



「研磨 Red Hat Certified Architect – Cluster 篇」」

【High Availability MySQL on RHCS 5.1】

就筆者所接觸的客戶，通常企業的資料庫伺服器都會採用備援機制。因為資料庫通常是企業內最重要的服務，一旦資料庫伺服器因硬體或其它原因停止服務，會造成莫大的損失。所以資料庫伺服器大多會採用 High Availability 架構，當原來的資料庫伺服器無法提供服務時，備援的伺服器會馬上接管（take over）資料庫服務。此篇文章便是要介紹如何在 RHCS 5.1 上組態 MySQL 資料庫，架構 High Availability MySQL Solution。



1 RHCS 基本理論

在介紹如何建置 MySQL on RHCS 之前，先概述 RHCS 的基本理論。凡事起頭難，如果一開始便陷入深奧的理論，讀者必然興趣缺缺。所以筆者一直在想如何用淺顯易懂的方式讓大家先對 RHCS 的專有名詞及架構有基本的瞭解。

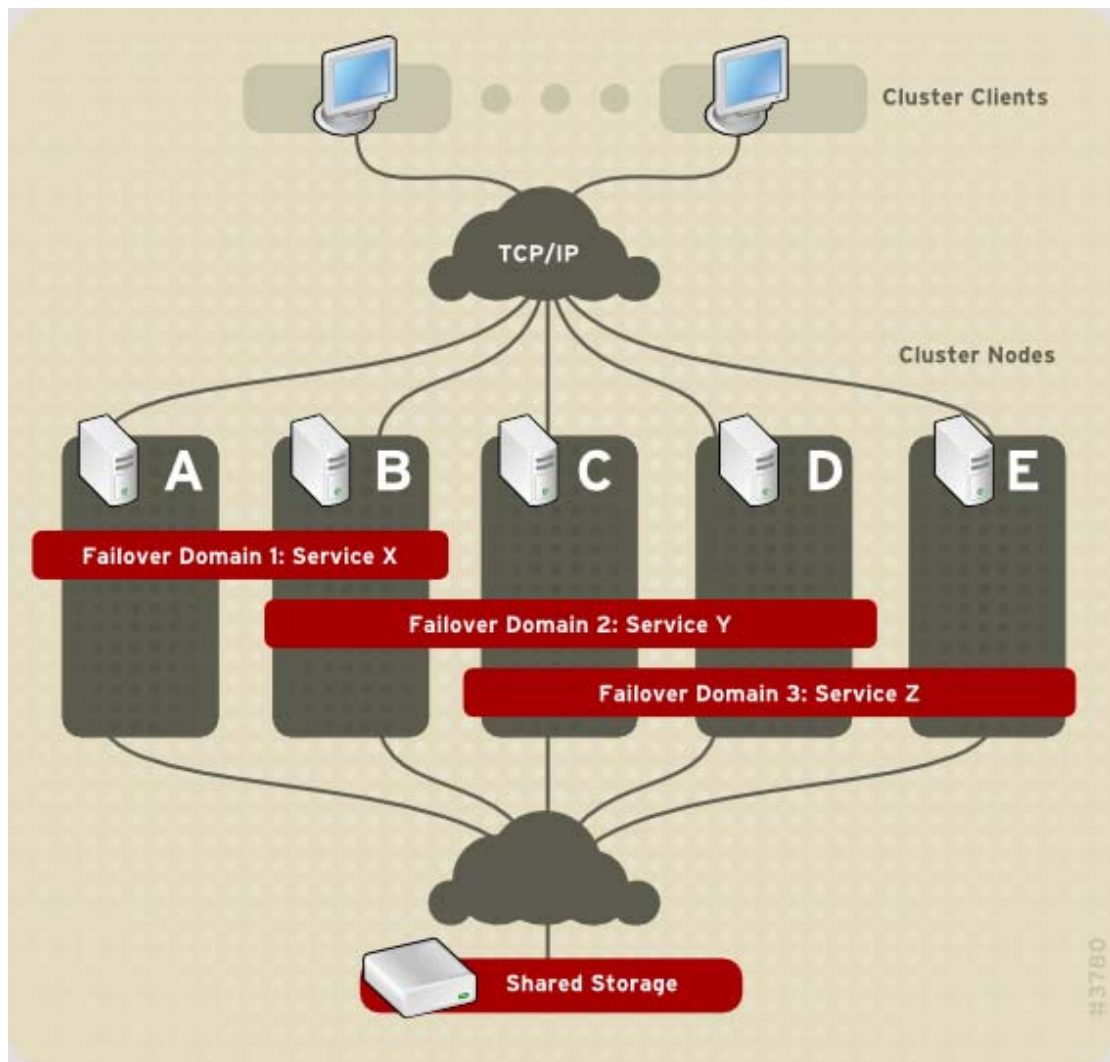


圖 1：RHCS Failover Domain 示意圖

圖片來源：

http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/en-US/RHEL510/Cluster_Suite_Overview/s1-service-management-overview-CSO.html



Red Hat 官方文件中有三張圖（圖 1~圖 3）可說是整個 RHCS 架構的精髓，圖 1 中有幾個重要名詞：

- Cluster Nodes
- Failover Domain
- Service

1.1 Cluster Nodes

Cluster Nodes 就是 Cluster 的成員，就是一台台伺服器。以前筆者也提過：Cluster 是什麼？一般的定義，「所謂 Cluster 就是由一台以上的機器，為了某種特定需求所組成的架構」，根據不同的需求，可將 Cluster 分為以下三種，並對應 Red Hat 由何種軟體提供相關功能。

- High availability cluster

增加伺服器和以網路為基礎的應用程式的高可用性及備援性；由 Cluster Suite 中的 Cluster Manager 技術提供。

- Load Balancing cluster

將服務需求分派給多台伺服器，可視系統負載隨時彈性增加伺服器；由 Cluster Suite 中的 Linux Virtual Server (Piranha) 技術提供。

- High performance cluster (HPC) — 提供同步運算及平行處理的能力；Cluster Suite 不提供（另外有 lam、pvm 套件，規劃由 WS 擔綱）。IBM Cluster 1350 是業界非常有名的 HPC 解決方案。

此篇文章所介紹的 Cluster 是屬於「High availability cluster」，這裏有一點需要跟讀者特別強調的，一般的 High availability cluster，通常用兩台機器組成 High availability cluster（例如 IBM HACMP 解決方案），主要伺服器停止服務時，備援伺服器便會馬上接手主要伺服器的 service。但 Red Hat 所提供「High availability cluster」解決方案，最好用三台以上的伺服器組成 High availability cluster。如果只用兩台伺服器組成 High availability cluster，則拔除傳遞 heartbeat 的網路線時，會發生 split-brain（裂腦）的問題，Red Hat 可利用 quorum disk 的技術來解決這



個問題，不過這個部份理論較複雜，此篇文章先不討論此種作法，後面實作的部份，筆者會用 3 個 nodes 來組成 Cluster，其目的是為了避免 split-brain（裂腦）問題。

1.2 Failover Domain

「Failover Domain」這個名詞比較難理解，先看看 Red Hat 官方文件的定義：

A failover domain is a **subset** of cluster nodes that are eligible to run a particular cluster service

簡單來說，「Failover Domain」就是 Cluster 的子集（subset），就是 Cluster 的部份。由圖 1 可以得知：

- Cluster Nodes 為 nodeA、nodeB、nodeC、nodeD 及 nodeE。
- nodeA 和 nodeB 組成 Failover Domain 1
- nodeB、nodeC 及 nodeD 組成 Failover Domain 2
- nodeC、nodeD 及 nodeE 組成 Failover Domain 3

這樣照字面解釋完後，我想讀者一定還有疑問，那為什麼要分「Failover Domain」？還記得筆者在「Single Point Of Failure」文章內所提過：

「"failover"機制指的是如果一台伺服器停機或故障，另一台伺服器可以接手 (takeover) 故障伺服器所提供的服務。」

OK！我們再回頭看圖 1，Failover Domain1 後緊接著 Service X、Failover Domain2 後緊接著 Service Y、Failover Domain1 後緊接著 Service Z。

所以 Failover Domain 的目的是指定 Service X 只能在 Failover Domain1 內移轉，也就是 Service X 可在 nodeA 和 nodeB 或上執行。同理 Service Y 在 Failover Domain2 內移轉，也就是 Service Y 可在 nodeB、nodeC 或 nodeD 或上執行。



再提醒大家一件事，這是「High availability cluster」，最主要是提供備援機制，Service 同時間只能在某一個 node 上啟動，以 Failover Domain1 而言，ServiceX 只能在 nodeA 或 nodeB 上執行，假設一開始是在 nodeA 上執行，當 nodeA 掛掉時，ServiceX 就會自動移轉至 nodeB 上。

1.3 Service

那什麼是 Service？Service 很容易讓大家誤以為就是 `/etc/init.d/httpd` 這類應用程式，在 RHCS 中的 Service 其實是一堆 Resource 的組合。各位想想，如果主要的 WWW 伺服器掛掉，另一台備援的伺服器要接管 WWW 服務，只需啟動 `/etc/init.d/httpd` 嗎？不對吧！應該還要把原來 Client 所連接的 IP（常稱 service IP）移轉到備援伺服器，還有要把原來的網頁檔案系統（應該要在 share disk）掛載起來，最後才是執行 `/etc/init.d/httpd start`。經過以上講解，再看圖二，就會豁然開朗，圖中 content service 是由 IP resource（service ip）、Application resource（`/etc/init.d/httpd`）及 File system resource（網頁檔案系統）所組成。

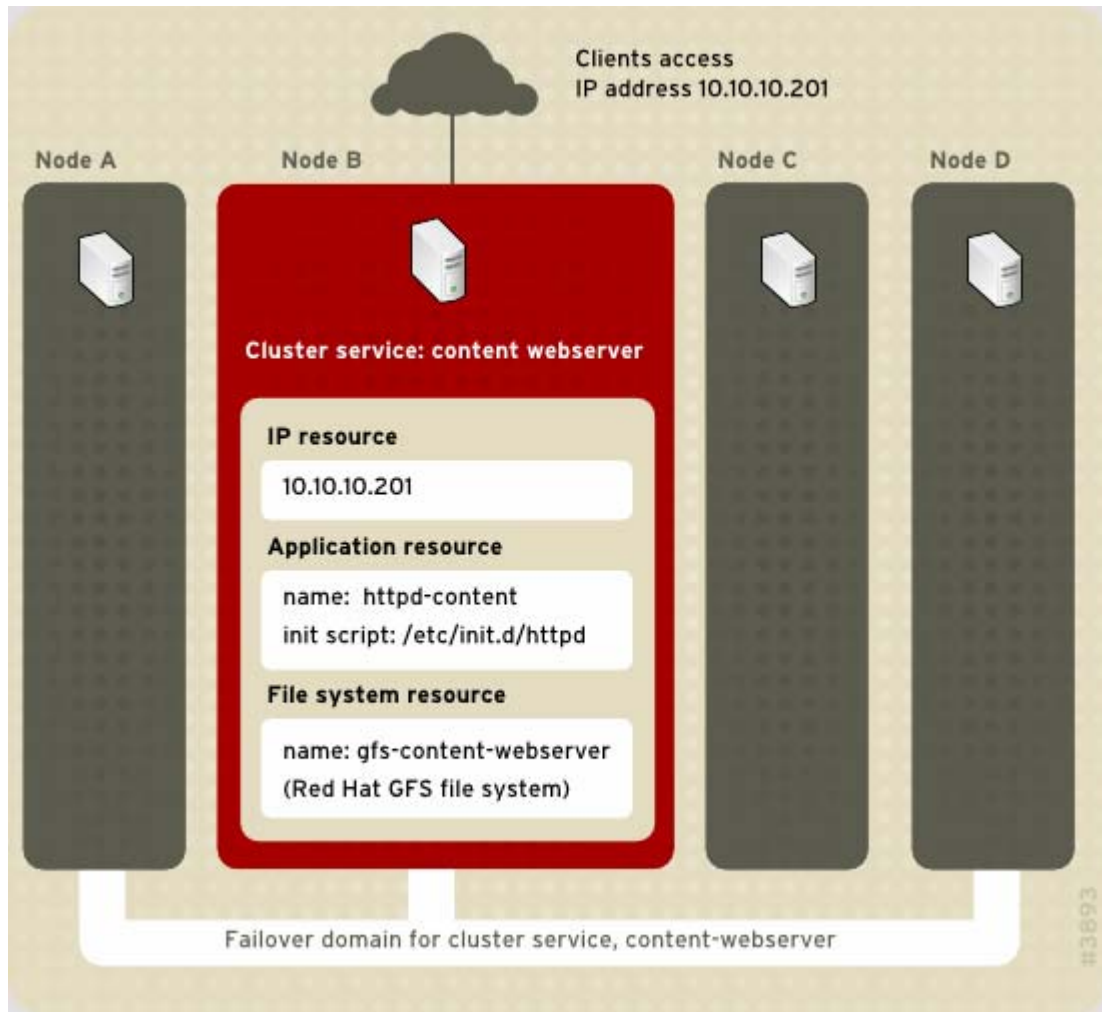


圖 2：Web Server Cluster Service Example

圖片來源：

http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/en-US/RHEL510/Cluster_Suite_Overview/s1-service-management-overview-CSO.html

1.4 Fencing Device

圖 3 中，有個奇怪的裝置「Fencing Device」，「Fencing Device」可能是大家最陌生的，因為其它 Unix 的 HA Solution 並不需要此裝置。先不要管 Fence 中文意義是什麼「柵欄、籬笆」，等一下筆者再解釋為什麼叫 Fencing Device。RHCS 中提到的 Fencing Device，是一種可以直接對伺服器做電源 Power ON/Power OFF 的裝置，注意筆者說的是「電源」，不是去執行作業系統的開關機指令。RHCS 支援的 Fencing Device 很多，不過有些並不常見，例如有種電源延長線，你可以



用 telnet 指令，對某一個插座做 Power ON/Power OFF 的動作，那麼接在這個位置上的伺服器不管作業系統是否完全無反應（有時伺服器 loading 過重，會連 ssh 也無法登入）都會重開機，這種電源延長線有時我們稱作 NPS（Network Power Switch）。不過 NPS，在台灣使用的人很少，就筆者所接觸的案子，客戶通常是採購本身有支援 IPMI 的伺服器，直接用伺服器 IPMI 的功能，或是額外購買 IBM RSAII 卡或是 HP iLO 卡。

說了那麼多，還是沒解釋為什麼要「Fencing Device」，別急，也看一下原廠文件的說明。

「Fencing is the **disconnection of a node from the cluster's shared storage**. Fencing **cuts off I/O from shared storage**, thus ensuring **data integrity**. 」

為什麼 RHCS 需要「Fencing Device」，最主要是避免一種情況發生，什麼情況呢？實務上，尤其是資料庫伺服器可能會遇到這種情況，就是伺服器 loading 過重，完全無反應，連 heartbeat 也無法傳遞，但此時伺服器可能不是真的掛掉，筆者開玩笑的說法是「假死」，經過一段時間，loading 沒那麼重時，又會活過來，heartbeat 又可以傳遞。但是 RHCS 的機制，只要發現主要伺服器的 heartbeat 不通，讀者可以把 heartbeat 想成備援伺服器每隔時間就會 ping 主要伺服器（實際上 RHCS 並不是用 ping 的指令），就像是去聆聽心跳，如果沒有回應（Missed too many heartbeats），就判斷主要伺服器死了，那麼備援伺服器便會著手接管 Service。但是萬一主要伺服器是假死，但 Service 中有包含 File system resource，那麼可能會造成兩台伺服器同時掛載 File system，就會造成檔案系統資料不一致，嚴重的話可能會毀損整個檔案系統。

RHCS 為了避免這種情況，想出了一個辦法，就是「Fencing Device」，當 RHCS 發現主要伺服器的 heartbeat 不通（Missed too many heartbeats），第一件事不是著手接管 Service，而是利用「Fencing Device」將主要伺服器重開機 Power ON/Power OFF，筆者常開玩笑說，這樣一來，假死就變成真死了，也不可能發生主要伺服器還掛載 File system resource 的情況。

現在，讀者應該可以體會為什麼要稱為「Fencing Device」，Fencing 有隔離/圍起來的意思，就是當伺服器假死或是無法 umount File system 時，就把它隔離起來



(power reboot) 讓它無法存取 File system resource (share storage) 。

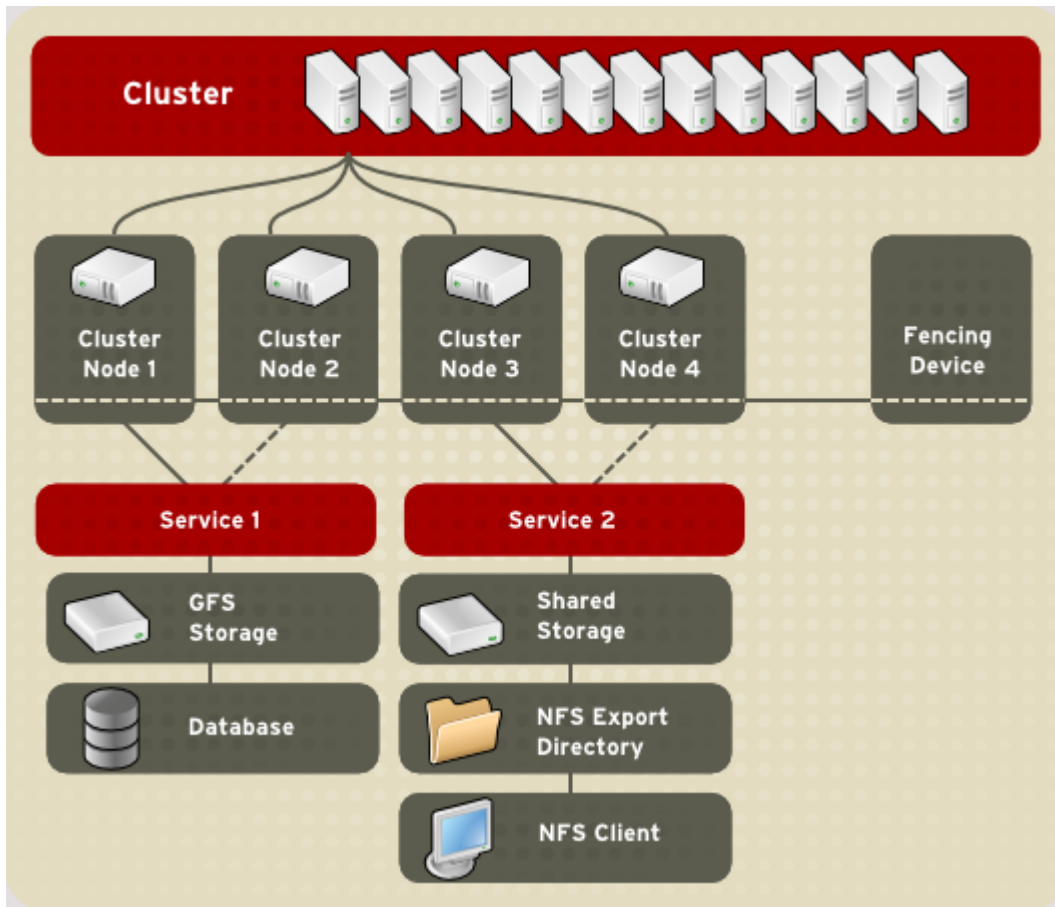


圖 3 : cluster config flowchart

圖片來源：

http://www.redhat.com/docs/manuals/enterprise/RHEL-5-manual/en-US/RHEL510/Cluster_Administration/s2-config-cluster-CA.html



2 iSCSI 規劃及建置

基於大家可能很難接觸 Fiber Storage，筆者這篇文章利用 iSCSI target server 提供 share storage，並在 VMware 的環境下測試，所以讀者不需昂貴的設備就可練習 High Availability MySQL on RHCS 5.1 的建置。

表 1：iSCSI 環境規劃表

功能	作業系統	IP Address	Share Disk
iSCSI target server (server1.example.com)	RHEL 5.1	192.168.0.254	/dev/sdb1 (1GB) /dev/sdb2 (2GB)
iSCSI initiator (node1.example.com)	RHEL 5.1	192.168.0.101	/dev/sdb (1GB) /dev/sdc (2GB)
iSCSI initiator (node2.example.com)	RHEL 5.1	192.168.0.102	/dev/sdb (1GB) /dev/sdc (2GB)
iSCSI initiator (node3.example.com)	RHEL 5.1	192.168.0.103	/dev/sdb (1GB) /dev/sdc (2GB)

讀者可參考上期【RHEL 5 上的 iSCSI 環境】，並根據表一建置所需的 Share Storage，設定完成後，不管在那一個 node 應該都可以 Share Disk /dev/sdb 及 /dev/sdc，並用 fdisk 切割/dev/sdb，然後再格式化，此塊空間未來是要用來存放 MySQL 的資料庫。

```
[root@node1 ~]# fdisk -l /dev/sdb
```

```
Disk /dev/sdb: 1028 MB, 1028127744 bytes
```

```
32 heads, 62 sectors/track, 1012 cylinders
```

```
Units = cylinders of 1984 * 512 = 1015808 bytes
```



Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	1012	1003873	83	Linux

```
[root@node1 ~]# mke2fs -j /dev/sdb1
```

```
mke2fs 1.39 (29-May-2006)
```

```
Filesystem label=
```

```
OS type: Linux
```

```
Block size=4096 (log=2)
```

```
Fragment size=4096 (log=2)
```

```
125696 inodes, 250968 blocks
```

```
12548 blocks (5.00%) reserved for the super user
```

```
First data block=0
```

```
Maximum filesystem blocks=260046848
```

```
8 block groups
```

```
32768 blocks per group, 32768 fragments per group
```

```
15712 inodes per group
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376
```

```
Writing inode tables: done
```

```
Creating journal (4096 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 31 mounts or  
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

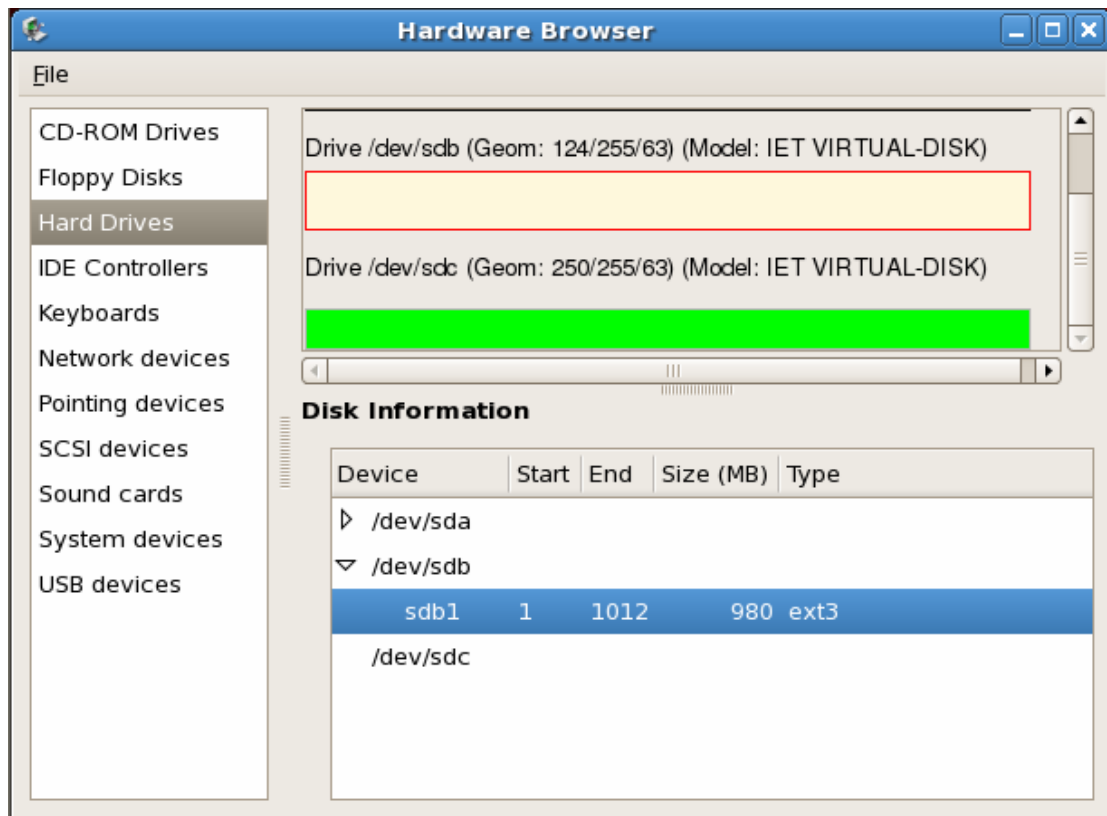


圖 4：檢查 node1 上的 iSCSI Share Storage



3 RHCS 規劃及建置

接下來，筆者會根據表 2 來建置 High Availability MySQL，其中有關 Fencing Device 的設定，實務上，購買不同的 Fencing Device，設定方法會有所不同。筆者是在 VMware 環境上建置 High Availability MySQL，所以 fencing device 選擇 Manual Fencing。

表 2：RHCS 架構規劃表

Cluster Name : eu_cluster			
Cluster Nodes	node1.example.com	node2.example.com	node3.example.com
Heartbeat/service LAN (eth0)	192.168.0.101	192.168.0.102	192.168.0.102
Fencing Device	Manual	Manual	Manual
Failover Domain1	db_domain	db_domain	
Failover Domain2		www_domain	www_domain
Service X : DB Failover Domain : db_domain			
IP Resource	Service IP : 192.168.0.100		
File system Resource	Name : mysql_db mount point : /var/lib/mysql Device File : /dev/sdb1 (iSCSI Share Storage)		
Application Resource	Name : mysqld Script : /etc/init.d/mysqld		

3.1 安裝 RHCS 5.1

RHEL 5.1 已內含 Red Hat Cluster 及 GFS 軟體，如果安裝是有輸入產品序號並選擇安裝全部套件，則會自動會安裝 RedHat Cluster 及 GFS 軟體。若是安裝是沒有輸入產品序號，則不會安裝 Red Hat Cluster 及 GFS 軟體。則必須架好 YUM Server



後，再利用「system-config-packages」進行安裝，勾選「Clustering」及「Cluster Storage」套件組。

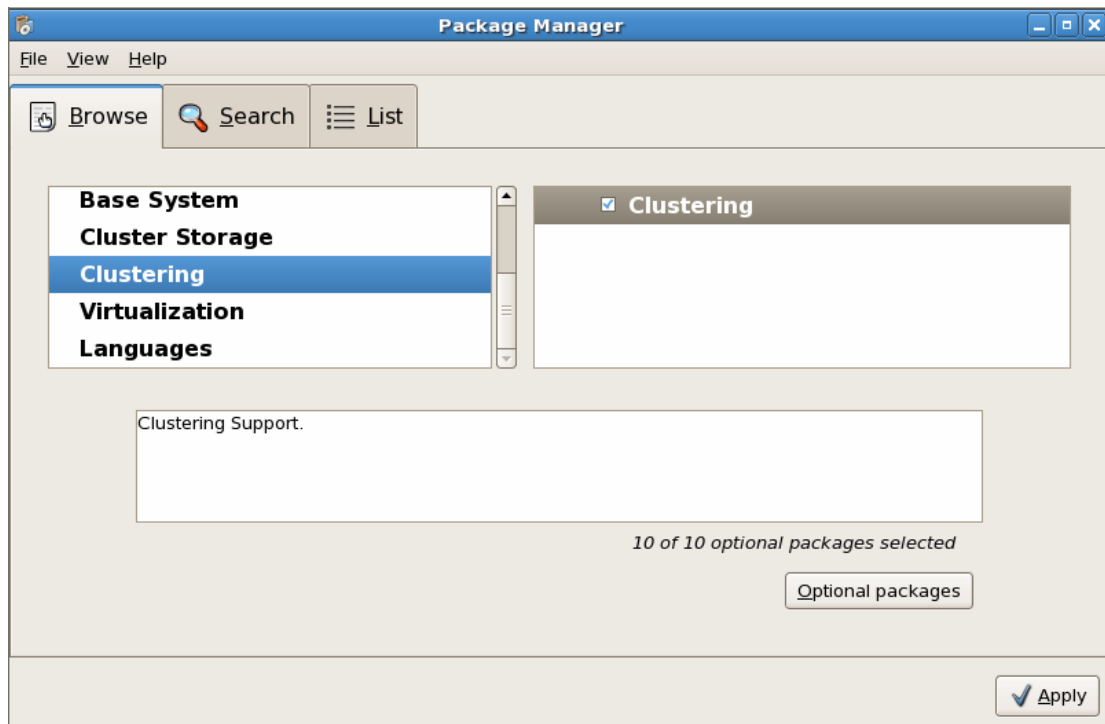


圖 5：system-config-packages 畫面

3.2 建立 Cluster

這期文章筆者先利用「system-config-cluster」工具來建置 Cluster，大致流程如下：先在 node1 設定 Cluster，設定完成後，再將 cluster.conf 複製到其它的 nodes。

啟動「system-config-cluster」後，會出現如圖 6 的畫面，選擇「Create New Configuration」只需填入 cluster 的名稱，其餘採用預設值即可。接著按下右下方「Add a Cluster Node」，加入 node1.example.com～node3.example.com。「Quorum Votes:」不用填。

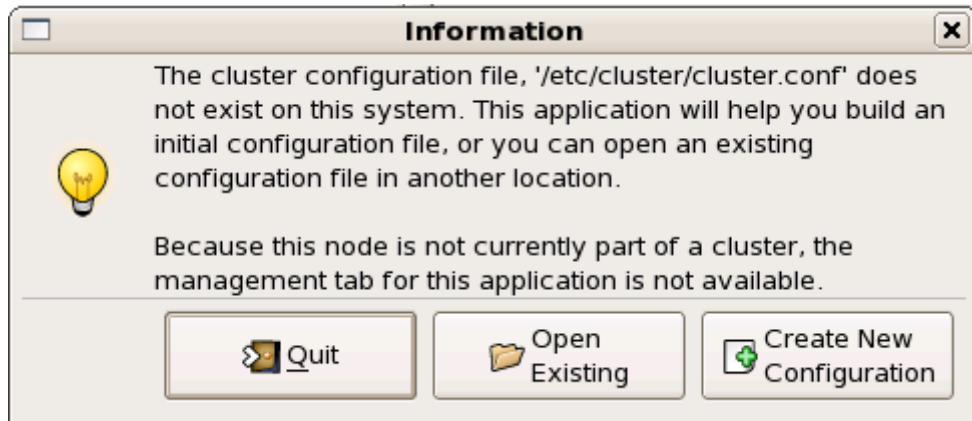


圖 6 : Create New Configuration

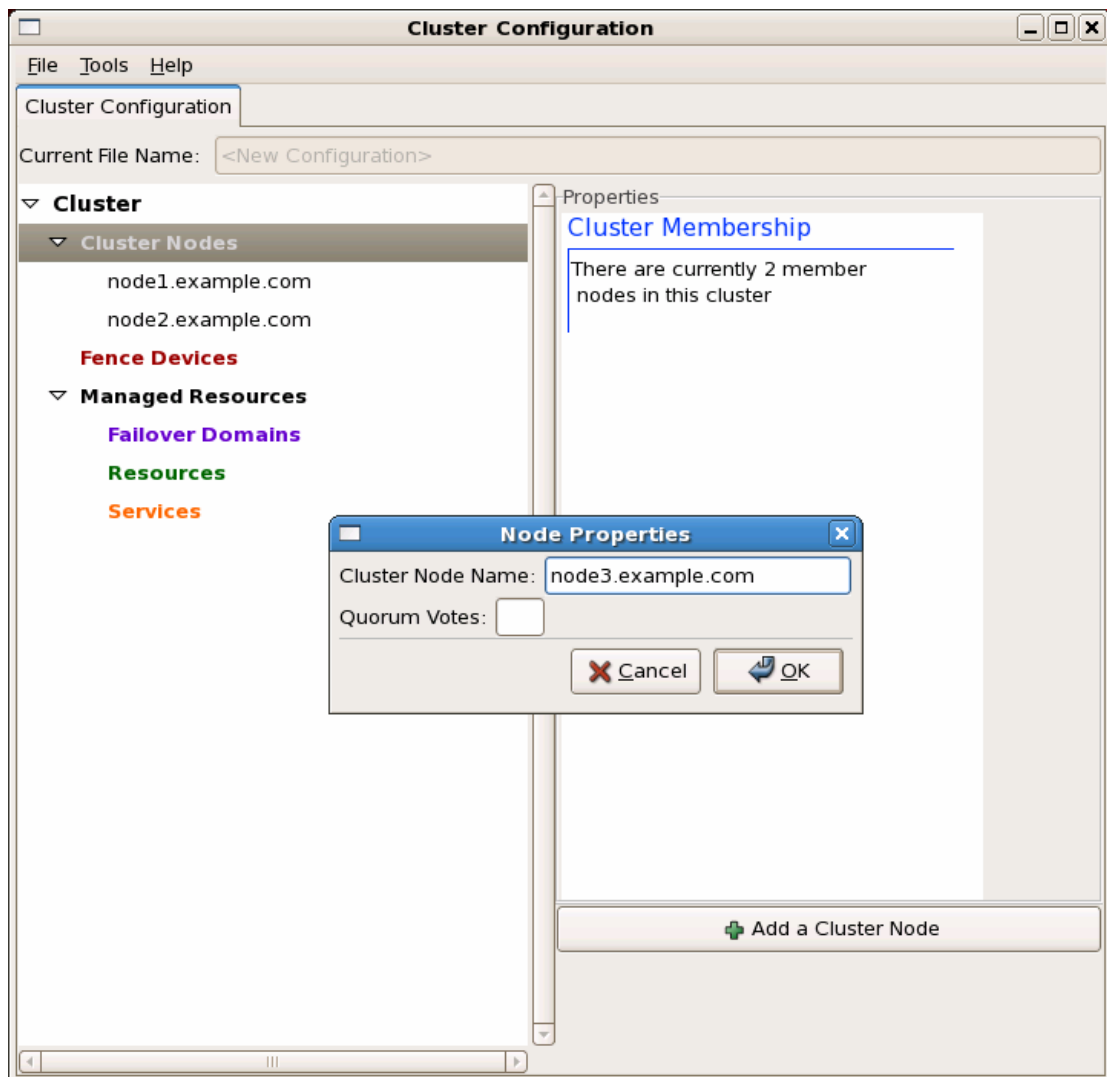


圖 7 : 新增 node1.example.com 至 node3.example.com



3.3 設定 Fence Devices

這個部份，讀者要特別留意，實務上得根據所採購的裝置，例如 IBM 的 RSAII 卡，必須先設置 RSAII 的 IP 及登入的資訊。因為筆者是利用 VMWare 來建置，所以選擇「Manual Fencing」，參考圖 8。

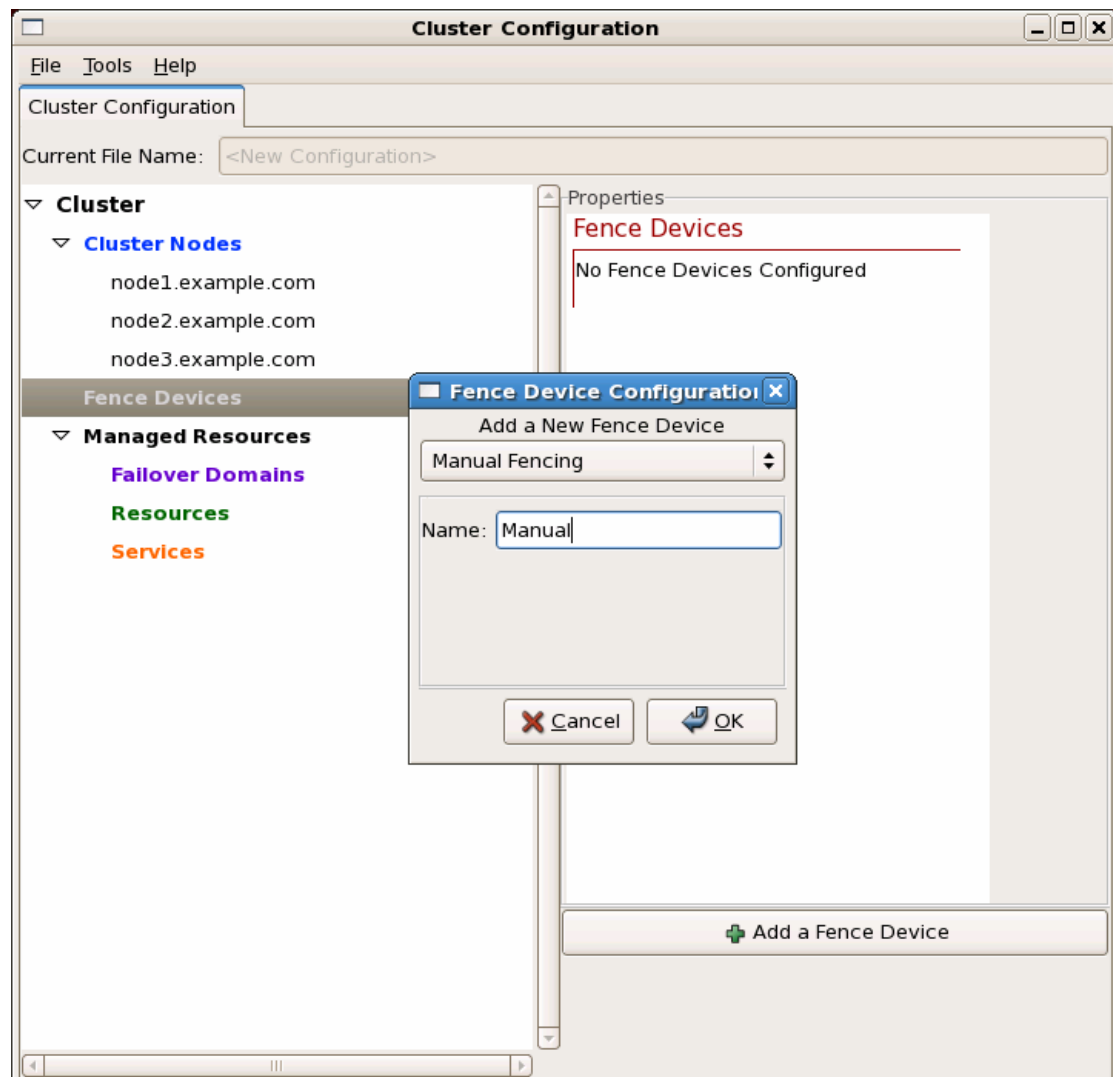


圖 8：Add a Fence Device

新增 Fence Device 後，得回頭指定用這個新增的 Fence Device 來 fencing 這些 nodes。步驟如下：先點選「node1.example.com」，然後按右下方的「Add a Fence Device」，便會出現 Fence Configuration 的設定視窗（圖 9）。

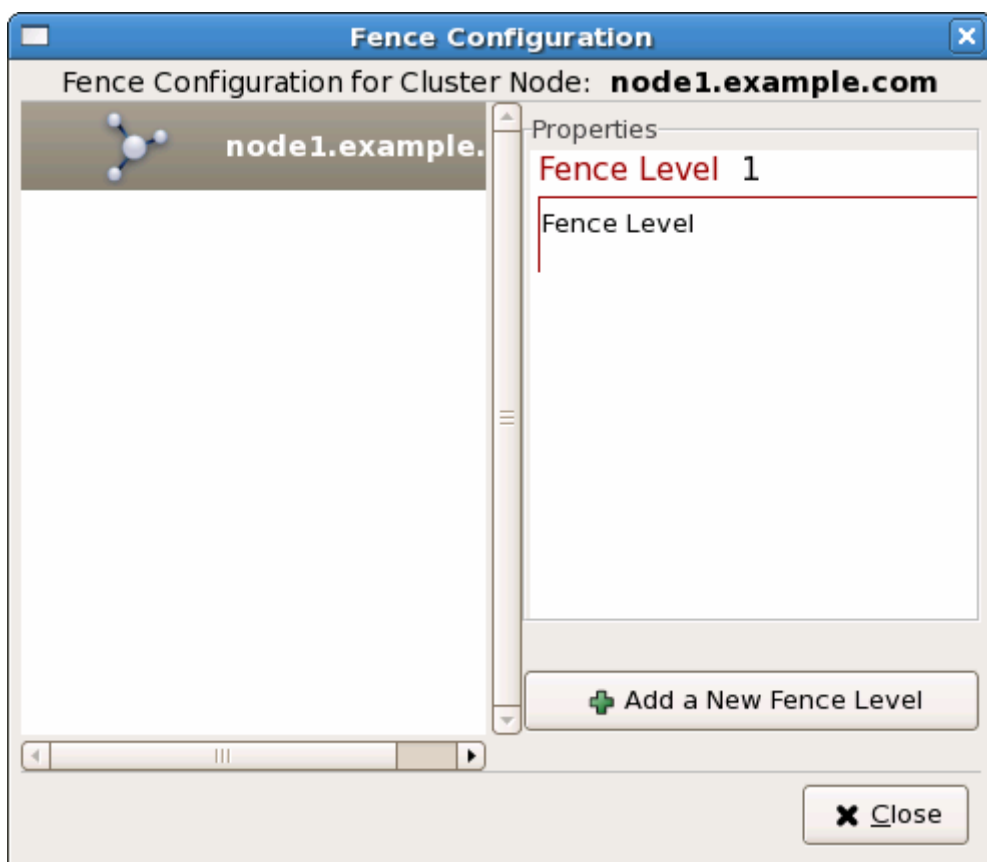


圖 9：Fence Configuration 的設定視窗

出現「Fence Configuration」的設定視窗後，按下「Add a New Fence Level」按鈕後，再點選所新增出來的「Fence-Level-1」，接著按右下方的「Add a New Fence to this Level」，選擇剛剛新增出來的「Manual」Fence Device。

node1.example.com 設定完畢後，node2.example.com 及 node3.example.com 也依此步驟設定。

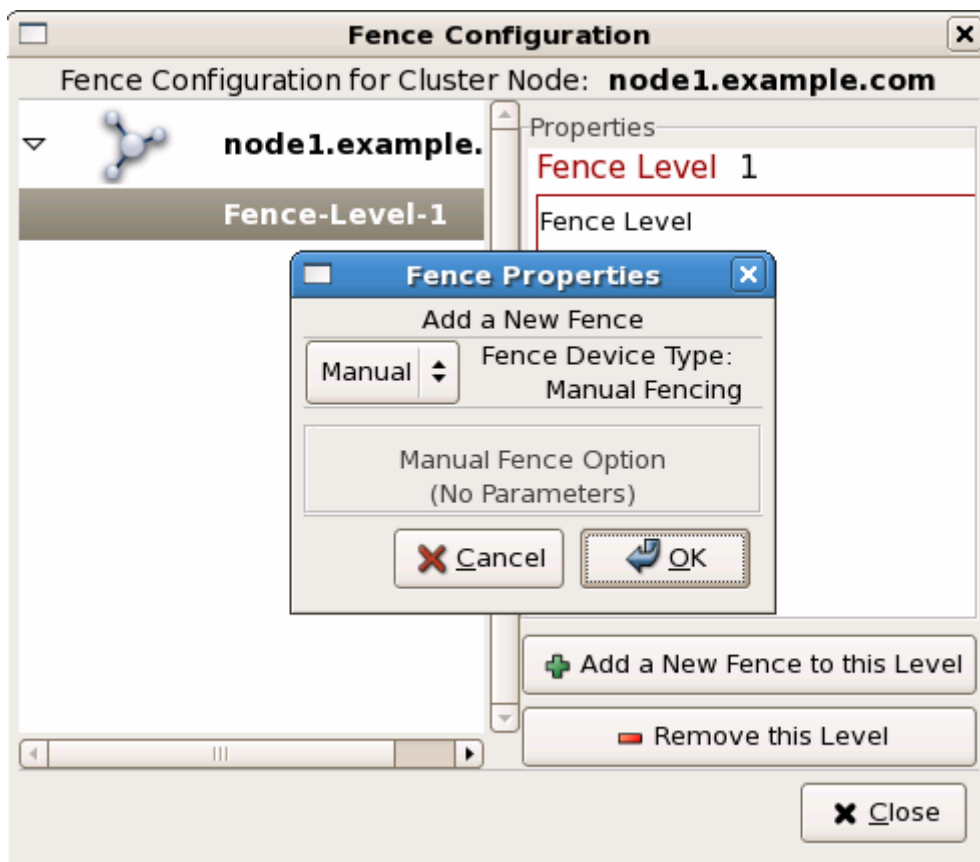


圖 10：Fence Properties 視窗

3.4 設定 Failover Domain

如圖 11 點選「Failover Domains」，按下「Create a Failover Domain」，指定 Failover Domain 名稱，例如 db_domain，利用「Available Cluster Nodes」下拉選單加入成員 node1.example.com 及 node2.example.com，記得勾選「Restrict Failover to This Domains Members」；同樣的，如果要新增 www_domain，步驟也是一樣。

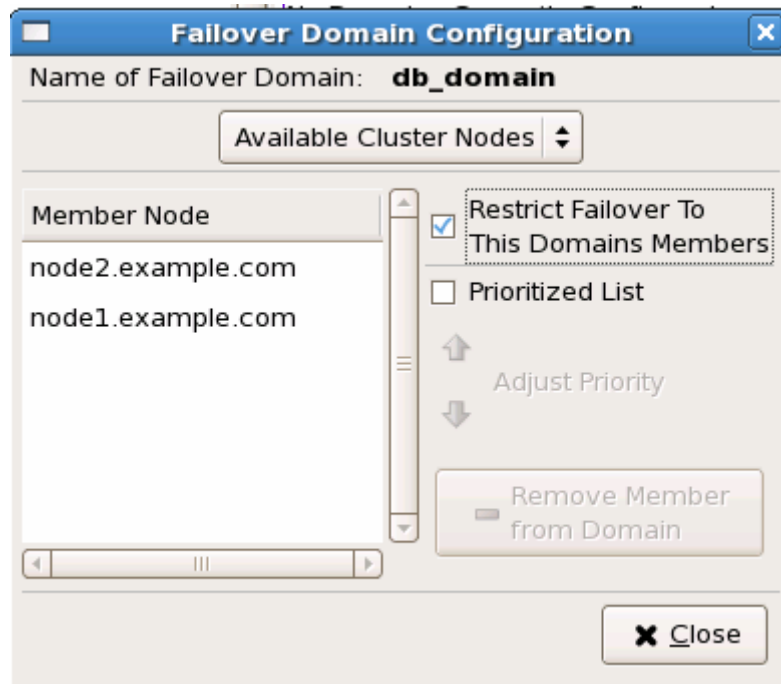


圖 11 : Failover Domain Configuration

3.5 新增 Cluster Resources

根據表 2 新增 3 個 Resources : IP Resource(圖 12)、File system resource(圖 13)、Application Resource(圖 14)。新增 resource 的部份比較簡單，不過設定 File system resource 要特別留意，記得勾選「Force umount」及「Reboot host node if umount fails」。



Resource Configuration

Select a Resource Type:

IP Address

IP Address Resource Configuration

192 . 168 . 0 . 100

Monitor Link

Cancel OK

圖 12 : IP Resource 畫面

Resource Configuration

Select a Resource Type:

File System

File System Resource Configuration

Name: mysql_db

File System Type: ext3

Mount Point: /var/lib/mysql

Device: /dev/sdb1

Options:

File System ID:

Force unmount

Reboot host node if unmount fails

Check file system before mounting

Cancel OK

圖 13 : File System Resource

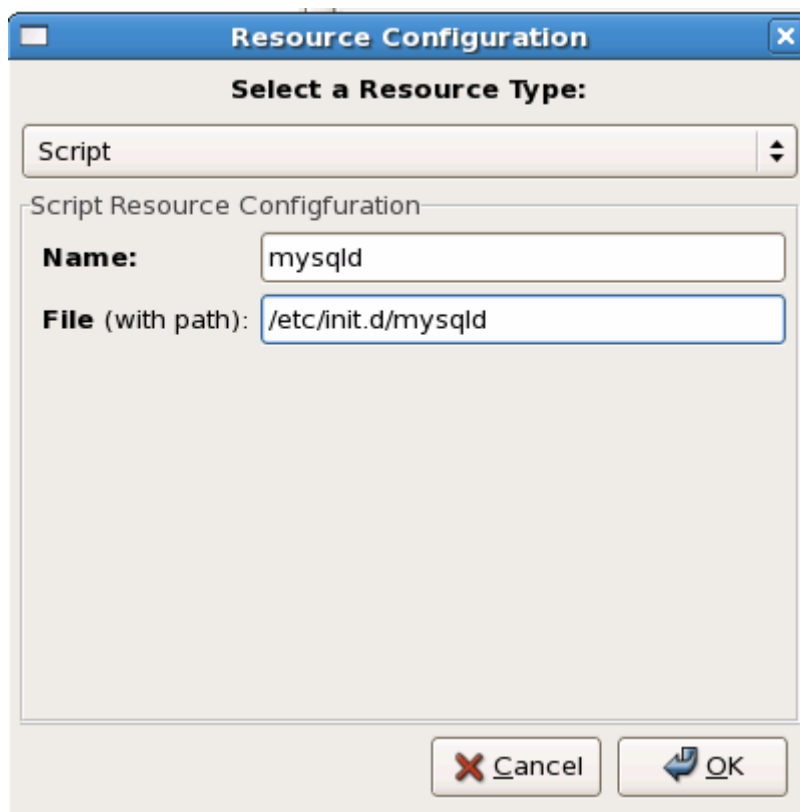


圖 14：Application Resource

3.6 新增 Cluster Service

前面有提過在 RHCS 中的 Service 其實是一堆 Resource 的組合，所以只要原理搞懂，設定就沒什麼問題。筆者用圖 15 解釋相關設定，首先指定 service 的名稱；接著看到「Autostart This Service」這個選項，「Autostart This Service」的用途是當 RHCS 啟動時會自動將該服務啟動。實務上，通常不勾選這個項目，待 RHCS 服務全部啟動成功後，再手動指定該 Service 要啟動在那一個 node。

下拉 Failover Domain 選單，指定該服務只能在 db_domain Failover Domain 內移轉。Recovery Policy 指定「relocate」，「relocate」的作用是當 DB service 有問題時不會嘗試 restart 該服務，會直接將 DB service 移轉至另一台 node。

最後利用「Add a Shared Resource to this service」將這 3 個 resource 加到這個 service。

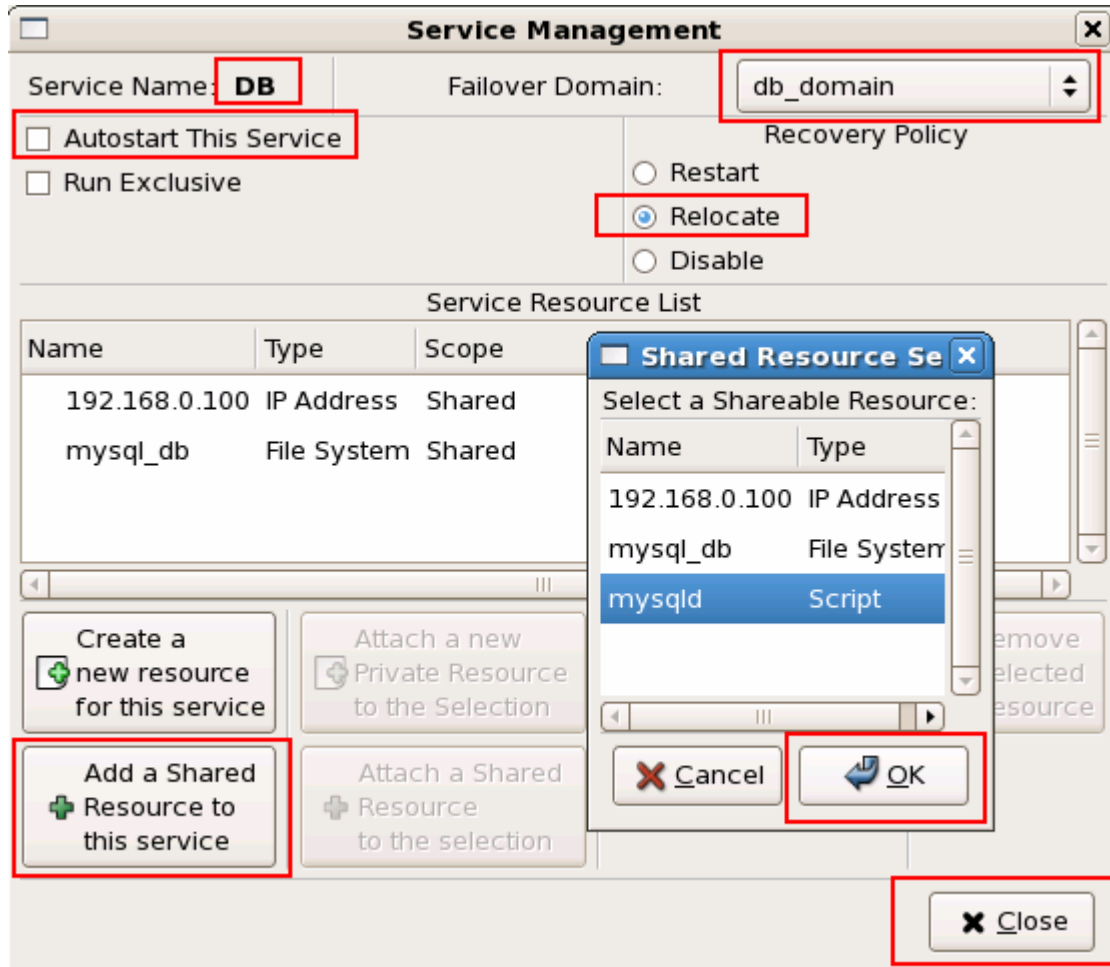


圖 15：Service Management 設定畫面

3.7 部署 cluster.conf

設定完成後，點選左上方的「File」→「Save」將 cluster 的設定存檔，然後再利用 scp 指令將/etc/cluster/cluster.conf 複製到其它兩個 node。

筆者的 cluster.conf 內容如下：

```
<?xml version="1.0" ?>
<cluster config_version="2" name="eu_cluster">
  <fence_daemon post_fail_delay="0" post_join_delay="3"/>
  <clusternodes>
```



```
<clusternode name="node1.example.com" nodeid="1"
votes="1">
    <fence>
        <method name="1">
            <device name="Manual"
nodename="node1.example.com"/>
        </method>
    </fence>
</clusternode>
<clusternode name="node2.example.com" nodeid="2"
votes="1">
    <fence>
        <method name="1">
            <device name="Manual"
nodename="node2.example.com"/>
        </method>
    </fence>
</clusternode>
<clusternode name="node3.example.com" nodeid="3"
votes="1">
    <fence>
        <method name="1">
            <device name="Manual"
nodename="node3.example.com"/>
        </method>
    </fence>
</clusternode>
</clusternodes>
</cman/>
```



```
<fencedevices>
    <fencedevice agent="fence_manual" name="Manual"/>
</fencedevices>

<rm>

    <failoverdomains>
        <failoverdomain name="db_domain" ordered="0"
restricted="1">
            <failoverdomainnode
name="node1.example.com" priority="1"/>
            <failoverdomainnode
name="node2.example.com" priority="1"/>
        </failoverdomain>
        <failoverdomain name="www_domain" restricted="1">
            <failoverdomainnode
name="node2.example.com" priority="1"/>
            <failoverdomainnode
name="node3.example.com" priority="1"/>
        </failoverdomain>
    </failoverdomains>

    <resources>
        <ip address="192.168.0.100" monitor_link="1"/>
        <fs device="/dev/sdb1" force_fsck="0"
force_unmount="1" fsid="50698" fstype="ext3" mountpoint="/var/l
ib/mysql" name="mysql_db" options="" self_fence="1"/>
        <script file="/etc/init.d/mysqld" name="mysqld"/>
    </resources>

    <service autostart="0" domain="db_domain" name="DB"
recovery="relocate">

        <ip ref="192.168.0.100"/>
```



```
<fs ref="mysql_db"/>
<script ref="mysqld"/>
</service>
</rm>
</cluster>
```

3.8 啟動 RHCS 服務

將 cluster.conf 複製到 node2.example.com 及 node3.example.com 後，請按照下列順序啟動 RHCS 服務。

1. 同時在 node1~node3 啟動 cman，指令為：service cman start
2. 同時在 node1~node3 啟動 rgmanager，指令為：service rgmanager start
3. 在 node1 上執行 clustat 可查看整個 cluster 狀態（圖 16）。

```
root@node1:~ [80x17]
[root@node1 ~]# clustat
Member Status: Quorate

Member Name          ID   Status
-----
node1.example.com    1   Online, Local, rgmanager
node2.example.com    2   Online, rgmanager
node3.example.com    3   Online, rgmanager

Service Name      Owner (Last)      State
-----
service:DB        (none)            disabled
[root@node1 ~]#
```

圖 16：查看整個 cluster 狀態

4. 利用「clusvcadm -e DB -m node1.example.com」將 DB service 啟動在 node1.example.com。



```
root@node1:~ [80x17]
[root@node1 ~]# clusvcaadm -e DB -m node1.example.com
Member node1.example.com trying to enable service:DB...Success
service:DB is now running on node1.example.com
[root@node1 ~]# clustat
Member Status: Quorate

Member Name          ID   Status
-----
node1.example.com    1   Online, Local, rgmanager
node2.example.com    2   Online, rgmanager
node3.example.com    3   Online, rgmanager

Service Name      Owner (Last)      State
-----
service:DB        node1.example.com started
[root@node1 ~]#
```

圖 17：enable DB service

5. 檢驗 Service IP(ip addr show eth0)、File System Resource(df -h)及 Application Resource (/etc/init.d/mysqld status) 是否真的在 node1 上執行。

```
root@node1:~ [80x16]
[root@node1 ~]# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:bd:f6:84 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.101/24 brd 192.168.0.255 scope global eth0
    inet 192.168.0.100/32 scope global eth0
    inet6 fe80::20c:29ff:febd:f684/64 scope link
        valid_lft forever preferred_lft forever
[root@node1 ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       9.7G  2.4G  6.9G  26% /
/dev/sda1       76M   11M   62M  16% /boot
tmpfs           252M   0    252M   0% /dev/shm
/dev/sdb1       965M   38M  878M   5% /var/lib/mysql
[root@node1 ~]# /etc/init.d/mysqld status
mysqld (pid 17589) is running...
[root@node1 ~]#
```

圖 18：檢驗 node1 上 resource 情況

3.9 手動移轉

DB service 現於 node1.example.com 上執行，若是 node1.example.com 要升級硬體或維修需要停機（預期性停機），可先利用「clusvcaadm -r DB -m node2.example.com」將 DB service 移轉至 node2.example.com，然後再停機 node1.example.com 進行維修。在此架構下，Client 所連接的 IP 為



「192.168.0.100」，受影響的時間非常短，這就是筆者一直強調 RHCS 的主要用途是備援，提可整個系統的可用性。

```
root@node1:~ [80x16]
[root@node1 ~]# clusvcadm -r DB -m node2.example.com
Trying to relocate service:DB to node2.example.com...Success
service:DB is now running on node2.example.com
[root@node1 ~]# clustat
Member Status: Quorate

Member Name          ID      Status
-----
node1.example.com    1      Online, Local, rgmanager
node2.example.com    2      Online, rgmanager
node3.example.com    3      Online, rgmanager

Service Name      Owner (Last)      State
-----
service:DB        node2.example.com  started
[root@node1 ~]#
```

圖 19：手動移轉 DB Service 至 node2.example.com

3.10 Failover 測試

假設 DB service 現在 node1.example.com 上執行，如果 node1.example.com 異常當機（非預期性停機），筆者利用「poweroff」指令測試（圖 20），則 node2.example.com 會自動接管此服務（圖 21）。

```
PieTTY (inactive)
[root@node1 ~]# poweroff

Broadcast message from root (pts/1) (Mon Dec 15 15:34:23 2008):

The system is going down for system halt NOW!
[root@node1 ~]#
```

圖 20：poweroff node1.example.com



```
root@node2:~  
[root@node2 ~]# clustat  
Member Status: Quorate  
  
Member Name          ID  Status  
-----  
node1.example.com    1  Offline  
node2.example.com    2  Online, Local, rgmanager  
node3.example.com    3  Online, rgmanager  
  
Service Name          Owner (Last)          State  
-----  
service:DB            node2.example.com     started  
[root@node2 ~]#
```

圖 21：node2.example.com 接管 DB service

作者簡介

林彥明 (Alex YM Lin)：IBM 高級技術專員，現負責 Power Systems 及 Cluster 1350 業務技術支援工作，近來參與 NCHC IBM Cluster 1350 (亞洲運算能力僅次日本的超級電腦)及中山大學 p595 HPC 超級電腦專案。具有 RHCA (Red Hat 架構師)、RHCDS (Red Hat Certified Datacenter Specialist)、RHCX (Red Hat 認證主考官)、RHCE、NCLP (Novell Linux 認證專家)、LPIC、IBM AIX ... 等國際認證。