



# Red Hat Enterprise Linux

## 7

# High Availability 外掛程式管理

---

配置與管理 High Availability 外掛程式



配置與管理 High Availability 外掛程式

## 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](https://creativecommons.org/licenses/by-sa/3.0/). 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

《High Availability Add-On 管理》詳述了在 Red Hat Enterprise Linux 7 上配置與管理 High Availability 外掛程式的相關資訊。

---

## 內容目錄

<b>章 1. 透過 Pacemaker 來建立 Red Hat High-Availability Cluster</b> .....	<b>2</b>
1.1. 叢集軟體安裝	2
1.2. 建立叢集	3
1.3. 隔離配置	4
<b>章 2. 一個在 Red Hat High Availability Cluster 中的主動/被動 Apache Web Server</b> .....	<b>6</b>
2.1. 為一個 LVM 卷冊配置 ext4 檔案系統	7
2.2. 網站伺服器配置	8
2.3. 獨佔啓用 (exclusive activation) 叢集中的卷冊群組	8
2.4. 透過 pcs 指令來建立資源和資源群組	10
2.5. 測試資源配置	11
<b>章 3. 一部在 Red Hat High Availability Cluster 中的主動/被動 NFS Server</b> .....	<b>13</b>
3.1. 建立 NFS 叢集	13
3.2. 為一個 LVM 卷冊配置 ext4 檔案系統	13
3.3. NFS Share 設定	14
3.4. 獨佔啓用 (exclusive activation) 叢集中的卷冊群組	15
3.5. 配置叢集資源	16
3.6. 測試資源配置	19
<b>附錄 A. 修訂記錄</b> .....	<b>22</b>

# 章 1. 透過 Pacemaker 來建立 Red Hat High-Availability Cluster

本章詳述了透過 **pcs** 建立 Red Hat High Availability 雙節點叢集的程序。在您建立了叢集之後，您便可配置您需要的資源和資源群組。

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**。

本章分為三個部分。

- ※ [節 1.1, “叢集軟體安裝”](#) 提供了安裝叢集軟體的程序。
- ※ [節 1.2, “建立叢集”](#) 提供了配置雙節點叢集的程序。
- ※ [節 1.3, “隔離配置”](#) 提供了為各個叢集節點配置隔離裝置的程序。

## 1.1. 叢集軟體安裝

安裝和配置叢集的程序如下。

1. 在叢集中的各個節點上安裝 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-all
```

2. 若您正在執行 **firewalld** daemon，請執行下列指令來啓用 Red Hat High Availability Add-On 所需的連接埠。



### 注意

您可透過 **rpm -q firewalld** 指令來判斷 **firewalld** daemon 是否安裝在您的系統上。若 **firewalld** daemon 已安裝，您可透過 **firewall-cmd --state** 指令來判斷它是否正在執行。

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

3. 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.
```

4. 在您能夠配置叢集之前，**pcsd** daemon 必須啓用並且能在各個節點啓動時開始運作。這個 daemon 能與 **pcs** 指令搭配使用來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**。

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

5. 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
```

## 1.2. 建立叢集

此程序會建立一個包含 **z1.example.com** 與 **z2.example.com** 節點的 Red Hat High Availability Add-On 叢集。

1. 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 **my\_cluster**。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啓用叢集服務。

```
[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...
```

2. 啓用叢集服務以使其在節點啓動時，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，當您重新啓動一個節點時，您將需要在該節點上透過執行 **pcs cluster start** 指令來手動式啓用服務。

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

您可透過 **pcs cluster status** 指令來顯示叢集的目前狀態。由於在使用 **pcs cluster setup** 指令的 **--start** 選項啓動叢集服務時，會有些許延遲，所以在對叢集執行任何後續動作、進行配置之前，請確定叢集已經啓用並處於執行狀態。

```
[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
```

### 1.3. 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《Red Hat Enterprise Linux 7 High Availability Add-On 參考指南》。



#### 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
```

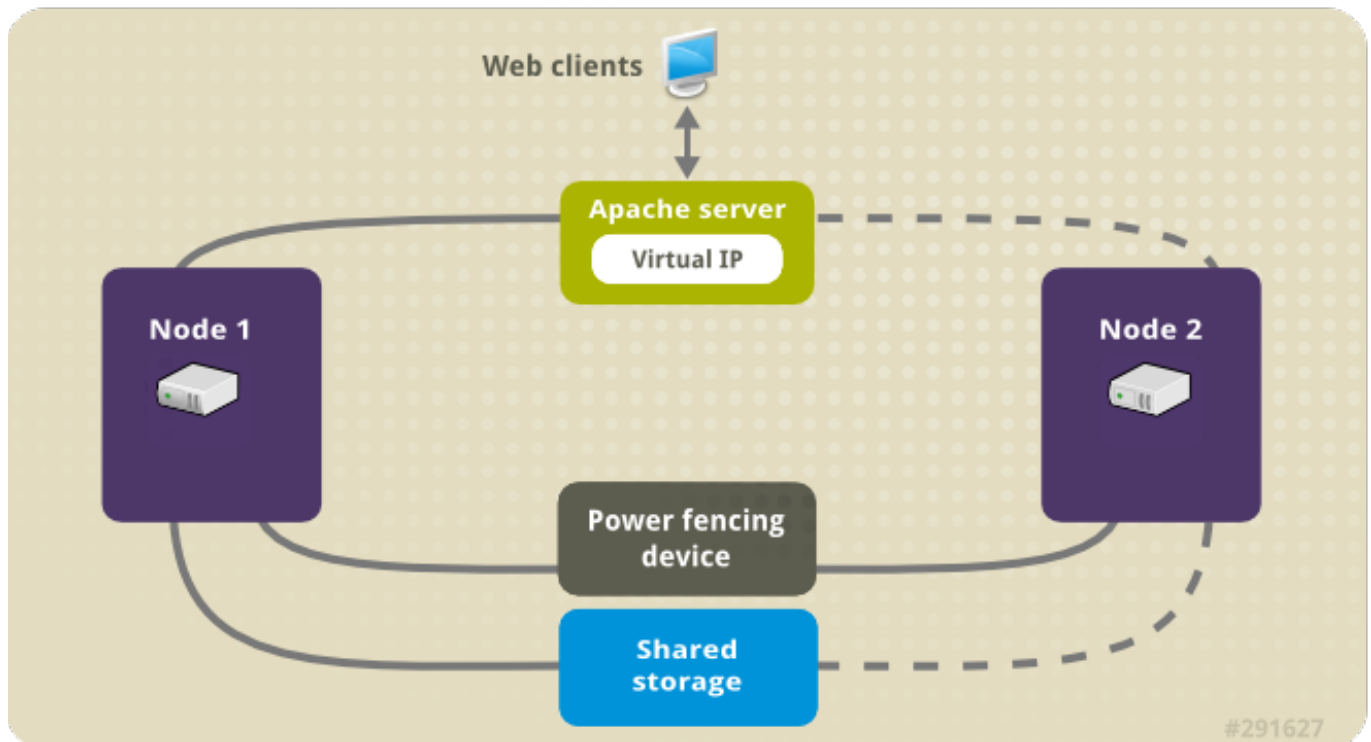


```
Resource: myapc (class=stonith type=fence_apc_snmp)
Attributes: ipaddr=zapc.example.com
pcmk_host_map=z1.example.com:1;z2.example.com:2 pcmk_host_check=static-list
pcmk_host_list=z1.example.com,z2.example.com login=apc passwd=apc
Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

## 章 2. 一個在 Red Hat High Availability Cluster 中的主動/被動 Apache Web Server

本章詳述了如何在一個雙節點的 Red Hat Enterprise Linux High Availability Add-On 叢集中配置一項主動/被動式的 Apache 網站伺服器，並使用 **pcs** 來配置叢集資源。在此使用案例中，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啓動，避免干擾服務。

[圖形 2.1, “在 Red Hat High Availability 雙節點叢集中的 Apache 網站伺服器”](#) 顯示了高層的叢集總覽。該叢集是個雙節點的 Red Hat High-Availability 叢集，並且配置了網路電源切換器以及共享儲存裝置。叢集節點連上了公用網路，以讓客戶端透過虛擬 IP 存取 Apache 網站伺服器。Apache 伺服器會在 Node 1 或是 Node 2 上執行，這兩個節點皆能存取存放了 Apache 資料的儲存裝置。



圖形 2.1. 在 Red Hat High Availability 雙節點叢集中的 Apache 網站伺服器

本使用案例假設您的系統包含了下列元件：

- ✦ 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 [〈章 1, 透過 Pacemaker 來建立 Red Hat High-Availability Cluster〉](#) 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 2.1, “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 2.2, “網站伺服器配置”](#) 中所述，配置一個網站伺服器。

3. 確保只有叢集能啟用包含了 `my_lv` 的卷冊群組，並且開機時該卷冊群組不會在叢集之外啟用，如〈[節 2.3, “獨佔啟用 \(exclusive activation\) 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 2.4, “透過 pcs 指令來建立資源和資源群組”](#)〉中所述。

## 2.1. 為一個 LVM 卷冊配置 ext4 檔案系統

這項使用案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
...
```

## 2.2. 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 2.1, “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
```

## 2.3. 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。

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. 執行以下指令以確定 `/etc/lvm/lvm.conf` 檔案中的 `locking_type` 設為 1，且 `use_lvmetad` 設為 0。這指令也會立即停用、停止任何 `lvmetad` 程序。

```
# lvmconf --enable-halvm --services --startstopservices
```

- 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
```

- 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啟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 = []`。

- 重建 `initramfs` 開機映像檔以確保開機映像檔不會嘗試啟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)
```

- 重新啟動節點。



### 注意

若您在啟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，而不是您重新啟動節點時執行的新 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 重新啟動後更新 `initrd` 檔案，然後重新啟動節點。

- 當節點重新啟動後，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，您能等到您重新啟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
```

## 2.4. 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 2.1, “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 2.1, “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 2.2, “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 1.3, “隔離配置”](#) 中所述），就預設值這些資源將不會啟用。

- 當叢集啟動後，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並未運作，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*》手冊。

## 2.5. 測試資源配置

在 [〈節 2.4, “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**。

- 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
```

- 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。



## 章 3. 一部在 Red Hat High Availability Cluster 中的主動/被動 NFS Server

本章詳述了如何在一個使用了共享儲存裝置的雙節點 Red Hat Enterprise Linux High Availability Add-On 叢集上，配置一項高可用性的主動/被動 NFS 伺服器。這項程序使用了 **pcs** 來配置一項 Pacemaker 叢集資源。在此使用案例中，客戶端會透過一組浮動 IP 位址來存取 NFS 檔案系統。此 NFS 伺服器會在叢集中的其中一個節點上執行。若 NFS 所執行於的節點停機的話，NFS 伺服器將會在叢集中的第二個節點上重新啓動，避免干擾服務。

本使用案例假設您的系統包含了下列元件：

- ✦ 兩個將被用來建立執行 Apache 網站伺服器的叢集的節點。在此範例中，所使用的節點為 **z1.example.com** 與 **z2.example.com**。
- ✦ 供 webfarm 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 **zpc.example.com**。
- ✦ 一組 NFS 伺服器所需的公用虛擬 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）。

若要在一個雙節點的 Red Hat Enterprise Linux High Availability 叢集上配置一個高可用性的主動/被動 NFS 伺服器，您需要進行下列步驟。

1. 建立能執行 NFS 伺服器的叢集，並為該叢集中的各個節點配置隔離（如 [節 3.1, “建立 NFS 叢集”](#) 中所述）。
2. 如 [〈節 3.2, “為一個 LVM 卷冊配置 ext4 檔案系統”](#) 中所述，配置一個 **ext4** 檔案系統並將它掛載至叢集節點的共享儲存裝置上的 LVM 邏輯卷冊 **my\_lv** 上。
3. 在 LVM 邏輯卷冊上的共享儲存裝置上配置一個 NFS share，如 [〈節 3.3, “NFS Share 設定”](#) 中所述。
4. 確保只有叢集能啓用包含了 **my\_lv** 邏輯卷冊的 LVM 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 3.4, “獨佔啓用 \(exclusive activation\) 叢集中的卷冊群組”](#) 中所述。
5. 如 [〈節 3.5, “配置叢集資源”](#) 中所述建立叢集資源。
6. 如 [〈節 3.6, “測試資源配置”](#) 中所述測試您所配置的 NFS 伺服器。

### 3.1. 建立 NFS 叢集

使用下列程序來安裝並建立 NFS 叢集。

1. 透過 [〈節 1.1, “叢集軟體安裝”](#) 中所提供的程序來在 **z1.example.com** 和 **z2.example.com** 節點上安裝叢集軟體。
2. 透過 [〈節 1.2, “建立叢集”](#) 中所提供的程序來建立包含了 **z1.example.com** 和 **z2.example.com** 的雙節點 webfarm 叢集。如範例程序，此使用範例將叢集命名為 **my\_cluster**。
3. 使用 [〈節 1.3, “隔離配置”](#) 中所提供程序來為 webfarm 叢集各個節點配置隔離裝置。此範例使用了 APC 電源切換器的兩個連接埠來配置隔離，並使用了 **zpc.example.com** 這組主機名稱。

### 3.2. 為一個 LVM 卷冊配置 ext4 檔案系統

這項使用案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。

```
[root@z1 ~]# 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**。

```
[root@z1 ~]# 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。

```
[root@z1 ~]# 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。

```
[root@z1 ~]# 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。

```
[root@z1 ~]# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

## 3.3. NFS Share 設定

下列程序將配置 NFS share 以提供 NFS daemon 備援。您需要僅需要在叢集中的一個節點上執行這項程序。

1. 建立 **/nfsshare** 目錄。

```
[root@z1 ~]# mkdir /nfsshare
```

- 將您在〈[節 3.2, “為一個 LVM 卷冊配置 ext4 檔案系統”](#)〉中建立的 **ext4** 檔案系統掛載在 **/nfsshare** 目錄上。

```
[root@z1 ~]# mount /dev/my_vg/my_lv /nfsshare
```

- 在 **/nfsshare** 目錄上建立一個 **exports** 目錄樹。

```
[root@z1 ~]# mkdir -p /nfsshare/exports
[root@z1 ~]# mkdir -p /nfsshare/exports/export1
[root@z1 ~]# mkdir -p /nfsshare/exports/export2
```

- 將檔案放置在 **exports** 目錄中以讓 NFS 客戶端存取。在此範例中，我們將建立名為 **clientdatafile1** 和 **clientdatafile2** 的測試檔案。

```
[root@z1 ~]# touch /nfsshare/exports/export1/clientdatafile1
[root@z1 ~]# touch /nfsshare/exports/export2/clientdatafile2
```

- 卸載 **ext4** 檔案系統並停用 LVM 卷冊群組。

```
[root@z1 ~]# umount /dev/my_vg/my_lv
[root@z1 ~]# vgchange -an my_vg
```

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

下列程序能配置 LVM 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。

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。

- 執行以下指令以確定 **/etc/lvm/lvm.conf** 檔案中的 **locking\_type** 設為 1，且 **use\_lvmetad** 設為 0。這指令也會立即停用、停止任何 **lvmetad** 程序。

```
# lvmconf --enable-halvm --services --startstopservices
```

- 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
```

- 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 = []`。

4. 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)
```

5. 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。

6. 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，透過以下指令啓用叢集中所有節點上的叢集服務。

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

## 3.5. 配置叢集資源

本部分提供了為此使用案例配置叢集資源的程序。


**注意**

建議當您透過 `pcs resource create` 建立叢集資源後，馬上執行 `pcs status` 指令以驗證資源是否有在運作。請注意，若您尚未為您的叢集配置隔離裝置（如〈[節 1.3, “隔離配置”](#)〉中所述），就預設值資源將不會啟用。

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。這將會啟用叢集控制之外的服務。在這時已配置的資源會再次運作，請執行 `pcs cluster cleanup resource` 以使叢集偵測到更新。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。

下列程序將會配置系統資源。為了確保這些資源全部皆會在相同節點上執行，它們將被配置為 `nfsgroup` 資源群組的一部分。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. 下列指令會建立名為 `my_lvm` 的 LVM 資源。這項指令將指定 `exclusive=true` 參數以確保只有叢集能夠啟用 LVM 邏輯卷冊。因為資源群組 `nfsgroup` 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 nfsgroup
```

檢查叢集的狀態以驗證資源是否有在執行。

```
root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Thu Jan  8 11:13:17 2015
Last change: Thu Jan  8 11:13:08 2015
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.12-a14efad
2 Nodes configured
3 Resources configured

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

Full list of resources:
myapc (stonith:fence_apc_snmp):      Started z1.example.com
Resource Group: nfsgroup
    my_lvm      (ocf::heartbeat:LVM):  Started z1.example.com

PCSD Status:
z1.example.com: Online
z2.example.com: Online

Daemon Status:
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

2. 為叢集配置一項 **Filesystem** 資源。

**注意**

您可透過 **options=options** 參數為 **Filesystem** 資源指定掛載選項作為資源配置的一部分。欲取得完整配置選項請執行 **pcs resource describe Filesystem** 指令。

下列指令將配置一個名為 **nfsshare** 的 **ext4 Filesystem** 資源，作為 **nfsgroup** 資源群組的一部分。此檔案系統使用了您在〈[節 3.2, “為一個 LVM 卷冊配置 ext4 檔案系統”](#)〉中建立的 LVM 卷冊群組和 **ext4** 檔案系統，並且將會被掛載至您在〈[節 3.3, “NFS Share 設定”](#)〉中所建立的 **/nfsshare** 目錄上。

```
[root@z1 ~]# pcs resource create nfsshare Filesystem \
device=/dev/my_vg/my_lv directory=/nfsshare \
fstype=ext4 --group nfsgroup
```

驗證 **my\_lvm** 和 **nfsshare** 資源是否有在執行。

```
[root@z1 ~]# pcs status
...
Full list of resources:
myapc (stonith:fence_apc_snmp):      Started z1.example.com
Resource Group: nfsgroup
    my_lvm (ocf::heartbeat:LVM):      Started z1.example.com
    nfsshare (ocf::heartbeat:Filesystem): Started z1.example.com
...
```

3. 建立名為 **nfs-daemon** 的 **nfserver** 資源，作為 **nfsgroup** 資源群組的一部分。

```
[root@z1 ~]# pcs resource create nfs-daemon nfserver \
nfs_shared_infodir=/nfsshare/nfsinfo nfs_no_notify=true \
--group nfsgroup
[root@z1 ~]# pcs status
...
```

4. 新增 **exportfs** 資源以匯出 **/nfsshare/exports** 目錄。這些資源為 **nfsgroup** 資源群組的一部分。這會為 NFSv4 客戶端建立一個虛擬目錄。NFSv3 客戶端亦可存取這些匯出目錄。

```
[root@z1 ~]# pcs resource create nfs-root exportfs \
clientspec=192.168.122.0/255.255.255.0 \
options=rw,sync,no_root_squash \
directory=/nfsshare/exports \
fsid=0 --group nfsgroup

[root@z1 ~]# # pcs resource create nfs-export1 exportfs \
clientspec=192.168.122.0/255.255.255.0 \
options=rw,sync,no_root_squash directory=/nfsshare/exports/export1 \
fsid=1 --group nfsgroup

[root@z1 ~]# # pcs resource create nfs-export2 exportfs \
clientspec=192.168.122.0/255.255.255.0 \
options=rw,sync,no_root_squash directory=/nfsshare/exports/export2 \
fsid=2 --group nfsgroup
```



- 加入 nfs 客戶端將會使用來存取 nfs share 的 IP 位址資源。您指定的浮動 IP 位址需要進行一項反向 DNS 查詢，否則它必須指定於叢集中所有節點上的 `/etc/hosts` 中。此資源為資源群組 `nfsgroup` 的一部分。在此範例建置中，我們將使用 192.168.122.200 作為浮動 IP 位址。

```
[root@z1 ~]# pcs resource create nfs_ip IPAddr2 \
ip=192.168.122.200 cidr_netmask=24 --group nfsgroup
```

- 當整個 NFS 建置初始化後，加入 `nfsnotify` 資源以傳送 NFSv3 重新啟動通知。



### 注意

若要正確處理 NFS 通知，浮動 IP 位址必須有一組與其相聯的主機名稱，並且它在 nfs 伺服器與 nfs 客戶端上皆必須一致。

```
[root@z1 ~]# pcs resource create nfs-notify nfsnotify \
source_host=192.168.122.200
```

在建立了資源和資源限制後，您便可檢查叢集的狀態。請注意，所有資源皆在相同節點上運作。

```
[root@z1 ~]# pcs status
...
Full list of resources:
myapc (stonith:fence_apc_snmp):      Started z1.example.com
Resource Group: nfsgroup
  my_lvm (ocf::heartbeat:LVM):      Started z1.example.com
  nfsshare (ocf::heartbeat:Filesystem): Started z1.example.com
  nfs-daemon (ocf::heartbeat:nfsserver): Started z1.example.com
  nfs-root (ocf::heartbeat:exportfs): Started z1.example.com
  nfs-export1 (ocf::heartbeat:exportfs): Started
z1.example.com
  nfs-export2 (ocf::heartbeat:exportfs): Started
z1.example.com
  nfs_ip (ocf::heartbeat:IPAddr2):   Started z1.example.com
  nfs-notify (ocf::heartbeat:nfsnotify): Started z1.example.com
...
```

## 3.6. 測試資源配置

您可透過下列程序來驗證您的系統配置。您應該能以 NFSv3 或 NFSv4 掛載已匯出的檔案系統。

- 請在一個叢集之外、位於與建置相同網路中的節點上驗證 NFS share 能藉由掛載 NFS share 來被看見。在此範例中，我們將使用 192.168.122.0/24 網路。

```
# showmount -e 192.168.122.200
Export list for 192.168.122.200:
/nfsshare/exports/export1 192.168.122.0/255.255.255.0
/nfsshare/exports         192.168.122.0/255.255.255.0
/nfsshare/exports/export2 192.168.122.0/255.255.255.0
```

- 若要驗證您是否能以 NFSv4 掛載 NFS share，請將 NFS share 掛載至客戶端節點上的一個目錄中。在進行掛載後，請驗證您是否能看見匯出目錄中的內容。請在測試結束後卸載此 share。

```
# mkdir nfsshare
# mount -o "vers=4" 192.168.122.200:export1 nfsshare
# ls nfsshare
clientdatafile1
# umount nfsshare
```

3. 請驗證您是否能以 NFSv3 掛載 NFS share。在掛載後，請驗證您是否能看見 **clientdatafile1** 這個測試檔案。NFSv3 與 NFSv4 不同，它不使用虛擬檔案系統，您必須掛載特定的 export。請在測試完成後卸載 share。

```
# mkdir nfsshare
# mount -o "vers=3" 192.168.122.200:/nfsshare/exports/export2 nfsshare
# ls nfsshare
clientdatafile2
# umount nfsshare
```

4. 若要測試備援，請執行下列步驟。

- a. 請在叢集外的一個節點上掛載 nfs share 並驗證是否能存取我們在 [〈節 3.3, “NFS Share 設定”〉](#) 中所建立的 **clientdatafile1**。

```
# mkdir nfsshare
# mount -o "vers=4" 192.168.122.200:export1 nfsshare
# ls nfsshare
clientdatafile1
```

- b. 從叢集中的某個節點，決定叢集中的哪個節點要執行 **nfsgroup**。在這範例中，**nfsgroup** 是在 **z1.example.com** 上執行的。

```
[root@z1 ~]# pcs status
...
Full list of resources:
myapc (stonith:fence_apc_snmp):          Started z1.example.com
Resource Group: nfsgroup
  my_lvm (ocf::heartbeat:LVM):          Started z1.example.com
  nfsshare (ocf::heartbeat:Filesystem):  Started
z1.example.com
  nfs-daemon (ocf::heartbeat:nfsserver): Started
z1.example.com
  nfs-root (ocf::heartbeat:exportfs):    Started
z1.example.com
  nfs-export1 (ocf::heartbeat:exportfs): Started
z1.example.com
  nfs-export2 (ocf::heartbeat:exportfs): Started
z1.example.com
  nfs_ip (ocf::heartbeat:IPaddr2):      Started
z1.example.com
  nfs-notify (ocf::heartbeat:nfsnotify): Started
z1.example.com
...
```

- c. 從叢集中的某個節點，將執行 **nfsgroup** 的節點改成待命模式。

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



- d. 驗證 **nfsgroup** 是否已成功啓用於其它叢集節點上。

```
[root@z1 ~]# pcs status
...
Full list of resources:
Resource Group: nfsgroup
  my_lvm      (ocf::heartbeat:LVM):    Started z2.example.com
  nfsshare    (ocf::heartbeat:Filesystem):  Started
z2.example.com
  nfs-daemon  (ocf::heartbeat:nfsserver):      Started
z2.example.com
  nfs-root    (ocf::heartbeat:exportfs):      Started
z2.example.com
  nfs-export1 (ocf::heartbeat:exportfs):      Started
z2.example.com
  nfs-export2 (ocf::heartbeat:exportfs):      Started
z2.example.com
  nfs_ip      (ocf::heartbeat:IPaddr2):        Started
z2.example.com
  nfs-notify  (ocf::heartbeat:nfsnotify):      Started
z2.example.com
...
```

- e. 從叢集外部、掛載 NFS 共享目錄的節點，驗證這個外部節點仍然擁有存取 NFS 掛載點的測試檔案之權限。

```
# ls nfsshare
clientdatafile1
```

在備援時，用戶端的服務會被短暫中斷，但應該會在不需要使用者干預的情況下回復。預設上，使用 NFSv4 的用戶端可能需要至多 90 秒才能回復掛載點；這 90 秒是 NFSv4 的檔案租賃寬限期，是由伺服器在啓動時所觀察的寬限期。NFSv3 的用戶端應該可以在幾秒內就回復對這掛載點的存取功能。

- f. 從叢集中的某個節點，移除從待命模式開始執行 **nfsgroup** 的節點。這不會自己移動叢集資源至此節點上。

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

## 附錄 A. 修訂記錄

<b>修訂 1.2-3.1</b> 翻譯、校閱完成。	<b>Tue Feb 9 2016</b>	<b>Chester Cheng</b>
<b>修訂 1.2-3</b> 準備文件，出版 7.2 GA。	<b>Mon Nov 9 2015</b>	<b>Steven Levine</b>
<b>修訂 1.2-2</b> 準備文件，出版 7.2 Beta。	<b>Tue Aug 18 2015</b>	<b>Steven Levine</b>
<b>修訂 1.1-19</b> 7.1 GA 發行版本	<b>Mon Feb 16 2015</b>	<b>Steven Levine</b>
<b>修訂 1.1-10</b> 7.1 Beta 發行版本	<b>Thu Dec 11 2014</b>	<b>Steven Levine</b>
<b>修訂 1.1-9</b> 新增了 nfs 叢集配置程序	<b>Tue Dec 9 2014</b>	<b>Steven Levine</b>
<b>修訂 1.1-6</b> 更新了Load Balancer 叢集的程序。	<b>Mon Dec 8 2014</b>	<b>Steven Levine</b>
<b>修訂 1.1-5</b> 7.1 Beta 發行版本	<b>Thu Dec 04 2014</b>	<b>Steven Levine</b>
<b>修訂 0.1-33</b> 7.0 GA 發行版本	<b>Mon Jun 2 2014</b>	<b>Steven Levine</b>
<b>修訂 0.1-31</b> 修正了 #886235 記載了 volume_list 的使用方法	<b>Wed May 21 2014</b>	<b>Steven Levine</b>
<b>修訂 0.1-29</b> 重建格式，並更新草稿	<b>Tue May 20 2014</b>	<b>Steven Levine</b>
<b>修訂 0.1-20</b> 更新了 Beta 草稿	<b>Wed Apr 9 2014</b>	<b>Steven Levine</b>
<b>修訂 0.1-8</b> Beta 草稿	<b>Fri Dec 6 2013</b>	<b>Steven Levine</b>
<b>修訂 0.0-1</b> Red Hat Enterprise Linux 7 的第一版本	<b>Wed Jan 16 2013</b>	<b>Steven Levine</b>