會建立一項 colocation 限制式，以確保資源 **webfarm-stats** 會在與一份啓用中的 **webfarm-clone** 複本相同的節點上執行。

```
# pcs constraint colocation add webfarm-stats with webfarm-clone
```

### 8.1.3. 複本綁定

為了能擁有穩定的配置模式，複本就預設值會被綁定。若未提供 **resource-stickiness** 的值，複本將會使用 1 這個值。因為這個值較小，因此不會造成其它資源分數計算上的干擾，而卻足以預防 Pacemaker 在不必要的情況下在叢集之間移動複本。

## 8.2. 多狀態資源：含有多重模式的資源

多狀態資源為特殊的複本資源。它們能允許事例以兩種作業模式之一進行；這些模式為 **Master** 與 **Slave**。模式的名稱沒有特殊的意義，除了事例啓用時必須處於 **Slave** 狀態下。

您能以下列單一指令來將資源建立為 master 或是 slave 複本。

```
pcs resource create resource_id standard:provider:type|type [resource
options] \
--master [meta master_options]
```

Master/slave 複本的名稱會是 **resource\_id-master**。

此外，您亦可透過以下指令來從一個先前建立的資源或資源群組建立一項 master/slave 資源：當使用這項指令時，您能為 master/slave 複本指定一組名稱。若您未指定名稱的話，master/slave 複本的名稱將會是 **resource\_id-master** 或是 **group\_name-master**。

```
pcs resource master master/slave_name resource_id|group_name
[master_options]
```

欲取得更多有關於資源選項上的資訊，請參閱〈[節 5.1, “資源建立”](#)〉。

[表格 8.2, “多狀態資源的內容”](#) 詳述了您可為多狀態資源指定的選項。

**表格 8.2. 多狀態資源的內容**

欄位	詳述
<b>id</b>	多狀態資源的名稱
<b>priority</b> 、 <b>target-role</b> 、 <b>is-managed</b>	請參閱〈 <a href="#">表格 5.3, “資源 Meta 選項”</a> 〉。

欄位	詳述
<b>clone-max</b> 、 <b>clone-node-max</b> 、 <b>notify</b> 、 <b>globally-unique</b> 、 <b>ordered</b> 、 <b>interleave</b>	請參閱 <a href="#">〈表格 8.1, “資源複製選項”</a> 。
<b>master-max</b>	可將多少資源複本推送至 <b>master</b> 狀態；預設值為 1。
<b>master-node-max</b>	在一個節點上可將多少資源複本推送至 <b>master</b> 狀態；預設值為 1。

### 8.2.1. 監控多狀態資源

若只要為 master 資源新增監控作業，您可新增一項額外監控作業至該資源。然而請注意，資源上的每項監控作業皆必須使用不同的間隔。

下列範例會在 master 資源上為 **ms\_resource** 配置一項間隔為 11 秒的監控作業。此作業乃預設監控間隔為 10 秒鐘的預設監控作業以外的監控作業。

```
# pcs resource op add ms_resource interval=11s role=Master
```

### 8.2.2. 多狀態限制式

在大部分情況下，多狀態的資源在各個啓用的叢集節點上皆會有一個複本。若非如此，您可使用資源位置限制式來決定叢集應將複本指定給哪些節點。這些限制式和一般資源的限制式編寫上並無差異。

欲取得更多有關於資源位置限制式上的相關資訊，請參閱 [〈節 6.1, “位置限制式”](#)。

您可建立一項 colocation 限制式，以指定資源是 master 或是 slave 資源。下列指令會建立一項資源 colocation 限制式。

```
pcs constraint colocation add [master|slave] source_resource with
[master|slave] target_resource [score] [options]
```

欲取得 colocation 限制式上的相關資訊，請參閱 [〈節 6.3, “資源的 Colocation”](#)。

當配置一項包含了多狀態資源的順序限制式時，您可為資源指定的其中一項動作為 **promote**，代表資源將由 slave 升為 master。此外，您亦可指定一項 **demote** 動作，這代表資源將由 master 降為 slave。

配置順序限制式的指令如下。

```
pcs constraint order [action] resource_id then [action] resource_id
[options]
```

欲取得更多有關於資源順序限制式上的相關資訊，請參閱 [〈節 6.2, “順序限制”](#)。

### 8.2.3. 多狀態綁定

為了能擁有穩定的配置模式，多狀態資源就預設值會被綁定。若未提供 **resource-stickiness** 的值，多狀態資源將會使用 1 這個值。因為這個值較小，因此不會造成其它資源分數計算上的干擾，而卻足以預防 Pacemaker 在不必要的情況下在叢集之間移動複本。

### 8.3. 監控資源的事件通知

Pacemaker 叢集是個事件導向的系統，事件可能會是資源錯誤或是配置變更。`ocf:pacemaker:ClusterMon` 資源能監控叢集狀態並根據各項叢集事件啓動警示。這項資源會以正常間隔在背景中執行 `crm_mon` 並使用 `crm_mon` 的功能來傳送電子郵件訊息 (SMTP)。它亦可透過 `extra_options` 參數來執行一項外部程式。

下列範例會配置一項名為 `ClusterMon-SMTP`，並且會傳送電子郵件通知的 `ClusterMon` 資源。Pacemaker 事件會使電子郵件透過 `mail.example.com` 郵件主機來從 `pacemaker@nodeX.example.com` 將郵件寄至 `pacemaker@example.com`。這項資源會被以複本形式建立，如此一來它便能在叢集中的各個節點上執行。

```
# pcs resource create ClusterMon-SMTP ClusterMon user=root update=30 \
extra_options="-T pacemaker@example.com -F pacemaker@nodeX.example.com \
-P PACEMAKER -H mail.example.com" --clone
```

下列範例會配置一項名為 `ClusterMon-External` 的 `ClusterMon` 資源，它會執行 `/root/bin/bin.priv/logger.sh` 程式，這項程式會判斷該如何對應叢集通知做出反應。`ClusterMon` 會將叢集狀態輸出至一個 `html` 檔案中，在此範例中為 `/var/www/html/cluster_mon.html`。`pidfile` 會偵測 `ClusterMon` 是否早已在執行；在此範例中，該檔案為 `/var/run/crm_mon-external.pid`。這項資源會以複本形式建立，如此一來它便能在叢集中的所有節點上執行。

下列指令會建立這項資源將會使用的 `logger.sh` 程式。

```
# touch /root/bin/bin.priv/logger.sh
# chmod 755 /root/bin/bin.priv/logger.sh
# cat /root/bin/bin.priv/logger.sh
#!/bin/sh
logger "crm_mon: CRM_MON triggered this message.";
exit;
```

以下指令會建立監控資源。

```
# pcs resource create ClusterMon-External ClusterMon user=root \
update=30 extra_options="-E /root/bin/bin.priv/logger.sh" \
htmlfile=/var/www/html/cluster_mon.html \
pidfile=/var/run/crm_mon-external.pid --clone
```

### 8.4. PACEMAKER\_REMOTE 服務

`pacemaker_remote` 服務能允許未執行 `corosync` 的節點整合入叢集，並讓叢集管理其資源，如同真實的叢集節點一般。這代表 Pacemaker 叢集現在已能不在虛擬環境中執行 `pacemaker` 或 `corosync` 的情況下，管理虛擬環境 (KVM/LXC) 以及在這些虛擬環境中的資源。

下列為描述 `pacemaker_remote` 服務的各項名詞。

- **叢集節點** - 一個執行 High Availability 服務 (`pacemaker` 和 `corosync`) 的節點。
- **遠端節點** — 一個執行 `pacemaker_remote` 的節點，以在不需要 `corosync` 叢集成員身份的情況下遠端整合入叢集中。

- **容器** — 一項包含了額外資源的 Pacemaker 資源。例如一項包含了網站伺服器資源的 KVM 虛擬機器資源。
- **容器遠端節點** — 一個執行 `pacemaker_remote` 服務的虛擬客座端遠端節點。它詳述了特定的遠端節點使用案例，在此情況下叢集所管理的虛擬客座資源會由叢集所啟用，並作為遠端節點整合入叢集之中。
- `pacemaker_remote` — 一項能夠在 Pacemaker 叢集環境中與獨立（非叢集）環境中的客座節點（KVM 和 LXC）內執行遠端應用程式管理的服務 daemon。這項服務乃 Pacemaker 本機資源管理 daemon（LRMD）的增強版本，它能在遠端客座上管理和監控 LSB、OCF、upstart 以及 systemd 資源。它亦能允許 `pcs` 在遠端節點上運作。
- **LXC** — 一個由 `libvirt-lxc` Linux 容器驅動程式所定義的 Linux Container。

一個執行 `pacemaker_remote` 服務的 Pacemaker 叢集會擁有下列特性。

- 執行 `pacemaker_remote` 服務的虛擬遠端節點（在虛擬機器端僅需要進行非常少量的配置）。
- 執行於叢集節點上的叢集堆疊（`pacemaker` 和 `corosync`）會啟動虛擬機器並即刻連上 `pacemaker_remote` 服務，允許虛擬機器整合入叢集中。

虛擬機器遠端節點與叢集節點之間的關鍵差異在於遠端節點不會執行叢集堆疊。這代表遠端節點不會被仲裁。這也意味著遠端節點不會被綁定至與叢集堆疊相聯的伸縮性限制。就資源管理而言，除了仲裁限制之外，遠端節點的行為會與叢集節點相同。叢集擁有完整的能力以管理和監控各個遠端節點上的資源。您可根據遠端節點建立限制式，使其處於待命模式，或進行任何其它您在叢集節點上所進行的動作。遠端節點會如同叢集節點一般地出現在叢集狀態輸出中。

### 8.4.1. 容器遠端節點資源選項

當配置虛擬機器或是 LXC 資源使其作為遠端節點運作時，您可建立一項用來管理虛擬機器的 `VirtualDomain` 資源。請使用下列指令來查看您能為 `VirtualDomain` 資源設置的選項。

```
# pcs resource describe VirtualDomain
```

除了 `VirtualDomain` 資源選項以外，您可配置 `metadata` 選項來啟用資源作為遠端節點和定義連線參數。表格 8.3, “用來將 KVM/LXC 資源配置為遠端節點的 `Metadata` 選項” 詳述了這些 `metadata` 選項。

表格 8.3. 用來將 KVM/LXC 資源配置為遠端節點的 `Metadata` 選項

欄位	預設	詳述
<code>remote-node</code>	<none>	這項資源定義的遠端節點名稱。這會啟用資源來作為遠端節點和定義使用來辨識遠端節點的獨特名稱。若未設置其它參數，這個值也會被視為連接埠 3121 上，將被連至的主機名稱。警告：這個值不可與任何資源或節點 ID 重疊。
<code>remote-port</code>	3121	配置一個自訂連接埠，以讓客座連上 <code>pacemaker_remote</code> 。
<code>remote-addr</code>	用來作為主機名稱的 <code>remote-node</code> 值	若遠端名稱非客座主機名稱，需連至的 IP 位址或主機名稱

欄位	預設	詳述
<code>remote-connect-timeout</code>	60s	等待處理的客座連線逾時前所需等待的時間

下列指令會建立一項 `VirtualDomain` 名為 `vm-guest1` 的資源，它是個能透過使用 `remote-node meta` 屬性來執行資源的遠端節點。

```
# pcs resource create vm-guest1 VirtualDomain hypervisor="qemu:///system"
config="vm-guest1.xml" meta remote-node=guest1
```

#### 8.4.2. 主機與客座認證

叢集節點與遠端節點之間的連線認證與加密乃透過使用 TLS 搭配 PSK 加密/認證於 TCP 連接埠 3121 上來達成的。這代表叢集節點與遠端節點必須共享相同的私密金鑰。就預設值，這組金鑰必須放置在叢集節點與遠端節點上的 `/etc/pacemaker/authkey` 中。

#### 8.4.3. 更改預設的 `pacemaker_remote` 選項

若您需要更改 Pacemaker 或是 `pacemaker_remote` 的預設連接埠或 `authkey` 位置，您可設置幾項能影響這兩個 daemon 的環境變數。這些環境變數可如下放置在 `/etc/sysconfig/pacemaker` 檔案中來啟用。

```
### Pacemaker Remote
# Use a custom directory for finding the authkey.
PCMK_authkey_location=/etc/pacemaker/authkey
#
# Specify a custom port for Pacemaker Remote connections
PCMK_remote_port=3121
```

#### 8.4.4. 配置總覽：KVM 遠端節點

這部分提供了能讓 Pacemaker 啟用虛擬機器，並透過 `libvirt` 與 KVM 虛擬客座來將該機器整合為遠端節點的基本概要總覽。

1. 在安裝了虛擬化軟體並在叢集節點上啟用了 `libvirtd` 服務後，在所有叢集節點與虛擬機器上將一個 `authkey` 放置在 `/etc/pacemaker/authkey` 中。這將能確保遠端通訊與認證。

下列指令會建立一項 `authkey`。

```
# dd if=/dev/urandom of=/etc/pacemaker/authkey bs=4096 count=1
```

2. 在所有虛擬機器上安裝 `pacemaker_remote` 套件，啟用 `pacemaker_remote` 服務並使它能在開機時執行，並在防火牆中允許開啓 TCP 連接埠 3121。

```
# yum install pacemaker-remote resource-agents
# systemctl start pacemaker_remote.service
# systemctl enable pacemaker_remote.service
# firewall-cmd --add-port 3121/tcp --permanent
```

3. 為各個虛擬機器提供靜態的網路位址與獨特的主機名稱。
4. 若要建立 **VirtualDomain** 資源代理程式來管理虛擬機器，Pacemaker 需要虛擬機器的 xml 配置檔案傾印到一個檔案或磁碟上。比方說，若您建立了一個名為 **guest1** 的虛擬機器，請使用下列指令來將 xml 傾印至主機上的某個檔案中。

```
# virsh dumpxml guest1 > /virtual_machines/guest1.xml
```

5. 建立 **VirtualDomain** 資源，配置 **remote-note** 資源的 meta 選項來顯示虛擬機器是個能執行資源的遠端節點。

在以下範例中，meta 屬性 **remote-node=guest1** 會告知 pacemaker 這項資源是主機名稱為 **guest1** 的遠端節點，並且它會被整合入叢集中。叢集會嘗試聯絡虛擬機器的 **pacemaker\_remote** 服務於主機名稱 **guest1**（在它啓用後）。

```
# pcs resource create vm-guest1 VirtualDomain
hypervisor="qemu:///system" config="vm-guest1.xml" meta remote-
node=guest1
```

6. 在建立了 **VirtualDomain** 資源後，您可將遠端節點視為叢集中的任何其它節點使用。比方說，您可建立一項資源，並在遠端節點上執行的資源上設置資源限制式。

```
# pcs resource create webserver apache params
configfile=/etc/httpd/conf/httpd.conf op monitor interval=30s
# pcs constraint location webserver prefers guest1
```

當遠端節點整合入叢集之後，您便能由遠端節點本身執行 **pcs** 指令，就如同遠端節點執行了 Pacemaker 一般。

## 章 9. PACEMAKER 規則

規則可被使用來令您的配置更加動態化。一項常見的實例就是在工作期間為 **resource-stickiness** 設置一個值，以避免資源被移回其最為偏好的位置，而針對週末沒有人會注意到服務中斷的情況另外設置一個值。

規則的另一使用方法就是根據時間將機器指定至不同的處理群組（使用一個節點屬性）然後在建立位置限制式時使用該屬性。

各項規則皆可包含數個表示式甚至是其它規則。表示式的結果會根據規則的 **boolean-op** 欄位結合，以判斷規則最後結果會是 **true** 或是 **false**。接下來會發生的動作則取決於規則所使用於的情況。

表格 9.1. 規則的內容

欄位	詳述
<b>role</b>	將規則限制為僅在資源屬於該角色時才套用。允許的值有： <b>Started</b> 、 <b>Slave</b> ，和 <b>Master</b> 。請注意：一項 <b>role="Master"</b> 的規則無法決定複本的初始位置。它僅會影響將被升階的啓用中項目為何。
<b>score</b>	若規則為 <b>true</b> 時所套用的分數值。僅限制使用於屬於位置限制式一部分的規則中。
<b>score-attribute</b>	規則為 <b>true</b> 時所查詢並使用來作為分數值的節點屬性。僅限制使用於屬於位置限制式一部分的規則中。
<b>boolean-op</b>	結合多重表示式物件之結果的方式。允許的值為： <b>and</b> 和 <b>or</b> 。預設值為 <b>and</b> 。

### 9.1. 節點屬性表示式

節點屬性表示式會被使用來根據節點所定義的屬性來控制一項資源。

表格 9.2. 表示式的內容

欄位	詳述
<b>value</b>	使用者所提供來進行比較的值
<b>attribute</b>	欲測試的節點屬性
<b>type</b>	判斷如何測試值。允許的值有： <b>string</b> 、 <b>integer</b> 、 <b>version</b>



欄位	詳述
<b>operation</b>	<p>欲進行的比較。允許的值有：</p> <ul style="list-style-type: none"> <li>* <b>lt</b> - 若節點屬性的值小於 <b>value</b> 則為 true</li> <li>* <b>gt</b> - 若節點屬性的值大於 <b>value</b> 則為 true</li> <li>* <b>lte</b> - 若節點屬性的值小於或等於 <b>value</b> 則為 true</li> <li>* <b>gte</b> - 若節點屬性的值大於或等於 <b>value</b> 則為 true</li> <li>* <b>eq</b> - 若節點屬性的值等於 <b>value</b> 則為 true</li> <li>* <b>ne</b> - 若節點屬性的值不等於 <b>value</b> 則為 true</li> <li>* <b>defined</b> - 若節點含有 <b>named</b> 屬性則為 true</li> <li>* <b>not_defined</b> - 若節點沒有 <b>named</b> 屬性則為 true</li> </ul>

## 9.2. 基於時間/日期的表示式

日期表示式主要被使用來根據目前的日期/時間來控制一項資源或叢集選項。它們能包含一項選用性的日期規格。

表格 9.3. 日期表示式的內容

欄位	詳述
<b>start</b>	符合 ISO8601 規格的日期/時間。
<b>end</b>	符合 ISO8601 規格的日期/時間。
<b>operation</b>	<p>根據內容比較目前的日期/時間和起始與/或結束日期。允許的值有：</p> <ul style="list-style-type: none"> <li>* <b>gt</b> - 若目前的日期/時間已經過了 <b>start</b> 則為 true</li> <li>* <b>lt</b> - 若目前的日期/時間在 <b>end</b> 之前則為 true</li> <li>* <b>in-range</b> - 若目前的日期/時間在 <b>start</b> 之後而在 <b>end</b> 之前則為 true</li> <li>* <b>date-spec</b> - 針對於目前的日期/時間進行一項類似 cron 的比較</li> </ul>

## 9.3. 日期規格

日期規格會被用來建立類似 cron、時間相關的表示式。各個欄位皆可包含單一數字或單獨時間範圍。與其預設為零，任何未提供的欄位都會被忽略。

比方說，**monthdays="1"** 相當於每個月的第一天，而 **hours="09-17"** 相當於 9am 和 5pm 之間（包含 9am 與 5pm）的小時數。然而，您不可指定 **weekdays="1,2"** 或 **weekdays="1-2,5-6"** 因為它們包含了多重範圍。

表格 9.4. 日期規格內容

欄位	詳述
<b>id</b>	日期的獨特識別子
<b>hours</b>	允許的值：0-23
<b>monthdays</b>	允許的值：0-31（視月份與年份而定）
<b>weekdays</b>	允許的值：1-7（1=星期一、7=星期天）
<b>yeardays</b>	允許的值：1-366（視年份而定）
<b>months</b>	允許的值：1-12
<b>weeks</b>	允許的值：1-53（視 weekyear 而定）
<b>years</b>	根據格里曆而定的年份
<b>weekyears</b>	可能會與格里曆的年份不同；比方說， <b>2005-001 Ordinal</b> 同時是 <b>2005-01-01 Gregorian</b> ，並且同時也是 <b>2004-W53-6 Weekly</b>
<b>moon</b>	允許的值有：0-7（0 代表新月，4 為滿月）。

## 9.4. 期間

「期間」會被用來計算 **end** 的值（當未向 **in\_range** 作業提供時）。它們包含了與 **date\_spec** 物件相同的欄位，不過卻無限制（例如您能將期間設為 19 個月）。如 **date\_specs** 一般，任何未提供的欄位皆會被忽略。

## 9.5. 透過 PCS 配置規則

若要配置一項規則，請使用下列指令。若 **score** 被忽略，它將會被預設為 INFINITY。若 **id** 被忽略，**constraint\_id** 將會產生一組 ID。**rule\_type** 應該要是 **expression** 或 **date\_expression**。

```
pcs constraint rule add constraint_id [rule_type] [score=score
[id=rule_id] expression|date_expression|date_spec options
```

若要移除一項規則，請使用下列指令。若您要移除的規則為其限制式中的最後一項規則，該限制式將會被移除。

```
pcs constraint rule remove rule_id
```

## 9.6. 基於時間的表示式（範本）

若現在時間為 2005 年內任何時間，下列指令便會配置一項為 true 的表示式。

```
# pcs constraint location webserver rule score=INFINITY date-spec
years=2005
```

■  
 下列指令會配置一項表示式並且 9am 至 5pm、Monday 至 Friday 為 true。請注意能與 16 這個小時值搭配的時間可達 16:59:59，因為此數值（小時）依然符合此範圍。

```
# pcs constraint location Webserver rule score=INFINITY date-spec
hours="9-16" weekdays="1-5"
```

下列指令會配置一項表示式，這項表示式若在 13 號星期五遇上圓月的話，其狀態將會是 true。

```
# pcs constraint location Webserver rule date-spec weekdays=5 monthdays=13
moon=4
```

## 9.7. 使用規則來判斷資源位置

您可藉由下列指令來使用一項規則以判斷資源的位置。

```
pcs resource constraint location resource_id rule [rule_id]
[role=master|slave] [score=score expression]
```

*expression* 能夠是以下其中之一：

- **defined|not\_defined *attribute***
- ***attribute* lt|gt|lte|gte|eq|ne *value***
- **date [start=*start*] [end=*end*] operation=gt|lt|in-range**
- **date-spec *date\_spec\_options***

## 章 10. PACEMAKER 叢集內容

叢集內容能控制叢集在進行作業中遇到特定狀況時的行為。

- [表格 10.1, “叢集內容”](#) 詳述了叢集內容選項。
- [節 10.2, “設定與移除叢集內容”](#) 詳述了如何設置叢集內容。
- [節 10.3, “查詢叢集內容設定”](#) 詳述了如何列出目前設置的叢集內容。

### 10.1. 叢集內容與選項的概要

[表格 10.1, “叢集內容”](#) 提供了 Pacemaker 叢集內容的摘要，並顯示了內容的預設值，以及您能為這些內容設置的值。



#### 注意

除了在此表格中詳述的內容之外，叢集軟體還有其它可使用的額外叢集內容。建議您不要修改這些內容的預設值。

表格 10.1. 叢集內容

選項	預設	詳述
<b>batch-limit</b>	30	遷移引擎 (transition engine, TE) 允許平行執行的工作數量。「正確」的值將取決於您網路和叢集節點的速度和負載。
<b>migration-limit</b>	-1 (無限)	TE 允許在一個節點上平行執行的遷移工作數量。
<b>no-quorum-policy</b>	stop	叢集無仲裁時所進行的動作。允許的值有： <ul style="list-style-type: none"> <li>* ignore - 繼續所有的資源管理</li> <li>* freeze - 繼續資源管理，不過不復原來自不屬於受影響分割區之節點的資源</li> <li>* stop - 停止受影響之叢集分割區中的所有資源</li> <li>* suicide - 隔離受影響之叢集分割區中的所有節點</li> </ul>
<b>symmetric-cluster</b>	true	顯示資源就預設值是否能在任何節點上執行。
<b>stonith-enabled</b>	true	顯示失效節點以及含有無法停下的資源的節點皆應被隔離。若您要保護您的資料，您需將此設為 <b>true</b> 。  若是 <b>true</b> 或為設置，叢集將會拒絕啓用資源，除非一或更多項 STONITH 資源也已經過配置。
<b>stonith-action</b>	reboot	傳送至 STONITH 裝置的動作。允許的值有： <b>reboot</b> 、 <b>off</b> 。您也允許使用 <b>poweroff</b> 這個值，不過這僅適用於繼承裝置。
<b>cluster-delay</b>	60s	網路上的來回時間延遲 (不包含動作執行)。「正確」的值取決於您網路和叢集節點的速度與負載。

選項	預設	詳述
<b>stop-orphan-resources</b>	true	顯示刪除的資源是否應被停止。
<b>stop-orphan-actions</b>	true	顯示刪除的動作是否應被取消。
<b>start-failure-is-fatal</b>	true	會顯示資源啓用失敗的情況是否會被視為嚴重錯誤。當設為 <b>false</b> 時，叢集將會使用資源的 <b>failcount</b> 和 <b>migration-threshold</b> 的值來代替。欲取得設定一項資源的 <b>migration-threshold</b> 選項上的相關資訊，請參閱〈節 7.2, “在發生錯誤時移動資源”〉。
<b>pe-error-series-max</b>	-1 (全部)	儲存 ERROR 中的 PE 輸入的次數。使用於回報問題時。
<b>pe-warn-series-max</b>	-1 (全部)	儲存 WARNING 中的 PE 輸入的次數。使用於回報問題時。
<b>pe-input-series-max</b>	-1 (全部)	儲存「normal」PE 輸入的次數。使用於回報問題時。
<b>cluster-infrastructure</b>		Pacemaker 目前執行於的訊息堆疊。使用於資訊及診斷用途；使用者不可進行配置。
<b>dc-version</b>		叢集的指定控制器 (Designated Controller, DC) 的版本。使用於診斷用途；使用者不可進行配置。
<b>last-lrm-refresh</b>		本機資源管理程式最後更新時，以在 epoca 經過後的秒數為單位。使用於診斷用途；使用者不可進行配置。
<b>cluster-recheck-interval</b>	60	選項、資源參數與限制式基於時間之變更的輪詢間隔。允許的值有：零會停用輪詢，正數則為間隔 (秒) 除非指定了其它 SI 單位，例如 5min。
<b>default-action-timeout</b>	20s	一項 Pacemaker 動作的逾時值。資源本身的作業設定總是會比設定為叢集選項的預設值優先。
<b>maintenance-mode</b>	false	Maintenance Mode (維護模式) 會使叢集進入一項「hands off」的模式中，並且不啓用或停用任何服務，直到被告知。當維護模式完成時，叢集會針對所有服務的目前狀態進行一項例行性檢查，然後啓用或停用任何需要它的服務。
<b>shutdown-escalation</b>	20min	放棄嘗試正常關機並直接退出前所需等待的時間。僅適用於進階使用者。
<b>stonith-timeout</b>	60s	等待 STONITH 動作完成的所需時間。
<b>stop-all-resources</b>	false	叢集是否應該停止所有資源。
<b>default-resource-stickiness</b>	5000	顯示資源偏好滯留的位置。建議您將這個值設為資源/作業的預設值，而非叢集選項。

選項	預設	詳述
<b>is-managed-default</b>	true	顯示叢集是否允許啓用和停用一項資源。建議您將這個值設為資源/作業預設值，而非叢集選項。
<b>enable-acl</b>	false	(Red Hat Enterprise Linux 6.6 和更新版本) 顯示叢集是否能使用存取控制清單，如透過 <b>pcs acl</b> 指令所設置一般。

## 10.2. 設定與移除叢集內容

若要設置叢集內容的值，請使用以下 **pcs** 指令。

```
pcs property set property=value
```

比方說，若要將 **symmetric-cluster** 的值設為 **false**，請使用以下指令。

```
# pcs property set symmetric-cluster=false
```

您可透過以下指令由配置中移除叢集內容。

```
pcs property unset property
```

此外，您亦可藉由將 **pcs property set** 指令的值欄位保留為空白來將叢集內容從配置中移除。這會將該內容還原為其預設值。比方說，若您先前將 **symmetric-cluster** 內容設為了 **false**，下列指令便會將您由配置設置的值移除，並將 **symmetric-cluster** 的值還原成 **true**，也就是其預設值。

```
# pcs property set symmetric-cluster=
```

## 10.3. 查詢叢集內容設定

在大部分情況下，當您使用 **pcs** 指令來顯示各種叢集元件的值時，您可交互使用 **pcs list** 或 **pcs show**。在下列範例中，**pcs list** 乃用來顯示超過一個內容的所有設定的格式，而 **pcs show** 則為用來顯示特定內容的值的格式。

若要顯示已為叢集設置的內容設定值，請使用以下 **pcs** 指令。

```
pcs property list
```

若要顯示叢集內容設定所有的值（包括尚未明確設置的內容設定預設值）請使用以下指令。

```
pcs property list --all
```

若要顯示特定叢集內容目前的值，請使用以下指令。

```
pcs property show property
```

比方說，若要顯示 **cluster-infrastructure** 內容目前的值，請執行下列指令：

```
# pcs property show cluster-infrastructure
Cluster Properties:
```

---

```
cluster-infrastructure: cman
```

若要取得資訊，您可藉由下列指令顯示內容的所有預設值，無論它們是否已被設為一個預設值以外的值。

```
pcs property [list|show] --defaults
```

## 附錄 A. RED HAT ENTERPRISE LINUX RELEASE 6.5 與 RED HAT ENTERPRISE LINUX RELEASE 6.6（和更新版本）的叢集建立

若要透過 Pacemaker 在 Red Hat Enterprise Linux 6.6（與更新版本）中配置一項 Red Hat High Availability Cluster，您需要一組和在 Red Hat Enterprise Linux 6 中，透過 **rgmanager** 來配置叢集時所使用的不同配置工具與管理介面。〈節 A.1, “透過 **rgmanager** 和 Pacemaker 建立叢集”〉概述了各種叢集元件之間的不同配置。

Red Hat Enterprise Linux 6.6 發行版為 Pacemaker 提供了一些用來進行叢集配置的新功能。〈節 A.2, “在 Red Hat Enterprise Linux Release 6.5 與 Red Hat Enterprise Linux Release 6.6（和更新版本）中透過 Pacemaker 建立叢集”〉概述了 Red Hat Enterprise Linux release 6.5 中的 **pcs** 支援，與 Red Hat Enterprise Linux release 6.6（與更新版本）中的 **pcs** 支援之間的微幅配置差異。

### A.1. 透過 RGMANAGER 和 PACEMAKER 建立叢集

表格 A.1, “比較透過 **rgmanager** 與 Pacemaker 所進行的叢集配置” 提供了有關於如何在 Red Hat Enterprise Linux release 6.6（與更新版本）中使用 **rgmanager** 和 Pacemaker 來配置叢集元件的廣泛資訊。

表格 A.1. 比較透過 **rgmanager** 與 Pacemaker 所進行的叢集配置

配置元件	rgmanager	Pacemaker
叢集配置檔案	各個節點上的叢集配置檔案為 <b>cluster.conf</b> 檔案，若您希望的話您亦可直接編輯該檔案。否則，請使用 <b>luci</b> 或 <b>ccs</b> 介面來定義叢集配置。	叢集與 Pacemaker 的配置檔案為 <b>cluster.conf</b> 與 <b>cib.xml</b> 。請勿直接編輯這些檔案；請使用 <b>pcs</b> 介面來代替。
網路設定	在配置叢集之前先配置 IP 位址與 SSH。	在配置叢集之前先配置 IP 位址與 SSH。
叢集配置工具	<b>luci</b> 、 <b>ccs</b> 指令，手動編輯 <b>cluster.conf</b> 檔案。	<b>pcs</b>
安裝	安裝 <b>rgmanager</b> （這會拉進所有相依性項目，包括 <b>ricci</b> 、 <b>luci</b> 和資源與隔離代理程式）。若必要的話，請安裝 <b>lvm2-cluster</b> 與 <b>gfs2-utils</b> 。	安裝 <b>pacemaker</b> 、 <b>cman</b> 、 <b>pcs</b> 以及您所需要的資源和隔離代理程式。若必要的話，請安裝 <b>lvm2-cluster</b> 和 <b>gfs2-utils</b> 。



配置元件	rgmanager	Pacemaker
<p>啓動叢集服務</p>	<p>透過下列程序來啓動並啓用叢集服務：</p> <ol style="list-style-type: none"> <li>啓用 <b>rgmanager</b>、<b>cman</b>，並且若有必要的話請啓用 <b>clvmd</b> 和 <b>gfs2</b>。</li> <li>啓用 <b>ricci</b>，若使用 <b>luci</b> 介面的話請啓用 <b>luci</b>。</li> <li>為需要的服務執行 <b>chkconfig on</b>，已讓它們能在 runtime 時啓動。</li> </ol> <p>此外，您亦可執行 <b>ccs --start</b> 來開始並啓用叢集服務。</p>	<p>透過下列程序來啓動並啓用叢集服務：</p> <ol style="list-style-type: none"> <li>在每個節點上執行 <b>service pcsd start</b>，然後執行 <b>service pcsd enable</b> 以在 runtime 時開始啓用 <b>pcsd</b>。</li> <li>在叢集的其中一個節點上執行 <b>pcs cluster start --all</b> 以啓用 <b>cman</b> 和 <b>pacemaker</b>。</li> </ol>
<p>控制配置工具的存取</p>	<p>使用 <b>luci</b> 時，root 或持有 <b>luci</b> 權限的使用者皆能存取 <b>luci</b>。所有存取皆需要使用節點的 <b>ricci</b> 密碼。</p>	<p>配置 GUI 不存在。</p>
<p>叢集建立</p>	<p>為叢集命名並透過 <b>luci</b> 或 <b>ccs</b> 來定義叢集中需包含哪些節點，或直接編輯 <b>cluster.conf</b> 檔案。</p>	<p>為叢集命名並透過 <b>pcs cluster setup</b> 指令來加入節點。</p>
<p>將叢集配置散佈至所有節點上</p>	<p>當透過 <b>luci</b> 來配置叢集時，散佈機制將會是自動的。當使用 <b>ccs</b> 時，請搭配 <b>--sync</b> 選項使用。您亦可使用 <b>cman_tool version -r</b> 指令。</p>	<p>叢集與 Pacemaker 配置檔案 <b>cluster.conf</b> 和 <b>cib.xml</b> 的散佈會在設定叢集或新增資源時自動進行。</p>
<p>全域叢集屬性</p>	<p><b>rgmanager</b> 支援搭配下列功能：</p> <ul style="list-style-type: none"> <li>* 您可配置系統使其能選擇在叢集網路中要使用哪組 multicast 位址來進行 IP multicasting。</li> <li>* 若無法使用 IP multicasting，您可使用 UDP Unicast 傳輸機制。</li> <li>* 您能配置叢集使其使用 RRP 協定。</li> </ul>	<p>Pacemaker 支援下列的叢集相關功能：</p> <ul style="list-style-type: none"> <li>* 您可為叢集設置 <b>no-quorum-policy</b>，以指定系統應在叢集未仲裁進行什麼動作。</li> <li>* 欲取得更多您可設置的額外叢集內容，請參閱〈表格 10.1, “叢集內容”〉。</li> </ul>
<p>記錄</p>	<p>您可設置全域和 daemon 特屬的紀錄配置。</p>	<p>欲取得有關於如何手動配置紀錄上的相關資訊，請參閱 <b>/etc/sysconfig/pacemaker</b> 檔案。</p>
<p>驗證叢集</p>	<p>使用 <b>luci</b> 和 <b>ccs</b> 時，叢集驗證會自動透過叢集結構描述進行。叢集會在啓用時自動驗證。</p>	<p>叢集會在啓用時自動驗證，或您亦可透過 <b>pcs cluster verify</b> 來驗證叢集。</p>

配置元件	rgmanager	Pacemaker
雙節點叢集中的仲裁	<p>當使用雙節點的叢集時，您可配置系統該如何判斷仲裁：</p> <ul style="list-style-type: none"> <li>* 配置仲裁磁碟</li> <li>* 使用 <b>ccs</b> 或編輯 <b>cluster.conf</b> 檔案以設置 <b>two_node=1</b> 和 <b>expected_votes=1</b> 並允許單一節點維持仲裁。</li> </ul>	<p><b>pcs</b> 會自動將雙節點叢集的必要選項加到 <b>cman</b>。</p>
叢集狀態	<p>在 <b>luci</b> 上，叢集的目前狀態可在介面的各個元件中看見，並且您亦可更新它。您可使用 <b>ccs</b> 指令的 <b>--getconf</b> 選項來查看目前的配置檔案。您可使用 <b>clustat</b> 指令來顯示叢集狀態。</p>	<p>您可透過 <b>pcs status</b> 來顯示目前的叢集狀態。</p>
資源	<p>您可透過 <b>luci</b> 或 <b>ccs</b> 指令，或是藉由編輯 <b>cluster.conf</b> 配置檔案來新增已定義類型的資源以及配置資源特屬的內容。</p>	<p>您可透過 <b>pcs resource create</b> 來新增已定義類型的資源，以及配置資源特屬的內容。欲取得透過 Pacemaker 來配置叢集資源上的相關資訊，請參閱 <a href="#">〈章 5, 配置叢集資源〉</a>。</p>
資源行為、分組，以及起始/停止順序	<p>定義叢集服務以配置資源如何進行互動。</p>	<p>使用 Pacemaker 時，您可使用資源群組來作為定義一組資源的捷徑方式，這些資源需要位在相同的位置上，並且循序性地起始與停止。此外，您需透過下列方式來定義資源的行為，及其進行互動的方式：</p> <ul style="list-style-type: none"> <li>* 您需將部分資源行為作為資源選項來設置。</li> <li>* 您需使用位置限制式來判斷資源能在哪些節點上運作。</li> <li>* 您需使用順序限制式來判斷資源執行的順序。</li> <li>* 您可使用 <b>colocation</b> 限制式來判斷一項資源的位置是否相依於另一項資源。</li> </ul> <p>欲取得更多關於這些主題上的相關資訊，請參閱 <a href="#">〈章 5, 配置叢集資源〉</a>。</p>
資源管理：移動、起始、停止資源	<p>透過 <b>luci</b>，您可管理叢集、個別叢集節點，以及叢集服務。您可透過 <b>ccs</b> 指令來管理叢集。您可使用 <b>clusvadm</b> 來管理叢集服務。</p>	<p>您可暫時性停用一個節點，如此一來它便無法透過 <b>pcs cluster standby</b> 指令來提供資源（因為這會造成資源遷移）。您可透過 <b>pcs resource disable</b> 指令來停止一項資源。</p>
完全移除一項叢集配置	<p>使用 <b>luci</b> 時，您可選擇叢集中的所有節點，並刪除整個叢集。您亦可將 <b>cluster.conf</b> 從叢集中的各個節點上移除。</p>	<p>您可透過 <b>pcs cluster destroy</b> 指令來將叢集配置從一個節點上移除。</p>

配置元件	rgmanager	Pacemaker
------	-----------	-----------

在多重節點上運作的資源，以及在多重節點上以多重模式運作的資源	無等位	使用 Pacemaker 時，您可複製資源，如此一來它們便能在多重節點上運作，並且您可定義複本資源 (cloned resource) 為 master 與 slave 資源，如此一來它們便能在多重模式下運作。欲取得更多有關於複本資源和 master/slave 資源上的相關資訊，請參閱 <a href="#">章 8, 進階資源類型</a> 。
隔離 -- 為各個節點配置單一隔離裝置	建立全域或本機隔離裝置，並將它們加入節點。您可為整體叢集定義 <b>post-fail delay</b> 和 <b>post-join delay</b> 值。	透過 <b>pcs stonith create</b> 指令來為各個節點建立隔離裝置。當使用可隔離多重節點的裝置時，您僅需要定義它們一次，無需為各個節點獨立進行。您亦可定義 <b>pcmk_host_map</b> 以透過單一指令來為所有節點配置隔離裝置；欲取得有關於 <b>pcmk_host_map</b> 上的相關資訊，請參閱 <a href="#">〈表格 4.1, “隔離裝置的一般內容”</a> 〉。您可為整體叢集定義 <b>stonith-timeout</b> 值。
為各個節點設置多重 (備份) 隔離裝置	透過 <b>luci</b> 或 <b>ccs</b> 指令，或直接透過編輯 <b>cluster.conf</b> 檔案來定義備份裝置。	配置隔離等級。

## A.2. 在 RED HAT ENTERPRISE LINUX RELEASE 6.5 與 RED HAT ENTERPRISE LINUX RELEASE 6.6 (和更新版本) 中透過 PACEMAKER 建立叢集

若要在 Red Hat Enterprise Linux 6.5 中建立 Pacemaker 叢集，您必須建立叢集並在叢集中的各個節點上啟用叢集服務。比方說，若要建立一個名為 **my\_cluster**、包含了 **z1-rhel65.example.com** 與 **z2-rhel65.example.com** 節點的叢集，並在這些節點上啟用叢集服務，請由 **z1-rhel65.example.com** 與 **z2-rhel65.example.com** 執行下列指令。

```
[root@z1-rhel65]# pcs cluster setup --name my_cluster \
z1-rhel65.example.com z2-rhel65.example.com
[root@z1-rhel65]# pcs cluster start
```

```
[root@z2-rhel65]# pcs cluster setup --name my_cluster \
z1-rhel65.example.com z2-rhel65.example.com
[root@z2-rhel65]# pcs cluster start
```

在 Red Hat Enterprise Linux 6.6 (與更新版本) 中，您需在叢集中的一個節點上執行叢集建立指令。若僅在一個節點上執行下列指令，將會建立一個名為 **my\_cluster**、包含了 **z1-rhel66.example.com** 與 **z2-rhel66.example.com** 節點的叢集，並在這些節點上啟用叢集服務。

```
[root@z1-rhel66]# pcs cluster setup --start --name my_cluster \
z1-rhel66.example.com z2-rhel66.example.com
```

■

## 附錄 B. 透過 PCS 指令進行配置的範例

此附錄提供了一項逐步程序，用來在 Red Hat Enterprise Linux release 6.6 和更新版本上透過 **pcs** 指令配置一個雙節點的 Red Hat Enterprise Linux High Availability Add-On 叢集。它亦詳述了如何在此叢集中配置一個 Apache 網站伺服器。

若要配置本章之中所提供的叢集，您的系統必須包含下列元件：

- 兩個將被用來建立叢集的節點。在此範例中，所使用的節點為 **z1.example.com** 與 **z2.example.com**。
- 用於私密網路的網路切換器，需透過此硬體，叢集節點與其它叢集硬體（例如網路電源切換器和光纖頻道切換器）之間才能進行通訊。
- 供叢集中各個節點使用的電源隔離裝置。此範例使用了兩個 APC 電源切換器的連接埠，並且主機名稱為 **zapc.example.com**。

### B.1. 初始系統設定

這部分詳述了您將會使用來建立叢集的系統的初始設定。

#### B.1.1. 安裝叢集軟體

使用下列程序來安裝叢集軟體。

1. 確認 **pacemaker**、**cman** 以及 **pcs** 已安裝。

```
yum install -y pacemaker cman pcs
```

2. 安裝完成後，若要避免 **corosync** 在未使用 **cman** 便啓用的情況發生，請在叢集中的所有節點上執行下列指令。

```
# chkconfig corosync off
```

#### B.1.2. 建立和啓用叢集

這部分提供了用來建立初始叢集的程序，而您將會在該叢集上配置叢集資源。

1. 若要使用 **pcs** 來配置叢集並在節點之間進行通訊，您必須在各個節點上為 **hacluster** 這組使用者 ID 設置一組密碼，這是 **pcs** 的管理帳號。建議在各個節點上皆使用相同的 **hacluster** 使用者密碼。

```
# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

2. 在您有辦法配置叢集之前，**pcsd** daemon 必須先啓用。此 daemon 能與 **pcs** 指令搭配使用，以管理叢集節點之間的配置。

請在叢集中的各個節點上執行下列指令來啓用 **pcsd** 服務並在系統啓動時啓用 **pcsd**。

-

```
# service pcsd start
# chkconfig pcsd on
```

3. 在您將會執行 **pcs** 的節點上，為叢集中的各個節點認證 **pcs** 的使用者 **hacluster**。

下列指令將會認證範例雙節點叢集 (**z1.example.com** 和 **z2.example.com**) 中，兩個節點的 **z1.example.com** 上的使用者 **hacluster**。

```
root@z1 ~]# pcs cluster auth z1.example.com z2.example.com
Username: hacluster
Password:
z1.example.com: Authorized
z2.example.com: Authorized
```

4. 由 **z1.example.com** 執行下列指令以建立包含了 **z1.example.com** 與 **z2.example.com** 節點的雙節點叢集 **mycluster**。這會將叢集配置檔案散佈到叢集中的兩個節點上。這項指令包含了 **--start** 選項，這會在叢集中的兩個節點上啟用叢集服務。

```
[root@z1 ~]# pcs cluster setup --start --name my_cluster \
z1.example.com z2.example.com
z1.example.com: Succeeded
z1.example.com: Starting Cluster...
z2.example.com: Succeeded
z2.example.com: Starting Cluster...
```

5. 您亦可選用性地啟用叢集服務以使其在節點啟動時，在叢集中的各個節點上運作。



### 注意

您可選擇藉由跳過此步驟，以使您特定環境中的叢集服務保持停用。這能讓您確保在節點停機時，所有您叢集或資源上的問題，都能在節點重新加入叢集之前解決。若您將叢集服務保留為停用，當您重新啟動一個節點時，您將需要在該節點上透過執行 **pcs cluster start** 指令來手動式啟用服務。

```
# pcs cluster enable --all
```

您可透過 **pcs cluster status** 指令來顯示叢集的目前狀態。

```
[root@z1 ~]# pcs cluster status
Cluster Status:
Last updated: Thu Jul 25 13:01:26 2013
Last change: Thu Jul 25 13:04:45 2013 via crmd on z2.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
0 Resources configured
```

## B.2. 隔離配置

您必須為叢集中的各個節點配置隔離裝置。欲取得更多有關於配置隔離裝置上的一般資訊，請參閱〈[章 4，隔離：配置 STONITH](#)〉。



### 注意

當配置隔離裝置時，您應確認您的隔離裝置未與它所控制的節點共享電源。

此範例使用了 APC 電源切換器（主機名稱為 `zpc.example.com`）來隔離節點，並且它使用了 `fence_apc_snmp` 隔離代理程式。因為這兩個節點將會被相同的隔離代理程式所隔離，您可藉由使用 `pcmk_host_map` 和 `pcmk_host_list` 選項來將這兩個隔離裝置作為單一資源來配置。

您可藉由 `pcs stonith create` 指令以透過將裝置配置為 `stonith` 資源來建立隔離裝置。下列指令將配置名為 `myapc` 的 `stonith` 資源，這項資源將在 `z1.example.com` 和 `z2.example.com` 節點上使用 `fence_apc_snmp` 隔離代理程式。`pcmk_host_map` 選項會將 `z1.example.com` 對映到連接埠 1 上，並將 `z2.example.com` 對映到連接埠 2 上。APC 裝置的登錄值與密碼皆為 `apc`。就預設值，此裝置將會針對各個節點使用一個 60 秒的監控間隔。

請注意，您能在為節點指定主機名稱時使用一組 IP 位址。

```
[root@z1 ~]# pcs stonith create myapc fence_apc_snmp params \
ipaddr="zpc.example.com"
pcmk_host_map="z1.example.com:1;z2.example.com:2" \
pcmk_host_check="static-list"
pcmk_host_list="z1.example.com,z2.example.com" \
login="apc" passwd="apc"
```



### 注意

當您建立一個 `fence_apc_snmp stonith` 裝置時，您可能會看見下列警告訊息，您可安全地忽略此訊息：

```
Warning: missing required option(s): 'port, action' for
resource type: stonith:fence_apc_snmp
```

下列指令顯示了一個既有的 STONITH 裝置的參數。

```
[root@rh7-1 ~]# pcs stonith show myapc
Resource: myapc (class=stonith type=fence_apc_snmp)
Attributes: ipaddr=zpc.example.com
pcmk_host_map=z1.example.com:1;z2.example.com:2 pcmk_host_check=static-
list pcmk_host_list=z1.example.com,z2.example.com login=apc passwd=apc
Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

## B.3. 透過 PCS 指令來在一個 RED HAT HIGH AVAILABILITY CLUSTER 中配置 APACHE WEB SERVER

此部分詳述了如何在一個雙節點的 Red Hat Enterprise Linux High Availability Add-On 叢集中配置一個 Apache 網站伺服器，並使用 `pcs` 來配置叢集資源。在此範例中，客戶端會透過一組浮動 IP 位址來存取 Apache 網站伺服器。這個網站伺服器會在叢集中的其中一個節點上執行。若網站伺服器所執行於的節點停機的話，網站伺服器將會在叢集中的第二個節點上重新啟動，避免干擾服務。

這個範例假設您的系統已包含了下列元件：

- 一個雙節點的 Red Hat High Availability 叢集，並且各個節點皆配置了電源隔離。這項程序使用了〈節 B.1.2, “建立和啓用叢集”〉中所提供的叢集範例。
- 一個 Apache 網站伺服器所需的公共虛擬 IP 位址。
- 供叢集中的節點使用的共享儲存裝置（使用 iSCSI 或光纖頻道）。

配置了一個 Apache 資源群組的叢集，它包含了網站伺服器所需要的叢集元件：一項 LVM 資源、檔案系統資源、一組 IP 位址資源，以及一項網站伺服器資源。此資源群組能由叢集中的一個節點備援至另一節點，並允許這兩個節點執行網站伺服器。在為此叢集建立資源群組前，您將必須執行下列程序：

1. 如〈節 B.3.1, “為一個 LVM 卷冊配置 ext4 檔案系統”〉中所述，配置一項 **ext4** 檔案系統並將它掛載至 **my\_lv** 邏輯卷冊上。
2. 如〈節 B.3.2, “網站伺服器配置”〉中所述，配置一個網站伺服器。
3. 確保只有叢集能啓用包含了 **my\_lv** 的卷冊群組，並且開機時該卷冊群組不會在叢集之外啓用，如〈節 B.3.3, “獨佔啓用 (exclusive activation) 叢集中的卷冊群組”〉中所述。

再進行了這些程序後，您便可建立資源群組以及它所包含的資源（如節 B.3.4, “透過 pcs 指令來建立資源和資源群組”〉中所述。

### B.3.1. 為一個 LVM 卷冊配置 ext4 檔案系統

這項範例需要您在儲存裝置上建立一個 LVM 邏輯卷冊，並且該儲存裝置能讓叢集中的節點共享。

下列程序會建立一個 LVM 邏輯卷冊然後在該卷冊上建立一個 **ext4** 檔案系統。在此範例中，共享的分割區 **/dev/sdb1** 會被使用來儲存 LVM 實體卷冊，並從而建立 LVM 邏輯卷冊。



#### 注意

叢集節點所使用的 LVM 卷冊和相應分割區以及裝置必須只連至該叢集節點。

因為 **/dev/sdb1** 分割區乃共享儲存裝置，您僅需要在一個節點上進行這項程序即可。

1. 在 **/dev/sdb1** 分割區上建立一個 LVM 實體卷冊。

```
# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created
```

2. 建立包含了實體卷冊 **/dev/sdb1** 的卷冊群組 **my\_vg**。

```
# vgcreate my_vg /dev/sdb1
Volume group "my_vg" successfully created
```

3. 建立一個使用了卷冊群組 **my\_vg** 的邏輯卷冊。

```
# lvcreate -L450 -n my_lv my_vg
Rounding up size to full physical extent 452.00 MiB
Logical volume "my_lv" created
```



您可使用 **lvs** 指令來顯示邏輯卷冊。

```
# lvs
LV          VG          Attr          LSize   Pool Origin Data%  Move Log
Copy%  Convert
my_lv      my_vg      -wi-a---- 452.00m
...
```

4. 在邏輯卷冊 **my\_lv** 上建立一個 **ext4** 檔案系統。

```
# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

### B.3.2. 網站伺服器配置

下列程序將配置一個 Apache 網站伺服器。

1. 確認 Apache HTTPD 伺服器已安裝在叢集中的各個節點上。您也必須在叢集上安裝 **wget** 工具，以檢查 Apache 網站伺服器的狀態。

請在各個節點上執行以下指令。

```
# yum install -y httpd wget
```

2. 若要讓 Apache 資源代理程式能夠取得 Apache 網站伺服器的狀態，請確認下列文字是否有出現在叢集中各個節點上的 **/etc/httpd/conf/httpd.conf** 檔案中，並確認這些文字未被標記為註解。若這些文字不存在，請將它們附加至檔案尾端。

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

3. 建立一個網頁來讓 Apache 提供服務。請在叢集中的一個節點上掛載您在〈節 B.3.1, “為一個 LVM 卷冊配置 ext4 檔案系統”〉中建立的檔案系統、在該檔案系統上建立 **index.html** 檔案，然後卸載此檔案系統。

```
# mount /dev/my_vg/my_lv /var/www/
# mkdir /var/www/html
# mkdir /var/www/cgi-bin
# mkdir /var/www/error
# restorecon -R /var/www
# cat <<-END >/var/www/html/index.html
<html>
<body>Hello</body>
```

```
</html>
END
# umount /var/www
```

### B.3.3. 獨佔啓用 (exclusive activation) 叢集中的卷冊群組

下列程序能配置卷冊群組，以確保只有叢集能夠啓用卷冊群組，並且卷冊群組將無法在開機時於叢集外啓用。若卷冊群組被叢集外的系統啓用的話，卷冊群組的 metadata 將會有損毀的風險。

#### 注意

您必須確認使用 Pacemaker 時，**lvm** daemon 已停用。您可藉由執行下列指令來檢查 daemon 是否已停用，以及是否有任何 **lvm** 程序正在執行中。

```
# grep use_lvm /etc/lvm/lvm.conf
use_lvm = 0
# ps -ef | grep -i [l]vm
root      23843 15478  0 11:31 pts/0    00:00:00 grep --
color=auto -i lvm
```

若您取得的結果與此範例不同，請在 `/etc/lvm/lvm.conf` 檔案中設置 `use_lvm = 0`，並停止任何執行中的 **lvm** 程序。

此程序會修改 `/etc/lvm/lvm.conf` 配置檔案中的 `volume_list` 項目。列在 `volume_list` 項目中的卷冊群組允許在叢集管理員控制之外的本機節點上自動啓用。與節點的本機 root 和家目錄相關的卷冊群組應包含在此清單中。所有由叢集管理員所管理的卷冊群組皆必須排除在 `volume_list` 項目之外。請注意，此程序無需使用到 `clvmd`。

在叢集各個節點上執行以下程序。

1. 透過以下指令來判斷哪些卷冊群組目前已配置於您的本機儲存裝置上。這將會輸出一列目前已配置的卷冊群組。若您在此節點上為 root 和您的家目錄分配了獨立的卷冊群組，您將會在輸出中看見這些卷冊，如以下範例所示。

```
# vgs --noheadings -o vg_name
my_vg
rhel_home
rhel_root
```

2. 在 `/etc/lvm/lvm.conf` 配置檔案中，加入 `my_vg` (您剛為叢集定義的卷冊群組) 以外的卷冊群組作為 `volume_list` 的項目。比方說，若您在獨立的卷冊群組中為 root 與您的家目錄分配了空間，您將需要反註解 `lvm.conf` 檔案中的 `volume_list` 一行，並如下將這些卷冊群組作為 `volume_list` 的項目加入：

```
volume_list = [ "rhel_root", "rhel_home" ]
```

#### 注意

若在叢集管理員之外啓動的節點上沒有卷冊群組的話，您還是必須將 `volume_list` 項目初始化為 `volume_list = []`。

3. 重建 `initramfs` 開機映像檔以確保開機映像檔不會嘗試啓用一個由叢集所控制的卷冊群組。請透過以下指令更新 `initramfs` 裝置。這項指令要完成可能會花上一分鐘。

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

4. 重新啓動節點。



### 注意

若您在啓動了一部建立了開機映像檔的節點後安裝了一個新的 Linux kernel，新的 `initrd` 映像檔將會屬於您建立該映像檔時正在執行的 kernel，而不是您重新啓動節點時執行的新 kernel。您能藉由在重新開機之前或之後執行 `uname -r` 指令來判斷正在執行的 kernel 版本，並確認是否正在使用正確的 `initrd` 裝置。若版本不一樣的話，請在以新的 kernel 重新啓動後更新 `initrd` 檔案，然後重新啓動節點。

5. 當節點重新啓動後，請藉由在此節點上執行 `pcs cluster status` 指令，以檢查叢集服務是否有再次啓用於該節點上。若出現了 **Error: cluster is not currently running on this node** 錯誤訊息的話，請執行以下指令：

```
# pcs cluster start
```

此外，您能等到您重新啓動了叢集中的各個節點後，透過以下指令啓用各個節點上的叢集服務。

```
# pcs cluster start --all
```

### B.3.4. 透過 pcs 指令來建立資源和資源群組

這個使用案例需要您建立四個叢集資源。為了確保這些資源皆執行於相同節點上，它們將配置為資源群組 `apachegroup` 的一部分。需建立的資源如下，以它們啓用的順序列出。

1. 一項名為 `my_lvm` 的 LVM 資源，它使用了您在〈節 B.3.1, “為一個 LVM 卷冊配置 ext4 檔案系統”〉中所建立的 LVM 卷冊群組。
2. 一項名為 `my_fs` 的 Filesystem 資源，它使用了您在〈節 B.3.1, “為一個 LVM 卷冊配置 ext4 檔案系統”〉中所建立的檔案系統裝置 `/dev/my_vg/my_lv`。
3. 一項 `IPaddr2` 資源，它乃 `apachegroup` 資源群組的浮動 IP 位址。IP 位址必須是一組不與實體節點相聯的位址。若 `IPaddr2` 資源的 NIC 裝置未指定，浮動 IP 便必須駐存在與叢集節點所使用的靜態指定之 IP 位址相同的網路上，否則將無法正常偵測欲為其分配浮動 IP 位址的 NIC 裝置。
4. 一項名為 `Website` 的 `apache` 資源，它使用了 `index.html` 檔案以及您在〈節 B.3.2, “網站伺服器配置”〉中定義的 Apache 配置。

下列程序將建立資源群組 `apachegroup` 以及該群組包含的資源。資源將會以您將其新增至該群組中的順序開始，並且將會以反向的順序停止。請只在叢集的一個節點上執行這項程序。

1. 下列指令會建立 LVM 資源 `my_lvm`。這項指令將指定 `exclusive=true` 參數以確保只有叢集能夠啓用 LVM 邏輯卷冊。因為資源群組 `apachegroup` 尚未存在，這項指令將會建立資源群組。

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg \
exclusive=true --group apachegroup
```

當您建立資源時，資源會自動啓用。您可使用以下指令來確認資源是否已建立並已啓用。

```
# pcs resource show
Resource Group: apachegroup
my_lvm (ocf::heartbeat:LVM): Started
```

您可透過 **pcs resource disable** 和 **pcs resource enable** 指令來手動停用和啓用一項資源。

2. 下列指令會為配置建立剩餘的資源，並將它們加入到既有的資源群組 **apachegroup** 中。

```
[root@z1 ~]# pcs resource create my_fs Filesystem \
device="/dev/my_vg/my_lv" directory="/var/www" fstype="ext4" --group \
\
apachegroup

[root@z1 ~]# pcs resource create VirtualIP IPAddr2 ip=198.51.100.3 \
cidr_netmask=24 --group apachegroup

[root@z1 ~]# pcs resource create Website apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://127.0.0.1/server-status" --group apachegroup
```

3. 在建立了資源以及包含這些資源的資源群組後，您可檢查叢集的狀態。請注意所有這四項資源皆在相同的節點上執行。

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 16:38:51 2013
Last change: Wed Jul 31 16:42:14 2013 via crm_attribute on
z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured

Online: [ z1.example.com z2.example.com ]

Full list of resources:
myapc (stonith:fence_apc_snmp): Started z1.example.com
Resource Group: apachegroup
my_lvm (ocf::heartbeat:LVM): Started z1.example.com
my_fs (ocf::heartbeat:Filesystem): Started z1.example.com
VirtualIP (ocf::heartbeat:IPAddr2): Started z1.example.com
Website (ocf::heartbeat:apache): Started z1.example.com
```

請注意，若您尚未為您的叢集配置一項隔離裝置（如 [節 B.2, “隔離配置”](#) 中所述），就預設值這些資源將不會啓用。

4. 當叢集啓動後，您可在瀏覽器中輸入您定義為 **IPAddr2** 資源的 IP 位址，以檢視包含了單一「Hello」字串的範本。

```
Hello
```

若您發現您配置的資源沒有在運作，您可執行 `pcs resource debug-start 資源` 指令來測試資源配置。欲取得有關於 `pcs resource debug-start` 指令上的相關資訊，請參閱《*High Availability Add-On 參照*》指南。

### B.3.5. 測試資源配置

在〈節 B.3.4, “透過 pcs 指令來建立資源和資源群組”〉中顯示的叢集狀態裡，所有資源皆在節點 `z1.example.com` 上執行。您可藉由使用下列程序來將第一個節點設為 `standby` 模式（在這之後該節點將無法再為資源提供服務），以測試資源群組是否會備援至節點 `z2.example.com`。

1. 下列指令會使 `z1.example.com` 節點進入 `standby` 模式中。

```
root@z1 ~]# pcs cluster standby z1.example.com
```

2. 在將 `z1` 節點設為 `standby` 模式後，請檢查叢集狀態。請注意，所有資源現在都應在 `z2` 上運作。

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 17:16:17 2013
Last change: Wed Jul 31 17:18:34 2013 via crm_attribute on
z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured

Node z1.example.com (1): standby
Online: [ z2.example.com ]

Full list of resources:

myapc (stonith:fence_apc_snmp): Started z1.example.com
Resource Group: apachegroup
  my_lvm (ocf::heartbeat:LVM): Started z2.example.com
  my_fs (ocf::heartbeat:Filesystem): Started z2.example.com
  VirtualIP (ocf::heartbeat:IPaddr2): Started z2.example.com
  Website (ocf::heartbeat:apache): Started z2.example.com
```

所定義之 IP 位址的網站應該會繼續顯示而不會受到干擾。

3. 若要使 `z1` 從 `standby` 模式中退出，請執行下列指令。

```
root@z1 ~]# pcs cluster unstandby z1.example.com
```



#### 注意

將節點退出 `standby` 模式不會造成資源備援至該節點。欲取得有關於如何控制資源能在哪個節點上執行，請參閱《*Red Hat High Availability Add-On 參照指南*》中有關於配置叢集資源的章節。

## 附錄 C. 修訂記錄

<b>修訂 3.0-8.3</b> 翻譯、校閱完成	<b>Wed Aug 5 2015</b>	<b>Terry Chuang</b>
<b>修訂 3.0-8.2</b> 讓翻譯檔案與 XML 來源 3.0-8 同步	<b>Wed Aug 5 2015</b>	<b>Terry Chuang</b>
<b>修訂 3.0-8.1</b> 讓翻譯檔案與 XML 來源 3.0-8 同步	<b>Mon Aug 3 2015</b>	<b>Terry Chuang</b>
<b>修訂 3.0-8</b> 重新發行了 RHEL 6.7 版本	<b>Thu Jul 23 2015</b>	<b>Steven Levine</b>
<b>修訂 3.0-6</b> 6.7 GA 發行版本	<b>Wed Jul 8 2015</b>	<b>Steven Levine</b>
<b>修訂 3.0-4</b> 6.7 Beta 發行版本	<b>Fri Apr 17 2015</b>	<b>Steven Levine</b>
<b>修訂 3.0-1</b> 解決了 #1211680 已從您能為隔離裝置設置的內容之表格中移除了 stonith-timeout 參數 解決了 #1202957 加入了新增節點至叢集的部分 解決了 #1208652 在配置範例中加入了有關於 exclusive LVM 啓用的部分，並記載了 lvmtool daemon 必須停用	<b>Fri Mar 27 2015</b>	<b>Steven Levine</b>
<b>修訂 2.0-7</b> 更新以符合 RHEL 6 歡迎頁的排序功能。	<b>Tue Dec 16 2014</b>	<b>Steven Levine</b>
<b>修訂 2.0-5</b> 6.6 GA 發行版本。	<b>Thu Oct 9 2014</b>	<b>Steven Levine</b>
<b>修訂 2.0-4</b> 解決了 #1131544 新增了 ACL 的文件	<b>Wed Oct 8 2014</b>	<b>Steven Levine</b>
<b>修訂 2.0-2</b> 6.6 Beta 發行版本	<b>Wed Aug 7 2014</b>	<b>Steven Levine</b>
<b>修訂 2.0-1</b> 6.6 Beta 草稿  解決了 #1126896, #1126018, #986462, #1045406, #1122145, #1079340 小幅訂正與技術性內容修正。  解決了 #1081225, #1081248, #1092720 更新了有關於支援配置節點之間的同步，與其它 internode 通訊功能的文件。	<b>Wed Jul 23 2014</b>	<b>Steven Levine</b>
<b>修訂 1.1-2</b> 6.5 GA 發行版本	<b>Wed Nov 20 2013</b>	<b>Steven Levine</b>
<b>修訂 0.1-4</b> 6.5 beta 草稿首次印刷	<b>Wed Oct 2 2013</b>	<b>Steven Levine</b>

