

红帽企业版 Linux 6

集群管理

配置和管理高可用性附加组件



红帽企业版 Linux 6 集群管理

配置和管理高可用性附加组件

版 0

Copyright © 2011 Red Hat Inc..

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution—Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at <http://creativecommons.org/licenses/by-sa/3.0/>. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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, MetaMatrix, 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.

All other trademarks are the property of their respective owners.

1801 Varsity Drive
Raleigh, NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701

《配置和管理高可用性附加组件》论述了在红帽企业版 Linux 6 中配置和管理高可用性附加组件。

简介	vii
1. 文档约定	viii
1.1. 排版约定	viii
1.2. 抬升式引用约定	ix
1.3. 备注及警告	x
2. 反馈	x
1. Red Hat 高可用性附加组件配置和管理概述	1
1.1. 红帽企业版 Linux 6.1 中新的和更改的功能	1
1.2. 配置基础	1
1.3. 设置硬件	2
1.4. 安装 Red Hat 高可用性附加组件软件	3
1.5. 配置 Red Hat 高可用性附加组件软件	3
2. 在配置红帽高可用性附加组件前	5
2.1. 常规配置注意事项	5
2.2. 兼容硬件	6
2.3. 启用 IP 端口	6
2.3.1. 在集群节点中启用 IP 端口	6
2.3.2. 在运行 luci 的计算机中启用 IP 端口	7
2.4. 将 ACPI 配置为使用整合的 Fence 设备	7
2.4.1. 使用 chkconfig 管理禁用 ACPI 软关闭	8
2.4.2. 使用 BIOS 禁用 ACPI 软关闭	9
2.4.3. 在 grub.conf 文件中完全禁用 ACPI。	10
2.5. 配置 HA 服务注意事项	11
2.6. 配置验证	14
2.7. NetworkManager 注意事项	17
2.8. 使用仲裁磁盘的注意事项	17
2.9. 红帽高可用性附加组件及 SELinux	18
2.10. 多播地址	18
2.11. ricci 注意事项	19
3. 使用 Conga 配置红帽高可用性附加组件	21
3.1. 配置任务	21
3.2. 启动 luci	22
3.3. 创建集群	23
3.4. 全局集群属性	25
3.4.1. 配置常规属性	25
3.4.2. 配置 Fence 守护进程属性	26
3.4.3. 网络配置	26
3.4.4. 仲裁磁盘配置	27
3.4.5. 日志配置	28
3.5. 配置 Fence 设备	28
3.5.1. 创建 Fence 设备	29
3.5.2. 修改 Fence 设备	29
3.5.3. 删除 Fence 设备	30
3.6. 为集群成员配置 Fencing	30
3.6.1. 为节点配置单一 Fence 设备	30
3.6.2. 配置备份 Fence 设备	31
3.6.3. 配置使用冗余电源的节点	31
3.7. 配置故障切换域	33
3.7.1. 添加故障切换域	34
3.7.2. 修改故障切换域	35
3.7.3. 删除故障切换域	35
3.8. 配置全局集群资源	35

3.9. 在集群中添加集群服务	36
4. 使用 Conga 管理 Red Hat 高可用性附加组件	39
4.1. 在 luci 界面中添加现有集群	39
4.2. 管理集群节点	39
4.2.1. 重启集群节点	39
4.2.2. 使节点离开或者加入集群	40
4.2.3. 在运行的集群中添加成员	40
4.2.4. 删除集群中的成员	41
4.3. 启动、停止、刷新和删除集群	41
4.4. 管理高可用性服务	42
5. 使用 ccs 命令配置红帽高可用性附加组件	45
5.1. 操作概述	45
5.1.1. 在本地系统中创建集群配置文件	46
5.1.2. 查看当前集群配置	46
5.1.3. 使用 ccs 命令指定 ricci 密码	46
5.1.4. 修改集群配置组件	47
5.2. 配置任务	47
5.3. 启动 ricci	47
5.4. 创建集群	48
5.5. 配置 Fence 设备	49
5.6. 为集群成员配置 Fencing	51
5.6.1. 为节点配置使用单一电源的 Fence 设备	51
5.6.2. 为节点配置单一存储 Fence 设备	52
5.6.3. 配置备用 Fence 设备	54
5.6.4. 配置使用冗余电源的节点	57
5.6.5. 删除 Fence 方法和 Fence 事务	60
5.7. 配置故障切换域	60
5.8. 配置全局集群资源	62
5.9. 在集群中添加集群服务	63
5.10. 配置仲裁磁盘	65
5.11. 其它集群配置	66
5.11.1. 集群配置版本	67
5.11.2. 多播配置	67
5.11.3. 配置双节点集群	68
5.12. 在集群节点中传播配置文件	68
6. 使用 ccs 管理 Red Hat 高可用性附加组件	69
6.1. 管理集群节点	69
6.1.1. 使节点离开或者加入集群	69
6.1.2. 在运行的集群中添加成员	69
6.2. 启动和停止集群	69
6.3. 诊断并修正集群中的问题	70
7. 使用命令行工具配置红帽高可用性附加组件	71
7.1. 配置任务	72
7.2. 配置基本集群配置文件	72
7.3. 配置 Fencing	75
7.4. 配置故障切换域	80
7.5. 配置 HA 服务	83
7.5.1. 添加集群资源	83
7.5.2. 在集群中添加集群服务	85
7.6. 确认配置	88
8. 使用命令行工具管理红帽高可用性附加组件	91

8.1. 启动和停止集群软件	91
8.1.1. 启动集群软件	92
8.1.2. 停止集群软件	92
8.2. 删除或者添加节点	93
8.2.1. 从集群中删除节点	93
8.2.2. 在集群中添加节点	96
8.2.3. 三节点和双节点配置示例	98
8.3. 管理高可用性服务	100
8.3.1. 使用 clustat 显示 HA 服务	100
8.3.2. 使用 clusvcadm 管理 HA 服务	101
8.4. 更新配置	103
8.4.1. 使用 cman_tool version -r 更新配置	103
8.4.2. 使用 scp 更新配置	105
9. 诊断并修正集群中的问题	109
9.1. 没有形成集群	109
9.2. 无法在 Fence 或者重启后重新加入集群的节点	109
9.3. 集群服务挂起	110
9.4. 无法启动集群服务	110
9.5. 无法迁移集群控制的服务	110
9.6. 双节点集群的每个节点都报告第二个节点无法工作	111
9.7. 在 LUN 路径失败中 Fence 的节点	111
9.8. 仲裁磁盘不作为集群成员出现	111
9.9. 异常故障切换行为	111
9.10. 随机发生 fencing	111
10. 使用红帽高可用性附加组件进行 SNMP 配置	113
10.1. SNMP 和红帽高可用性附加组件	113
10.2. 使用红帽高可用性附加组件配置 SNMP	113
10.3. 转发 SNMP 陷阱	114
10.4. 红帽高可用性附加组件产生的 SNMP 陷阱	114
A. Fence 设备参数	117
B. HA 资源参数	125
C. HA 资源行为	135
C.1. 资源间的上级、下级和同级关系	135
C.2. 同级资源启动顺序以及资源下级顺序	136
C.2.1. 归类子资源启动和停止顺序	137
C.2.2. 不归类子资源启动和停止顺序	138
C.3. 继承、<资源>块以及重新使用资源	140
C.4. 故障恢复和独立子树	141
C.5. 调整并测试服务和资源顺序	142
D. 命令行工具小结	143
E. 修订记录	145
索引	147

简介

本文档提供有关安装、配置和管理红帽高可用性附加组件元件的信息。红帽高可用性附加组件可让您连接到作为集群使用的一组计算机（称之为节点或者成员）。在这个文档中，集群或者很多集群指的是运行红帽高可用性附加组件的计算机组。

本文档的读者应该有丰富的 Red Hat Enterprise Linux 知识，并理解集群、存储和服务器计算的概念。

本文档由以下部分组成：

- [第 1 章 Red Hat 高可用性附加组件配置和管理概述](#)
- [第 2 章 在配置红帽高可用性附加组件前](#)
- [第 3 章 使用 Conga 配置红帽高可用性附加组件](#)
- [第 4 章 使用 Conga 管理 Red Hat 高可用性附加组件](#)
- [第 7 章 使用命令行工具配置红帽高可用性附加组件](#)
- [第 8 章 使用命令行工具管理红帽高可用性附加组件](#)
- [第 9 章 诊断并修正集群中的问题](#)
- [第 10 章 使用红帽高可用性附加组件进行 SNMP 配置](#)
- [附录 A, Fence 设备参数](#)
- [附录 B, HA 资源参数](#)
- [附录 C, HA 资源行为](#)
- [附录 D, 命令行工具小结](#)
- [附录 E, 修订记录](#)

有关 Red Hat Enterprise Linux 6 的详情请参考以下资源：

- 《Red Hat Enterprise Linux 安装指南》 — 提供有关安装 Red Hat Enterprise Linux 6 的信息。
- 《Red Hat Enterprise Linux 部署指南》 — 提供有关部署、配置和管理 Red Hat Enterprise Linux 6 的信息。

有关 Red Hat Enterprise Linux 6 高可用性附加组件以及相关产品的详情请参考以下资源：

- 《高可用性附加组件概述》 — 提供红帽高可用性附加组件的高级概述。
- 《逻辑卷管理器管理》 — 提供对逻辑卷管理器（LVM）的描述，包括在集群的环境中运行 LVM 的信息。
- 《全局文件系统 2：配置和管理》 — 提供有关安装、配置和维护红帽 GFS2（红帽全局文件系统 2）的信息，GFS2 包含在弹性存储附加组件中。
- 《设备映射器多路径》 — 提供有关使用红帽企业版 Linux 6 设备映射器多路径功能的信息。
- 《负载均衡管理》 — 提供使用负载均衡附加组件配置高性能系统和服务的信息。该组件是一组整合的软件组件，可为在一组真实服务器间平衡 IP 负载提供 Linux 虚拟服务器（LVS）。

- 《发行注记》 — 提供有关红帽产品当前发行本的信息。

高可用性附加组件文档以及其它红帽文档都有 HTML、PDF 和 RPM 版本，您可在红帽企业版 Linux 文档 CD 以及在线文档 <http://docs.redhat.com/> 中找到。

1. 文档约定

本手册使用几个约定来突出某些用词和短语以及信息的某些片段。

在 PDF 版本以及纸版中，本手册使用在 [Liberation 字体](https://fedorahosted.org/liberation-fonts/)¹ 套件中选出的字体。如果您在您的系统中安装了 Liberation 字体套件，它还可用于 HTML 版本。如果没有安装，则会显示可替换的类似字体。请注意：红帽企业 Linux 5 以及其后的版本默认包含 Liberation 字体套件。

1.1. 排版约定

我们使用四种排版约定突出特定用词和短语。这些约定及其使用环境如下。

单行粗体

用来突出系统输入，其中包括 shell 命令、文件名以及路径。还可用来突出按键以及组合键。例如：

要看到文件您当前工作目录中文件 `my_next_bestselling_novel` 的内容，请在 shell 提示符后输入 `cat my_next_bestselling_novel` 命令并按 Enter 键执行该命令。

以上内容包括一个文件名，一个 shell 命令以及一个按键，它们都以固定粗体形式出现，且全部与上下文有所区别。

组合键可通过使用连字符连接组合键的每个部分来与按键区别。例如：

按 Enter 执行该命令。

按 Ctrl+Alt+F2 切换到第一个虚拟终端。Ctrl+Alt+F1 返回您的 X-Windows 会话。

第一段突出的是要按的特定按键。第二段突出了两个按键组合（每个组合都要同时按）。下。

如果讨论的是源码、等级名称、方法、功能、变量名称以及在段落中提到的返回的数值，那么都会以上述形式出现，即固定粗体。例如：

与文件相关的等级包括用于文件系统的 `filesystem`、用于文件的 `file` 以及用于目录的 `dir`。每个等级都有其自身相关的权限。

比例粗体

这是指在系统中遇到的文字或者短语，其中包括应用程序名称、对话框文本、标记的按钮、复选框以及单选按钮标签、菜单标题以及子菜单标题。例如：

在主菜单条中选择「系统」→「首选项」→「鼠标」启动 鼠标首选项。在「按钮」标签中点击「惯用左手鼠标」复选框并点击 关闭切换到主鼠标按钮从左向右（让鼠标适合左手使用）。

要在 gedit 文件中插入一个特殊字符，请在主菜单中选择「应用程序」→「附件」→「字符映射表」。下一步在 字符映射表菜单条中选择「搜索」→「查找」，在「搜索」字段输入字符名称并点击 下一个 按钮。您输入的字符会在「字符表」中突出出

¹ <https://fedorahosted.org/liberation-fonts/>

来。双击这个突出的字符将其放入「要复制的文本」字段，然后点击 复制 按钮。现在切换回您的文档并在 `gedit` 菜单条中选择「编辑」→「粘贴」。

以上文本包括应用程序名称、系统范围菜单名称及项目、应用程序特定菜单名称以及按钮和 GUI 界面中的文本，所有都以比例粗体出现并与上下文区别。

固定粗斜体 或者 比例粗斜体

无论固定粗体或者比例粗体，附加的斜体表示是可替换或者变量文本。斜体表示那些不直接输入的文本或者那些根据环境改变的文本。例如：

要使用 `ssh` 连接到远程机器，请在 `shell` 提示符后输入 `ssh`
`username@domain.name`。如果远程机器是 `example.com` 且您在该其机器中的用户名为
`john`，请输入 `ssh john@example.com`。

`mount -o remount file-system` 命令会重新挂载命名的文件系统。例如：要重新挂载
`/home` 文件系统，则命令为 `mount -o remount /home`。

要查看目前安装的软件包版本，请使用 `rpm -q package` 命令。它会返回以下结果
`: package-version-release`。

请注意以上文字中的粗斜体字 — `username`、`domain.name`、`file-system`、`package`、`version` 和
`release`。无论您输入文本或者运行一个命令，还是该系统显示的文本，每个字都是一个占位符。

不考虑工作中显示标题的标准用法，斜体表示第一次使用某个新且重要的用语。例如：

`Publican` 是一个 `DocBook` 发布系统。

1.2. 抬升式引用约定

终端输出和源代码列表要与周围文本明显分开。

将发送到终端的输出设定为 `Mono-spaced Roman` 并显示为：

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

源码列表也设为 `Mono-spaced Roman`，但添加下面突出的语法：

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
    public static void main(String args[])
        throws Exception
    {
        InitialContext iniCtx = new InitialContext();
        Object          ref    = iniCtx.lookup("EchoBean");
        EchoHome        home   = (EchoHome) ref;
        Echo            echo    = home.create();

        System.out.println("Created Echo");

        System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
    }
}
```

1.3. 备注及警告

最后，我们使用三种视觉形式来突出那些可能被忽视的信息。



备注

备注是对手头任务的提示、捷径或者备选解决方法。忽略提示不会造成负面后果，但您可能会错过一个更省事的诀窍。



重要

重要框中的内容是那些容易错过的事情：配置更改只可用于当前会话，或者在应用更新前要重启的服务。忽略‘重要’框中的内容不会造成数据丢失但可能会让您抓狂。



警告

警告是不应被忽略的。忽略警告信息很可能导致数据丢失。

2. 反馈

如果您在这本手册中发现任何印刷错误，或者您对本手册有改进意见，我们非常乐于倾听！请在 Bugzilla (<http://bugzilla.redhat.com/bugzilla/>) 中根据组件 doc-Cluster_Administration 提交报告。

请确定包含了手册识别符：

Cluster_Administration(EN)-6 (2011-05-19T16:26)

通过这个手册识别符，我们可以了解您使用的具体版本。

如果您有针对文档的建议，请尽量具体描述。如果您发现任何错误，请附带章节号以及上下文，以便我们查找。

Red Hat 高可用性附加组件配置和管理概述

Red Hat 高可用性附加组件可让您连接到作为集群使用的一组计算机（称之为节点或者成员）。您可使用 Red Hat 高可用性附加组件适应您的集群需要（例如：为 GFS2 文件系统中的共享文件设置集群，或者设置服务故障切换）。

本章总结了红帽企业版 Linux 6 初始发行本以后添加到红帽高可用性附加组件的功能和更新文档，并包含 Red Hat 高可用性附加组件配置和管理概述。

1.1. 红帽企业版 Linux 6.1 中新的和更改的功能

红帽企业版 Linux 6.1 包括以下文档和功能更新及更改。

- 从红帽企业版 Linux 6.1 发行本开始，红帽高可用性附加组件提供 SNMP 陷阱支持。有关使用红帽高可用性附加组件配置 SNMP 陷阱的详情请参考 [第 10 章 使用红帽高可用性附加组件进行 SNMP 配置](#)。
- 从红帽企业版 Linux 6.1 发行本开始，红帽高可用性附加组件支持 ccs 集群配置命令。有关 ccs 命令的详情请参考 [第 5 章 使用 ccs 命令配置红帽高可用性附加组件](#) 和 [第 6 章 使用 ccs 管理 Red Hat 高可用性附加组件](#)。
- 已更新使用 Conga 配置和管理红帽高可用性附加组件软件的文档，包括更新的 Conga 页面和功能支持。
- 从红帽企业版 Linux 6.1 发行本开始，使用 ricci 需要在您第一次在任意节点中传播更新的集群配置文件时输入密码。有关 ricci 的详情请参考 [第 2.11 节 “ricci 注意事项”](#)。
- 您现在可为服务指定 Restart-Disable 失败策略，表明该系统应该在其失败的地方尝试重启，但如果重启服务失败，则会禁用该服务而不是将其移动到该集群的另一台主机中。这个功能在 [第 3.9 节 “在集群中添加集群服务”](#) 和 [附录 B, HA 资源参数](#) 中有记录。
- 您现在可以将独立子树配置为 non-critical，表示如果该资源失败，只禁用那个资源。有关这个功能的详情请参考 [第 3.9 节 “在集群中添加集群服务”](#) 和 [第 C.4 节 “故障恢复和独立子树”](#)。
- 这个文档现在包括一个新的章节 [第 9 章 诊断并修正集群中的问题](#)。

另外对整个文档进行了小的修改和说明。

1.2. 配置基础

要设置集群，您必须将节点连接到某些集群硬件，并将该节点配置到集群环境中。配置和管理 Red Hat 高可用性附加组件包括以下基本步骤：

1. 设置硬件。请参考 [第 1.3 节 “设置硬件”](#)。
2. 安装 Red Hat 高可用性附加组件软件。请参考 [第 1.4 节 “安装 Red Hat 高可用性附加组件软件”](#)。
3. 配置 Red Hat 高可用性附加组件软件。请参考 [第 1.5 节 “配置 Red Hat 高可用性附加组件软件”](#)。

1.3. 设置硬件

设置硬件包括将集群节点连接到需要运行 Red Hat 高可用性附加组件的其它硬件。硬件的数量和类型根据集群目的和可用性要求有所不同。通常企业级集群需要以下硬件类型（请参考 [图 1.1 “Red Hat 高可用性附加组件硬件概述”](#)）。如考虑硬件和其它集群配置问题，请参考 [第 2 章 在配置红帽高可用性附加组件前](#)，或者咨询授权 Red Hat 代表。

- 高可用性附加组件节点 — 可运行 Red Hat Enterprise Linux 6，且至少有 1GB 内存的计算机。
- 以太网开关或者公用网络集线器 — 为访问该集群的客户端所需。
- 以太网开关或者专用网络集线器 — 为集群节点间通讯以及其它集群硬件，比如网络电源开关和光纤开关所需。
- 网络电源开关 — 建议在企业级集群中使用网络电源开关执行 fencing。
- 光纤开关 — 光纤开关提供对光纤存储的访问。根据存储接口的不同还有其它可用存储选项，例如：iSCSI。可将光纤开关配置为执行 fencing。
- 存储 — 集群需要的一些存储类型，具体类型要视集群目的而定。

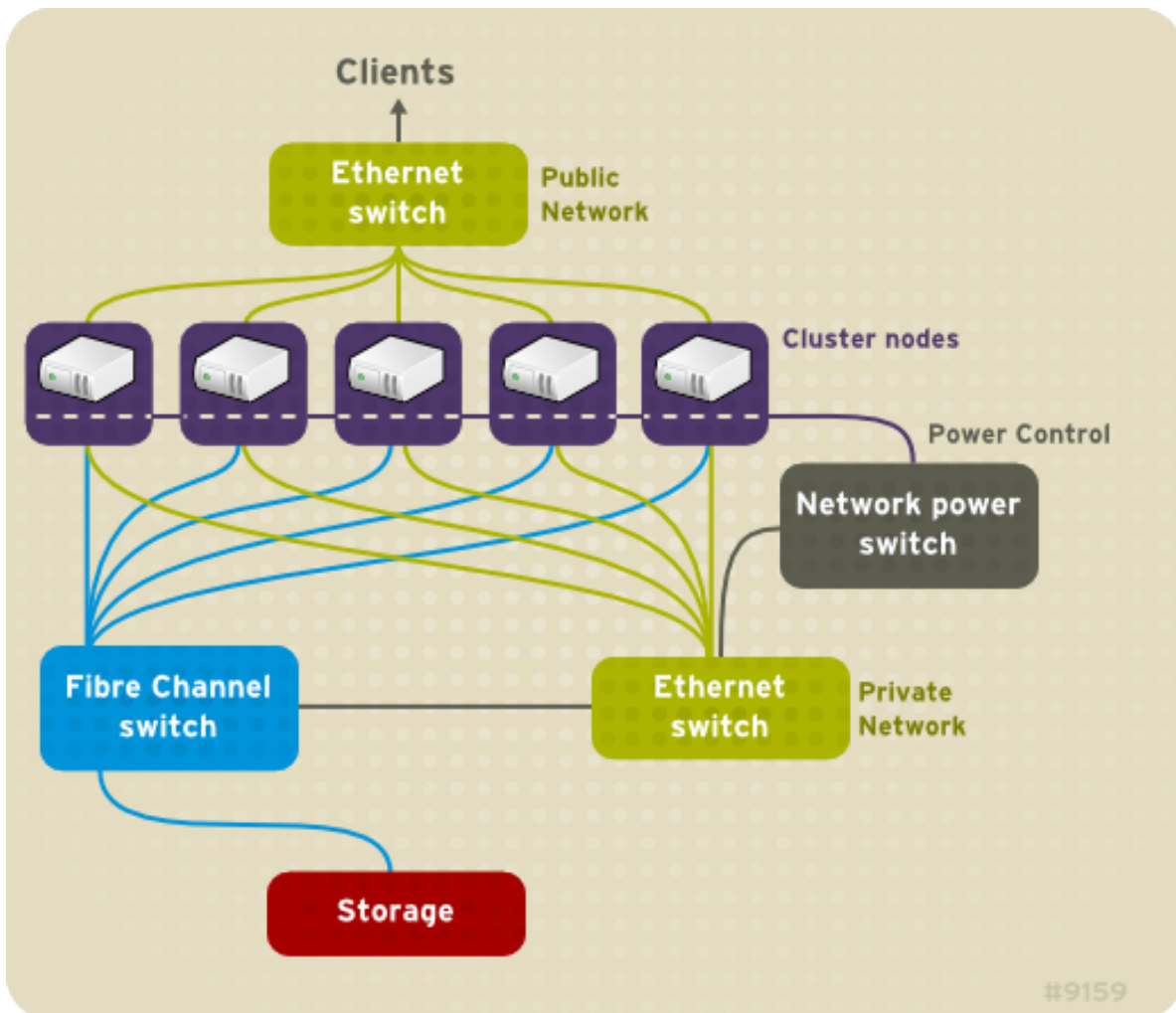


图 1.1. Red Hat 高可用性附加组件硬件概述

1.4. 安装 Red Hat 高可用性附加组件软件

要安装 Red Hat 高可用性附加组件软件，您必须有该软件授权。如果您使用 Conga 配置 GUI，则您可以让它安装集群软件。如果您使用其它工具配置该集群，请使用 Red Hat Enterprise Linux 软件锁定并安装该软件。

升级 Red Hat 高可用性附加组件软件

可在给定红帽企业版 Linux 主要发行本中不单独提出集群的情况下升级集群软件。这样做需要每次在一台主机中禁用该集群软件，升级该软件并在那台主机中重启该集群软件。

1. 在单一集群节点中关闭所有集群服务。有关在一个节点中停止集群软件的步骤，请参考 [第 8.1.2 节 “停止集群软件”](#)。停止 rgmanager 前最好手动重新定位集群管理的服务并使虚拟机脱离该主机。
2. 执行 yum update 命令安装新的 RPM。例如：

```
# yum update -y openais cman rgmanager lvm2-cluster gfs2-utils
```

3. 手动重启集群节点或者集群服务。有关在节点中启动集群服务的步骤请参考 [第 8.1.1 节 “启动集群软件”](#)。

1.5. 配置 Red Hat 高可用性附加组件软件

配置 Red Hat 高可用性附加组件软件包括使用配置工具指定集群组件之间的关系。以下集群配置工具可用于 Red Hat 高可用性附加组件：

- Conga — 这是一个用于安装、配置和管理 Red Hat 高可用性附加组件的综合用户界面。有关使用 Conga 配置和管理高可用性附加组件的详情请参考 [第 3 章 使用 Conga 配置红帽高可用性附加组件](#) 和 [第 4 章 使用 Conga 管理 Red Hat 高可用性附加组件](#)。
- ccs — 这个命令配置和管理 Red Hat 高可用性附加组件。有关使用 ccs 配置和管理高可用性附加组件的详情请参考 [第 5 章 使用 ccs 命令配置红帽高可用性附加组件](#) 和 [第 6 章 使用 ccs 管理 Red Hat 高可用性附加组件](#)。
- 命令行工具 — 这是一组配置和管理 Red Hat 高可用性附加组件的命令行工具。有关使用命令行工具配置和管理集群的详情请参考 [第 7 章 使用命令行工具配置红帽高可用性附加组件](#) 和 [第 8 章 使用命令行工具管理红帽高可用性附加组件](#)。首选命令行工具小结请参考 [附录 D，命令行工具小结](#)。



备注

在 RHEL 6 中不能使用 system-config-cluster。

在配置红帽高可用性附加组件前

本章描述了在安装和配置红帽高可用性附加组件前要执行的任务以及要考虑的问题，由以下小节组成：



重要

确定您部署的红帽高可用性附加组件满足您的需要并可支持。部署前联络授权红帽代表确认您的配置。另外预留充分时间测试失败模式。

- 第 2.1 节 “常规配置注意事项”
- 第 2.2 节 “兼容硬件”
- 第 2.3 节 “启用 IP 端口”
- 第 2.4 节 “将 ACPI 配置为使用整合的 Fence 设备”
- 第 2.5 节 “配置 HA 服务注意事项”
- 第 2.6 节 “配置验证”
- 第 2.7 节 “NetworkManager 注意事项”
- 第 2.8 节 “使用仲裁磁盘的注意事项”
- 第 2.9 节 “红帽高可用性附加组件及 SELinux”
- 第 2.10 节 “多播地址”
- 第 2.11 节 “ricci 注意事项”

2.1. 常规配置注意事项

您可使用各种方法配置红帽高可用性附加组件以适合您的需要。当您进行计划、配置和实施您的部署时请考虑以下常规注意事项：

支持的集群节点数

红帽高可用性附加组件最多支持的集群节点数为 16。

单点集群

现在只能完全支持单点集群。官方不支持在多个网络位置中分布的集群。有关详情以及多点集群的讨论，请联络您的红帽销售或者支持代表。

GFS2

虽然 GFS2 文件系统可作为独立系统使用，也可作为集群配置的一部分，但红帽不支持将 GFS2 作为单节点文件系统使用。红帽支持很多为单节点优化的高性能单节点文件系统，它们相对集群文件系统来说消耗更低。红帽建议您在那些只需要在单一节点挂载文件系统的情况下首选那些系统而不是 GFS2。红帽将继续为现有客户支持单节点 GFS2 文件系统。

当您将在 GFS2 文件系统作为集群文件系统配置时，您必须确定该集群中的所有节点都可访问共享的文件系统。不支持不对称集群配置，在这个样的集群中有些节点可访问该文件系统而其它节点则不能。这不要求所有节点确实挂载该 GFS2 文件系统。

无单点故障硬件配置

集群可包括一个双控制器 RAID 阵列、多绑定链路、集群成员和存储间的多路径以及冗余不间断供电（UPS）系统以保证没有单点故障造成的应用程序失败或者数据丢失。

另外，可设置一个低消耗集群以提供比无单点故障集群低的可用性。例如：您可以设置一个使用单控制器 RAID 阵列和只使用单以太网链路的集群。

某些低消耗备选方案，比如主机 RAID 控制器、不支持集群的软件 RAID 以及多启动器并行 SCSI 配置与共享集群存储不兼容或者不适合作为共享集群存储使用。

确保数据完整

要保证数据完整，则每次只能有一个节点可运行集群服务并访问集群服务数据。在集群硬件配置中使用电源开关可让一个节点在故障切换过程中重启节点 HA 服务前为另一个节点提供动力。这样就可防止两个节点同时访问同一数据并毁损数据。强烈建议使用 Fence 设备（远程电源、关闭和重启集群节点的硬件或者软件解决方案）以确保在所有失败情况下数据的完整性。

以太网通道绑定

集群仲裁以及节点是否正常运行是由在通过以太网在集群节点间的沟通信息确定的。另外，集群节点使用以太网执行各种重要集群功能（例如：fencing）。使用以太网通道绑定，可将多个以太网接口配置为作为一个接口动作，这样就减小了在集群节点间以及其它集群硬件间典型切换的以太网连接单点故障风险。

IPv4 和 IPv6

高可用性附加组件支持 IPv4 和 IPv6 互联网协议。在高可用性附加组件中支持 IPv6 是红帽企业版 Linux 6 的新功能。

2.2. 兼容硬件

在配置红帽高可用性附加组件软件前，请确定您的集群使用合适的硬件（例如：支持的 fence、存储以及光纤切换的设备）。有关大多数当前硬件兼容性信息，请参考硬件配置指南 http://www.redhat.com/cluster_suite/hardware/。

2.3. 启用 IP 端口

在部署红帽高可用性附加组件前，您必须在集群以及运行 luci（Conga 用户界面服务器）的计算机中启用某些 IP 端口。以下小节指出要启用的 IP 端口：

- 第 2.3.1 节 “在集群节点中启用 IP 端口”
- 第 2.3.2 节 “在运行 luci 的计算机中启用 IP 端口”

2.3.1. 在集群节点中启用 IP 端口

要允许红帽高可用性附加组件节点与其它节点沟通，您必须启用为某些红帽高可用性附加组件分配的 IP 端口。表 2.1 “在红帽高可用性附加组件节点中启用 IP 端口” 列出了 IP 端口号、其对应协议以及使用这些端口号的组件。在每个集群节点中，根据表 2.1 “在红帽高可用性附加组件节点中启用 IP 端口” 启用 IP 端口。您可以使用 `system-config-firewall` 启用 IP 端口。

表 2.1. 在红帽高可用性附加组件节点中启用 IP 端口

IP 端口号	协议	组件
5404, 5405	UDP	corosync/cman（集群管理器）
11111	TCP	ricci（传播更新的集群信息）

IP 端口号	协议	组件
21064	TCP	d1m (发布的锁定管理器)
16851	TCP	modclusterd

2.3.2. 在运行 luci 的计算机中启用 IP 端口

要让客户端计算机与运行 luci (Conga 用户界面服务器) 的计算机沟通, 您必须启用分配给 luci 的 IP 端口。请在每台运行 luci 的计算机中根据 表 2.2 “在运行 luci 的计算机中启用的 IP 端口” 启用 IP 端口。



备注

如果集群节点正在运行 luci, 则应该已经启用了端口 11111。

表 2.2. 在运行 luci 的计算机中启用的 IP 端口

IP 端口号	协议	组件
8084	TCP	luci (Conga 用户接口服务器)

2.4. 将 ACPI 配置为使用整合的 Fence 设备

如果您的集群使用整合的 fence 设备, 则您必须配置 ACPI (高级配置和电源界面) 以保证迅速和完全的 fencing。



备注

有关红帽高可用性附加组件支持的整合 fence 设备的最新信息请参考 http://www.redhat.com/cluster_suite/hardware/¹。

如果将集群节点配置为使用整合的 fence 设备执行 fence 操作, 则需要为该节点禁用 ACPI 软关机。禁用 ACPI 软关机可让整合的 fence 设备立即完全关闭节点, 而不是尝试彻底关闭 (例如: shutdown -h now)。否则, 如果启用了 ACPI 软关闭, 整合的 fence 设备将多消耗至少 4 秒时间关闭一个节点 (请参考下面的备注)。另外, 如果启用 ACPI 软关闭, 且在关闭过程中出现节点 panic 或者停滞, 则整合的 fence 设备将无法关闭该节点。在那些情况下, fencing 将被延迟或者失败。结果是当使用整合的 fence 设备对节点执行 fence 操作并启用 ACPI 软关闭时, 集群恢复会很慢并需要管理介入方可恢复。

¹ http://www.redhat.com/cluster_suite/hardware/



备注

Fence 一个节点所需时间取决于所使用的整合 fence 设备。有些整合 fence 设备的功能与长按电源开关一致，因此 fence 设备可在 4-5 秒内关闭该节点。其它整合 fence 设备性能与按一下电源开关一致，要依靠操作系统关闭该节点，因此 fence 设备关闭该节点的时间要大大超过 4-5 秒钟。

要禁用 ACPI 软关闭，请使用 `chkconfig` 管理并确认在执行 fence 操作后可立即关闭该节点。禁用 ACPI 软关闭的首选方法是使用 `chkconfig` 管理；但如果这个方法不适用于您的集群，您可以使用以下备选方法之一禁用 ACPI 软关闭：

- 将 BIOS 设置改为 "instant-off" 或者无延迟关闭该节点的对等设置



备注

使用 BIOS 禁用 ACPI 软关闭可能不适用于某些计算机。

- 在 `/boot/grub/grub.conf` 文件的内核引导命令行中附加 `acpi=off`



重要

这个方法可完全禁用 ACPI。有些计算机在完全禁用 ACPI 时无法正常引导。只有在其它方法对您的集群都无效时才使用这个方法。

以下小节提供禁用 ACPI 软关闭的首选和备用方法步骤：

- 第 2.4.1 节 “使用 `chkconfig` 管理禁用 ACPI 软关闭” — 首选方法
- 第 2.4.2 节 “使用 BIOS 禁用 ACPI 软关闭” — 第一备选方法
- 第 2.4.3 节 “在 `grub.conf` 文件中完全禁用 ACPI。” — 第二备选方法

2.4.1. 使用 `chkconfig` 管理禁用 ACPI 软关闭

您可以使用 `chkconfig` 管理，通过在 `chkconfig` 管理中删除 ACPI 守护进程 (`acpid`)，或者关闭 `acpid` 禁用 ACPI 软关闭。

**备注**

这是禁用 ACPI 软关闭的首选方法。

使用 `chkconfig` 管理在每个集群节点中禁用 ACPI 软关闭，如下：

1. 运行以下命令之一：

- `chkconfig --del acpid` — 这个命令从 `chkconfig` 管理中删除 `acpid`。
- 或者 —
- `chkconfig --level 2345 acpid off` — 这个命令关闭 `acpid`。

2. 重启该节点。

3. 当配置并运行集群时，确认在执行 `fence` 时立即关闭该节点。

**备注**

您可使用 `fence_node` 命令或者 `Conga fence` 该节点。

2.4.2. 使用 BIOS 禁用 ACPI 软关闭

禁用 ACPI 软关闭的首选方法是 `chkconfig` 管理（第 2.4.1 节 “使用 `chkconfig` 管理禁用 ACPI 软关闭”）。但如果首选方法对您的集群无效，请按照本节中的步骤执行。

**备注**

使用 BIOS 禁用 ACPI 软关闭可能不适用于某些计算机。

您可以通过配置每个集群节点中的 BIOS 禁用 ACPI 软关闭，如下：

1. 重启该节点并启动 BIOS CMOS Setup Utility 程序。
2. 浏览「电源」菜单（或者对等的电源管理菜单）
3. 在「电源」菜单中将「Soft-Off by PWR-BTTN」功能（或者对等的功能）设定为「Instant-Off」（或者使用电源按钮无延迟关闭节点的对等设置）。在 例 2.1 “BIOS CMOS Setup Utility: 将「使用电源按钮软关闭」设定为「立即关闭」。” 中将「电源」菜单中的「ACPI Function」设定为「Enabled」，并将「Soft-Off by PWR-BTTN」设定为「Instant-Off」的。



备注

与「ACPI Function」、「Soft-Off by PWR-BTTN」和「Instant-Off」对等的菜单在不同计算机中会有所不同。但这个步骤的目的是配置 BIOS 以便计算机可使用电源开关无延迟地关闭计算机。

- 4. 退出 BIOS CMOS Setup Utility 程序，保存 BIOS 配置。
- 5. 当配置并运行集群时，确认在执行 fence 时立即关闭该节点。



备注

您可使用 `fence_node` 命令或者 `Conga fence` 该节点。

例 2.1. BIOS CMOS Setup Utility: 将「使用电源按钮软关闭」设定为「立即关闭」。

+-----+-----+-----+		
	ACPI Function	[Enabled]
	ACPI Suspend Type	[S1 (POS)]
	x Run VGABIOS if S3 Resume	Auto
	Suspend Mode	[Disabled]
	HDD Power Down	[Disabled]
	Soft-Off by PWR-BTTN	[Instant-Off]
	CPU THRM-Throttling	[50.0%]
	Wake-Up by PCI card	[Enabled]
	Power On by Ring	[Enabled]
	Wake Up On LAN	[Enabled]
	x USB KB Wake-Up From S3	Disabled
	Resume by Alarm	[Disabled]
	x Date(of Month) Alarm	0
	x Time(hh:mm:ss) Alarm	0 : 0 :
	POWER ON Function	[BUTTON ONLY]
	x KB Power ON Password	Enter
	x Hot Key Power ON	Ctrl-F1
+-----+-----+-----+		

这个示例演示了如何将「ACPI Function」设定为「Enabled」；将「Soft-Off by PWR-BTTN」设定为「Instant-Off」。

2.4.3. 在 `grub.conf` 文件中完全禁用 ACPI。

禁用 ACPI 软关闭的首选方法是使用 `chkconfig` 管理（第 2.4.1 节 “使用 `chkconfig` 管理禁用 ACPI 软关闭”）。如果首选的方法不适用于您的集群，您可以使用 BIOS 电源管理（第 2.4.2 节 “使用 BIOS 禁用 ACPI 软关闭”）禁用 ACPI 软关闭。如果这两种方法都不适用于您的集群，您可以在 `grub.conf` 文件的内核引导命令行中附加 `acpi=off`，这样就可以完全禁用 ACPI。



重要

这个方法可完全禁用 ACPI。有些计算机在完全禁用 ACPI 时无法正常引导。只有在其它方法对您的集群都无效时才使用这个方法。

您可以通过编辑每个集群节点的 `grub.conf` 文件完全禁用 ACPI，如下：

1. 在文本编辑器中打开 `/boot/grub/grub.conf`。
2. 在 `/boot/grub/grub.conf` 的内核引导命令行中附加 `acpi=off`（请参考 [例 2.2 “附加了 `acpi=off` 的内核引导命令行”](#)）。
3. 重启该节点。
4. 当配置并运行集群时，确认在执行 `fence` 时立即关闭该节点。



备注

您可使用 `fence_node` 命令或者 `Conga fence` 该节点。

例 2.2. 附加了 `acpi=off` 的内核引导命令行

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#         all kernel and initrd paths are relative to /boot/, eg.
#         root (hd0,0)
#         kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol100
#         initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=5
serial --unit=0 --speed=115200
terminal --timeout=5 serial console
title Red Hat Enterprise Linux Server (2.6.18-36.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-36.el5 ro root=/dev/VolGroup00/LogVol100 console=ttyS0,115200n8 acpi=off
    initrd /initrd-2.6.18-36.el5.img
```

在这个示例中，在内核引导命令行中附加了 `acpi=off` — 以 `"kernel /vmlinuz-2.6.18-36.el5"` 开始的行。

2.5. 配置 HA 服务注意事项

您可以通过配置 HA（高可用性）服务创建集群以满足您对高可用性的要求。红帽高可用性附加组件中的 HA 服务管理关键组件 `rgmanager` 可为现成的应用程序部署冷故障切换。在红帽高可用性附加组件

中，使用其它集群资源配置的应用程序可组成一个 HA 服务，该服务可在集群节点间进行故障切换而不会中断集群客户端。如果一个集群节点失败，或者集群系统管理员在集群节点间移动该服务时（例如：计划的集群节点停转），则会发生 HA 服务故障切换。

要创建 HA 服务，您必须在集群配置文件中配置它。HA 服务会包含集群资源。集群资源构建您要在集群配置文件中创建和管理的块 — 例如：IP 地址、应用程序初始化脚本或者红帽 GFS2 共享分区。

HA 服务一次只能在一个集群节点中运行以保持数据完整性。您可以在故障切换域中指定故障切换优先性。指定故障切换优先性包括在故障切换域中为每个节点分配优先性等级。如果您没有指定故障切换优先性，则 HA 服务可故障切换到其故障切换域中的任何节点。另外，您还可以指定是否将 HA 服务严格限制在与其故障切换域关联的节点中。（当与非限制故障切换域关联时，HA 服务可在故障切换域所有成员都不可用的事件中在任意节点中启动。）

图 2.1 “网页服务器集群服务示例” 演示名为 “content-webserver” 的网页服务器的 HA 服务示例。它在集群节点 B 中运行，且其故障切换域由节点 A、B 和 D 组成。另外，使用故障切换优先权将故障切换域配置为节点 D 的优先权高于节点 A，并限制为只切换到故障切换域中的节点。HA 服务由以下集群资源组成：

- IP 地址资源 — IP 地址 10.10.10.201。
- 名为 “httpd-content” 的应用程序资源 — 网页服务器应用程序初始化脚本 /etc/init.d/httpd（指定 httpd）。
- 文件系统资源 — 名为 “gfs2-content-webserver” 的 Red Hat GFS2。

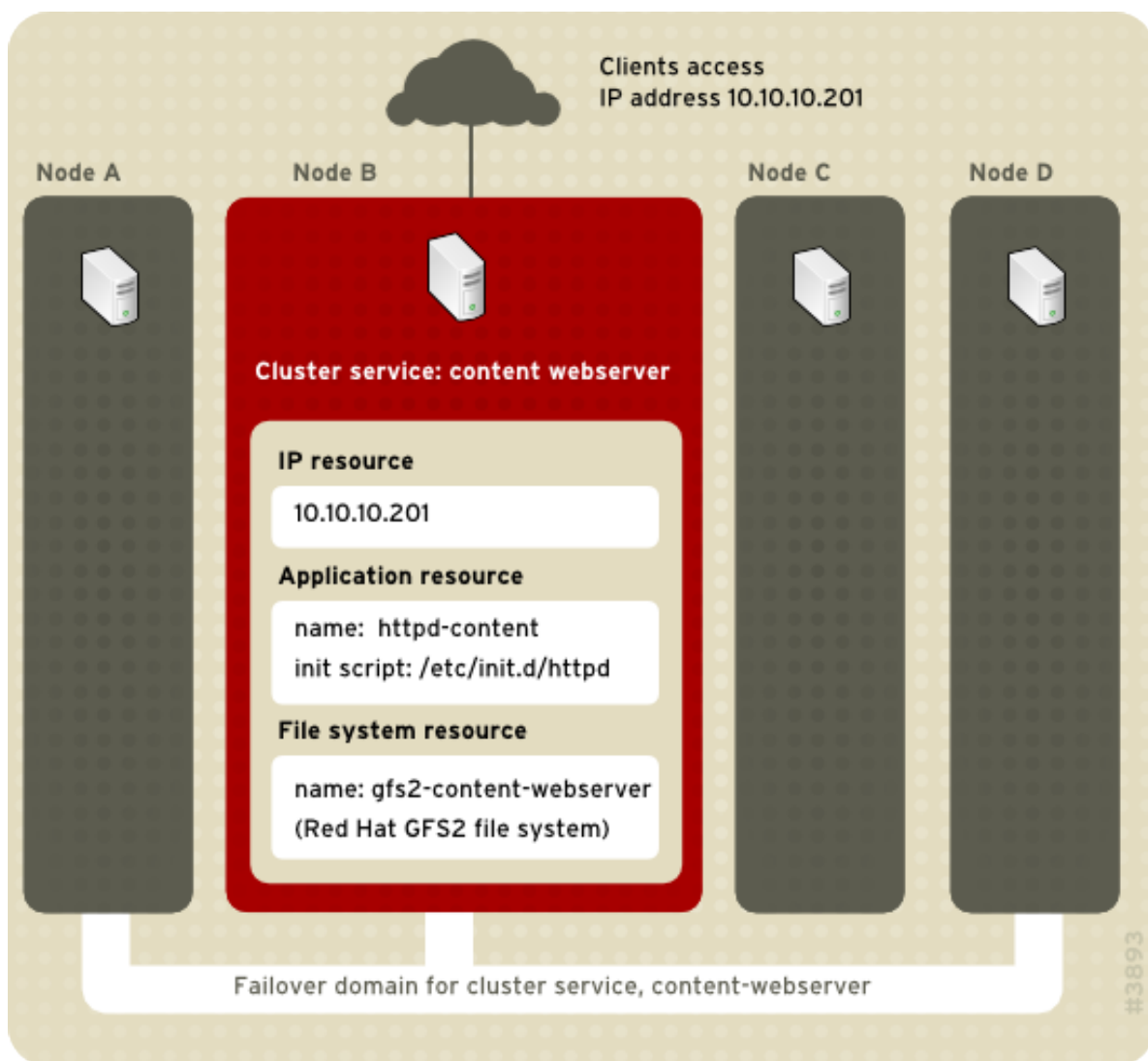


图 2.1. 网页服务器集群服务示例

客户端使用 IP 地址 10.10.10.201 访问 HA 服务，启用与网页服务器程序 httpd-content 间的互动。httpd-content 程序使用 gfs2-content-webserver 文件系统。如果节点 B 失败，则 content-webserver HA 服务会切换到节点 D。如果节点 D 不可用或者也失败了，则该服务会切换到节点 A。故障切换对集群客户端造成很小的服务中断。例如：在 HTTP 服务中，可能会丢失某些状态信息（比如会话数据）。在另一个集群节点中可使用与故障切换前相同的 IP 地址访问 HA 服务。



备注

有关 HA 服务以及故障切换域的详情请参考《高可用性附加组件概述》。有关配置故障切换域的详情请参考 [第 3 章 使用 Conga 配置红帽高可用性附加组件](#)（使用 Conga）或者 [第 7 章 使用命令行工具配置红帽高可用性附加组件](#)（使用命令行工具）。

HA 服务是一组在统一实体中配置的集群资源，可为客户端提供特定的服务。HA 服务在集群配置文件 `/etc/cluster/cluster.conf`（在每个集群节点中）中以资源树的形式出现。在集群配置文件中，每个资源树都使用一个 XML 代表，可指定每个资源及其属性和在资源树中与其它资源的关系（上级、下级和平级关系）。



备注

因为 HA 服务由分为层次树的资源组成，所有服务有时也指的是资源树或者资源组。这两个词组与 HA 服务有相同的含义。

在每个资源树的顶端是一个特殊的资源类型 — 服务资源。其它资源类型构成服务的其它部分，决定服务的特点。配置 HA 服务包括创建服务资源、创建下级集群资源以及将其组成统一实体形成该服务的分级限制。

高可用性附加组件支持以下 HA 服务：

- Apache
- Application (Script)
- LVM (HA LVM)
- MySQL
- NFS
- Open LDAP
- Oracle
- PostgreSQL 8
- Samba
- SAP
- Tomcat 6

配置 HA 服务时两个主要考虑的问题：

- 创建服务所需的资源类型
- 资源间的上级、下级和同级关系

资源类型以及资源的结构由您所要配置服务的类型决定。

集群资源位于 [附录 B](#)，[HA 资源参数](#)。有关资源间上级、下级和同级关系的描述，请参考 [附录 C](#)，[HA 资源行为](#)。

2.6. 配置验证

可在启动和重新载入配置时，根据集群方案 `/usr/share/cluster/cluster.rng` 自动验证集群配置。您还可以使用 `ccs_config_validate` 命令在任意时间验证集群配置。

您可在 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中查看注释的可用方案（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。

配置验证可检查以下基本错误：

- XML 验证 — 查看该配置文件是否为有效 XML 文件。
- 配置选项 — 查看选项（XML 元素和属性）是否有效。
- 选项值 — 查看选项是否包含有效数据（受限制的）

以下示例为演示有效性检查的有效配置和无效配置：

- 有效配置 — 例 2.3 “cluster.conf 示例配置：有效文件”
- 无效 XML — 例 2.4 “cluster.conf 示例配置：无效 XML”
- 无效选项 — 例 2.5 “cluster.conf 示例配置：无效选项”
- 无效选项值 — 例 2.6 “cluster.conf 示例配置：无效选项值”

例 2.3. cluster.conf 示例配置：有效文件

```
<cluster name="mycluster" config_version="1">
  <logging debug="off"/>
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
  </fencedevices>
  <rm>
  </rm>
</cluster>
```

例 2.4. cluster.conf 示例配置：无效 XML

```
<cluster name="mycluster" config_version="1">
  <logging debug="off"/>
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
      </fence>
```

```

    </clusternode>
  </clusternodes>
  <fencedevices>
</fencedevices>
<rm>
</rm>
<cluster>      <-----INVALID

```

在这个示例中，配置的最后一行（在此注释为 "INVALID"）缺少一个斜杠 — 应该是 `</cluster>` 而不是 `<cluster>`。

例 2.5. cluster.conf 示例配置：无效选项

```

<cluster name="mycluster" config_version="1">
  <logging debug="off"/>      <-----INVALID
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
</fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
</fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
</fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
</fencedevices>
<rm>
</rm>
<cluster>

```

在这个示例中配置的第二行（在此注释为 "INVALID"）包含无效 XML 元素 — 应该是 `logging` 而不是 `loging`。

例 2.6. cluster.conf 示例配置：无效选项值

```

<cluster name="mycluster" config_version="1">
  <logging debug="off"/>
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="-1"> <-----INVALID
      <fence>
</fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
</fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
</fence>

```

```

    </clusternode>
  </clusternodes>
  <fencedevices>
</fencedevices>
  <rm>
</rm>
<cluster>

```

在这个示例中，配置的第四行（在此注释为 "INVALID"）包含 XML 属性的无效值，即 node-01.example.com 的 clusternode 行中的 nodeid。该值应该是一个正数 ("1") 而不是负数 ("-1")。nodeid 属性值必须是一个正数。

2.7. NetworkManager 注意事项

不支持在集群节点中使用 NetworkManager。如果您已经在集群节点中安装了 NetworkManager，您应该删除或者禁用该程序。



注意

如果 NetworkManager 正在运行，或者将其配置为与 chkconfig 命令一同运行，则不会启动 cman 服务。

2.8. 使用仲裁磁盘的注意事项

仲裁磁盘是使用磁盘的仲裁守护进程 qdiskd，它可提供补充的试探法 (heuristics) 以确定节点是否正常运行。使用这些试探法，您可以确定在网络分区事件中对节点操作十分重要的因素。例如：在一个按 3:1 分割的有四个节点的集群中，最初三个节点自动“获胜”，因为三对一的占优。在那些情况下，只有一个节点被 fence。但使用 qdiskd，您可以设定试探法以便允许一个节点因访问重要资源获胜（例如：关键网络路径）。如果您的集群需要额外的方法确定节点工作正常，那么您应该将 qdiskd 配置为满足那些要求。



备注

不需要配置 qdiskd，除非您对节点正常工作有特殊的要求。例如："all-but-one" 配置。在 all-but-one 配置中会将 qdiskd 配置为提供足够的仲裁投票以便在只有一个节点工作时维护仲裁。



重要

总之，用于您部署的试探法以及其它 qdiskd 参数要根据所需环境和特殊要求而定。要了解启发式和其它 qdiskd 参数，请参考 qdisk(5) man page。如果您在了解和使用 qdiskd 方面需要帮助，请联络授权 Red Hat 支持代表。

如果您需要使用 `qdiskd`，请考虑以下注意事项：

集群节点投票

使用仲裁磁盘时，每个集群节点都必须有一票。

CMAN 成员超时值

CMAN 成员超时值（即 CMAN 认为节点已死，并不再是成员前该节点不响应的的时间）应该至少为 `qdiskd` 成员超时值的两倍。理由是仲裁守护进程必须自己探测失败的节点，且比 CMAN 要花更多的时间。CMAN 成员超时默认值为 10 秒。其他具体位置条件可能会影响 CMAN 和 `qdiskd` 成员超时值之间的关系。要在调整 CMAN 成员超时值方面获得帮助，请联络授权 Red Hat 支持代表。

Fencing

要在使用 `qdiskd` 时保证可靠的 fencing，请使用电源 fencing。虽然其它类型的 fencing 在没有配置 `qdiskd` 的集群中可靠，但它们并不适用于配置了 `qdiskd` 的集群。

最多节点数

配置了 `qdiskd` 的集群最多可支持 16 个节点。这个限制的原因是因为可扩展性的需要，增加节点计数会增加共享仲裁磁盘设备中的同步 I/O 竞争。

仲裁磁盘设备

仲裁磁盘设备应该是集群中所有节点可同时读取/写入的共享块设备。该块设备最小为 10MB。`qdiskd` 可使用的共享块设备示例有多端口 SCSI RAID 阵列、光纤 RAID SAN 或者配置为 RAID 的 iSCSI 目标。您可以使用群集仲裁磁盘工具 `mkqdisk` 创建仲裁磁盘设备。有关使用该工具的详情请参考 `mkqdisk(8)` man page。



备注

不建议使用 JBOD 作为仲裁磁盘。JBOD 无法提供可靠的性能，因此可能无法保证节点迅速写入仲裁磁盘。如果某个节点无法迅速写入仲裁磁盘设备，则会错误地将该节点从集群中驱除。

2.9. 红帽高可用性附加组件及 SELinux

红帽企业版 Linux 6 的高可用性附加组件在将 SELinux 策略类型设定为 `targeted` 时支持 SELinux 的 `enforcing` 状态。

有关 SELinux 的详情请参考红帽企业版 Linux 6 《部署指南》。

2.10. 多播地址

红帽高可用性附加组件节点使用多播地址彼此沟通。因此，必须将红帽高可用性附加组件中的每个网络开关以及相关网设备配置为启用多播地址并支持 IGMP（互联网组群管理协议）。确定红帽高可用性附加组件中的每个网络开关以及关联的网设备都可支持多播地址和 IGMP；之后，确定启用多播地址和 ICMP。没有多播和 IGMP，则不是所有节点都会加入集群，从而导致集群失败。



备注

配置网络开关以及相关联网设备的步骤在每个产品中都有所不同。请参考正确的零售商文档或者其它有关配置网络开关以及相关联网设备启用多播地址和 IGMP 的信息。

2.11. ricci 注意事项

在红帽企业版 Linux 6 中使用 ricci 替换 ccsd。因此必需在每个集群节点中都运行 ricci 方可传播更新的集群配置，您可以使用 `cman_tool -r`、`ccs` 命令或者 `luci` 用户界面服务器传播。您可以使用 `service ricci start` 启动 ricci，也可以在引导时使用 `chkconfig` 启动它。有关为 ricci 启用 IP 端口的详情请参考 [第 2.3.1 节 “在集群节点中启用 IP 端口”](#)。

从红帽企业版 Linux 6.1 发行本开始，您在任意节点中使用 ricci 传播更新的集群配置时要求输入密码。您在系统中安装 ricci 后使用 `passwd ricci` 命令为用户 ricci 将 ricci 密码设定为 root。

使用 Conga 配置红帽高可用性附加组件

本章描述了如何使用 Conga 配置红帽高可用性附加组件。有关使用 Conga 管理运行的集群的详情请参考 [第 4 章 使用 Conga 管理 Red Hat 高可用性附加组件](#)。



注意

Conga 是您用来管理红帽高可用性附加组件的图形用户界面。请注意，要有效使用这个界面，您需要对一些基础概念有明确和完整的理解。不建议您在用户界面中摸索可用功能了解集群配置，因为这样可使系统在组件失败时不足以保证所有服务运行。

本章由以下节组成:

- [第 3.1 节 “配置任务”](#)
- [第 3.2 节 “启动 luci”](#)
- [第 3.3 节 “创建集群”](#)
- [第 3.4 节 “全局集群属性”](#)
- [第 3.5 节 “配置 Fence 设备”](#)
- [第 3.6 节 “为集群成员配置 Fencing”](#)
- [第 3.7 节 “配置故障切换域”](#)
- [第 3.8 节 “配置全局集群资源”](#)
- [第 3.9 节 “在集群中添加集群服务”](#)

3.1. 配置任务

使用 Conga 配置红帽高可用性附加组件包括以下步骤:

1. 配置并运行 Conga 配置用户界面 — luci 服务器。请参考 [第 3.2 节 “启动 luci”](#)。
2. 创建集群。请参考 [第 3.3 节 “创建集群”](#)。
3. 配置全局集群属性。请参考 [第 3.4 节 “全局集群属性”](#)。
4. 配置 fence 设备。请参考 [第 3.5 节 “配置 Fence 设备”](#)。
5. 为集群成员配置 fencing。请参考 [第 3.6 节 “为集群成员配置 Fencing”](#)。
6. 创建故障切换域。请参考 [第 3.7 节 “配置故障切换域”](#)。
7. 创建资源。请参考 [第 3.8 节 “配置全局集群资源”](#)。
8. 创建集群服务。请参考 [第 3.9 节 “在集群中添加集群服务”](#)。

3.2. 启动 luci



安装 ricci

使用 luci 配置集群要求在集群节点中安装并运行 ricci，如 [第 2.11 节 “ricci 注意事项”](#) 所述。在该小节中指出，使用 ricci 需要一个密码，您在创建集群时需要为每个集群节点输入该密码，如 [第 3.3 节 “创建集群”](#) 所述。

在启动 luci 前，请确定您集群节点中的 IP 端口允许任意与 luci 沟通的节点中的 luci 服务器到端口 11111 的连接。有关在集群节点中启用 IP 端口的详情请参考 [第 2.3.1 节 “在集群节点中启用 IP 端口”](#)。

要使用 Conga 管理红帽高可用性附加组件，请安装并运行 luci，如下：

1. 选择托管 luci 的主机并在那台计算机中安装 luci 软件。例如：

```
# yum install luci
```



备注

通常在服务器架或者托管 luci 的数据中心中，但集群计算机也可托管 luci。

2. 使用 `service luci start` 启动 luci。例如：

```
# service luci start
Starting luci: generating https SSL certificates... done
[ OK ]

Please, point your web browser to https://nano-01:8084 to access luci
```

3. 在网页浏览器的 URL 地址栏中输入 luci 服务器的 URL，并点击按钮 Go（或者对等的按钮）。luci 服务器的 URL 语法为 `https://luci_server_hostname:8084`。您第一次访问 luci 时，网页浏览器会根据显示的自我签名 SSL 证书（luci 服务器的证书）给出具体提示。确认一个或者多个对话框后您的网页显示器会显示 luci 登录页面。
4. 在 luci 登录页面中输入托管 luci 的系统中任意用户的认证信息。
5. 登录后，luci 会显示「Homebase」页面，如 [图 3.1 “luci Homebase 页面”](#) 所示。

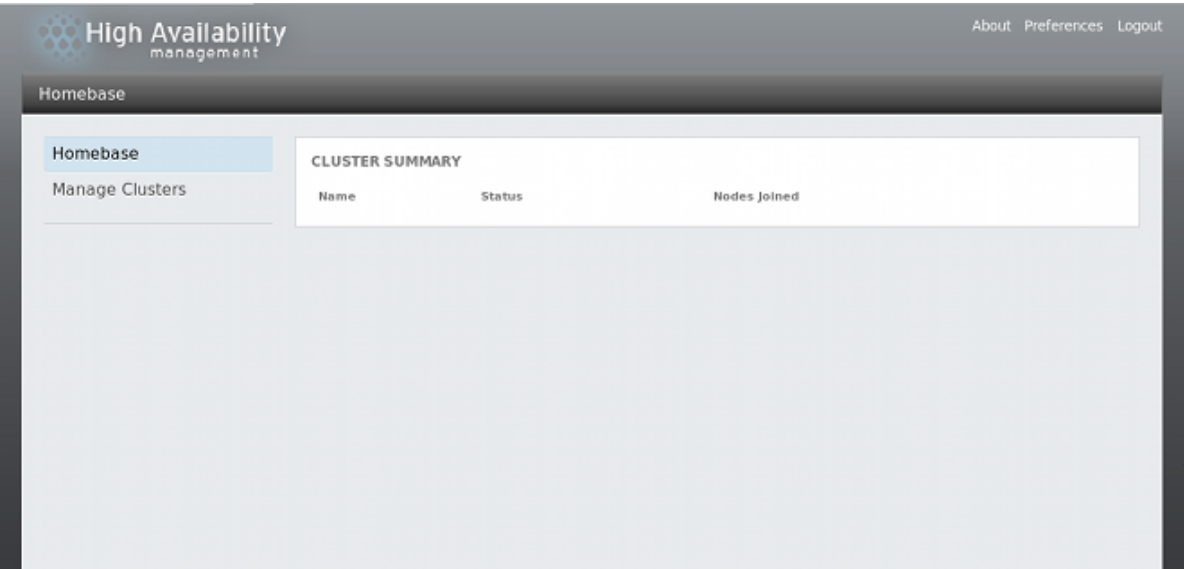


图 3.1. luci Homebase 页面

3.3. 创建集群

使用 luci 创建集群包括命名集群、在集群中添加集群节点、为每个节点输入 ricci 密码并提交创建集群请求。如果节点信息和密码正确，则 Conga 会自动在集群节点中安装软件（如果当前没有安装适当的软件包）并启动集群。按如下步骤创建集群：

1. 在 luci 「Homebase」页面左侧菜单中点击「管理集群」。此时会出现集群页面，如 所示。

图 3.2 “luci 集群管理页面”

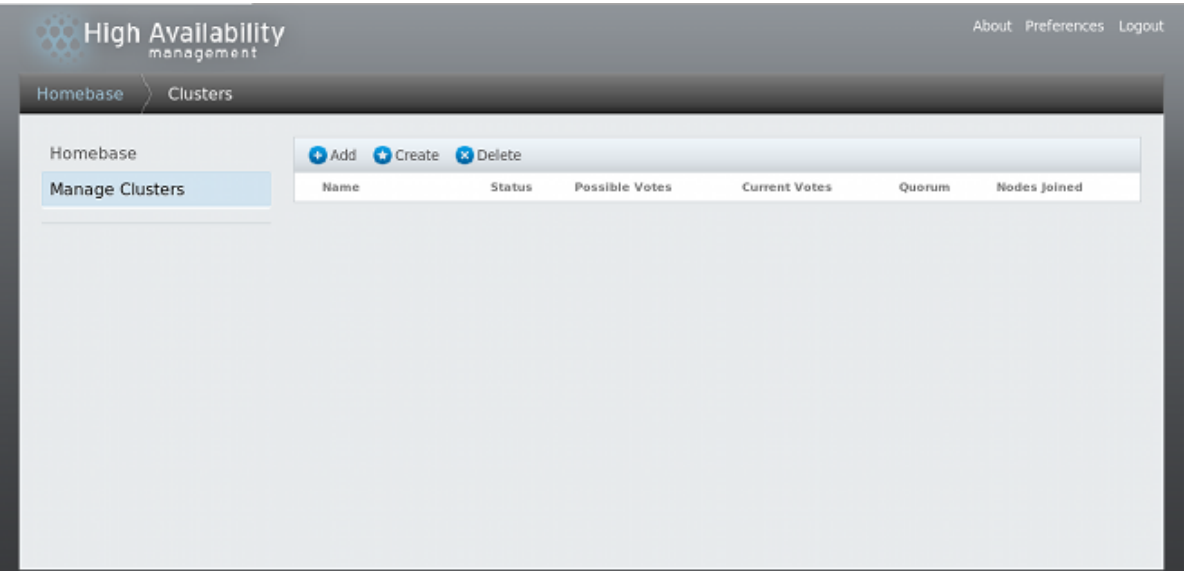


图 3.2. luci 集群管理页面

2. 点击「创建」后出现「创建集群页面」，如 图 3.3 “创建 luci 集群对话框” 所示。

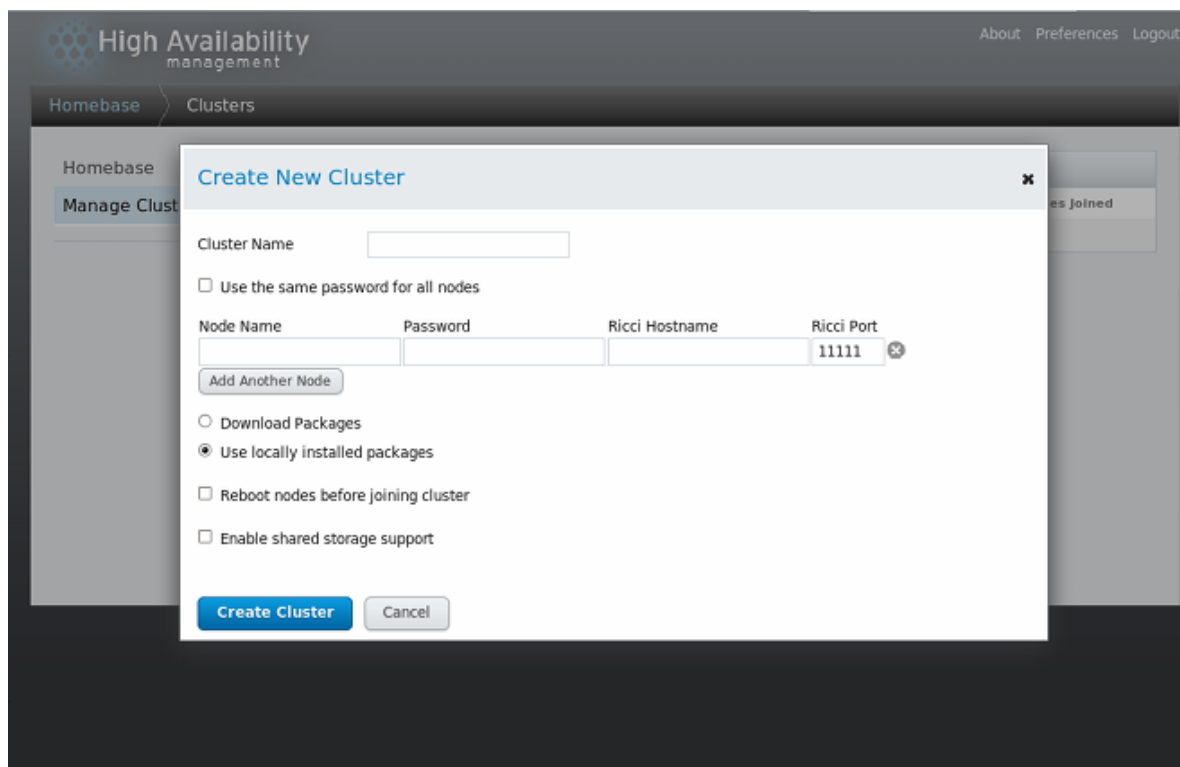


图 3.3. 创建 luci 集群对话框

3. 请根据需要在「创建新集群」页面中输入以下参数:

- 在「集群名称」文本框中输入集群名称。集群名称不能超过 15 个字符。
- 如果集群中的每个节点都有同样的 ricci 密码, 您可以选择「在所有节点中使用相同的密码」以便在您添加的节点中自动填写「密码」字段。
- 在「节点名称」栏中输入集群中节点的名称, 并在「密码」栏中为该节点输入 ricci 密码。
- 如果将您的系统配置为使用专门的私有网络用于集群流量, 则需要将 luci 配置为与 ricci 在一个不同于集群节点名称解析的地址进行沟通。您可以在「Ricci 主机名」中输入该地址达到此目的。
- 如果您要在 ricci 代理中使用不同的端口, 而不是默认的 11111 端口, 您可以更改那个参数。
- 点击「添加另一个节点」并输入节点名称, 同时为集群的每个附加节点输入 ricci 密码。
- 如果您不想要在创建集群时升级已经在节点中安装的集群软件包, 请必要选择「使用本地安装的软件包」选项。如果您要升级所有集群软件包, 请选择「下载软件包」选项。



注意

如果缺少任意基本集群组件 (cman、rgmanager、modcluster 及其所有相依性软件包), 无论是否选择「使用本地安装的软件包」或者「下载软件包」选项, 都会安装它们。如果没有安装它们, 则创建节点会失败。

- 需要时选择「加入集群前重启节点」。
 - 如果需要集群的存储，则请选择「启动共享存储支持」。这样做将下载支持集群存储的软件包并启用集群的 LVM。您应该只能在可访问弹性存储附加组件或者可扩展文件系统附加组件时选择这个选项。
4. 点击 创建集群。点击 创建集群 后会有以下动作：
- a. 如果您选择了「下载软件包」，则会在节点中下载并安装软件包。
 - b. 在节点中安装集群软件（或者确认安装了正确的软件包）。
 - c. 在集群的每个节点中更新并传播集群配置文件。
 - d. 加入该集群的添加的节点

显示的信息表示正在创建该集群。当集群准备好后，该显示会演示新创建集群的状态，如 图 3.4 “集群节点显示” 所示。请注意：如果没有在任何节点中运行 ricci，则该阶段创建会失败。

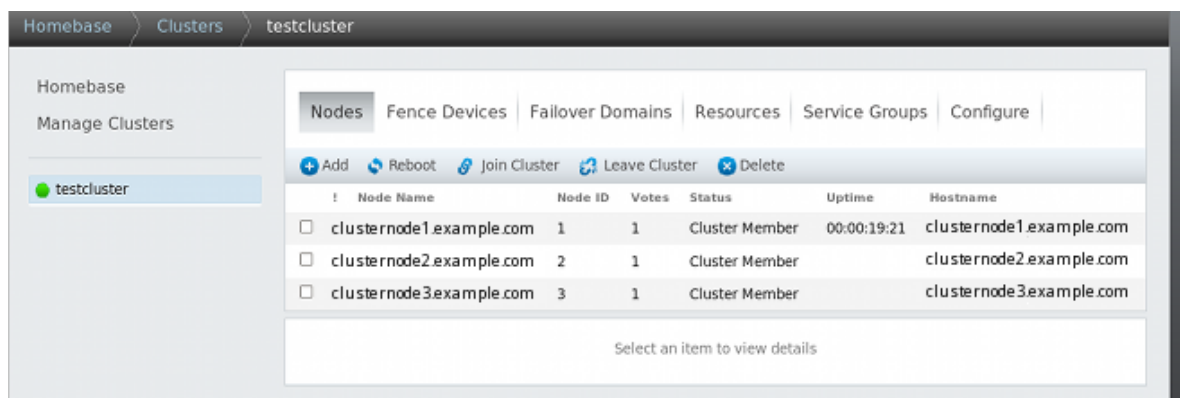


图 3.4. 集群节点显示

5. 点击 创建集群 按钮创建集群后，您仍可以通过点击集群节点显示页面上部菜单中的「添加」或者「删除」功能从集群中添加或者删除节点。除非您要删除整个集群，否则必须在删除节点前停止它们。有关从目前正在操作的现有集群中删除节点的详情请参考 第 4.2.4 节 “删除集群中的成员”。

3.4. 全局集群属性

当您选择一个集群进行配置时会显示具体集群页面。该页面提供配置集群范围属性的界面。您可以点击集群显示顶部的「配置」菜单配置集群范围属性。此时会出现分页界面，它可为您提供以下标签：「常规」、「Fence 守护进程」、「网络」以及「QDisk」和「日志」。要在那些标签中配置参数，请按照本节提供的步骤执行。如果您不需要在某个标签中配置参数，则请跳过该标签的步骤。

3.4.1. 配置常规属性

点击「常规」标签显示「常规属性」页面，该页面提供修改配置版本的界面。

- 「集群名称」文本框显示集群名称，它不接受更改集群名称。更改集群名称的唯一方法是创建有新名称的集群配置。
- 默认在创建集群时将「配置版本」值设定为 1，并在每次修改集群配置后自动增加该值。但如果您需要将其设定为其它值，您可以在「配置版本」文本框中指定该值。

如果您更改了「配置版本」值，请点击 应用 按钮以便此更改生效。

3.4.2. 配置 Fence 守护进程属性

点击「Fence 守护进程」标签显示「Fence 守护进程属性」页面，该页面提供配置「失败后延迟」和「加入后延迟」的界面。您为这些参数配置的值是集群的常规 fencing 属性。要为集群的节点配置具体 fence 设备，请使用集群显示中的「Fence 设备」菜单项，如第 3.5 节“配置 Fence 设备”所示。

- 「失败后延迟」参数为在节点失败之后，执行节点（fence 域中的成员）fencing 前 fence 守护进程（fenced）要等待的秒数。「失败后延迟」的默认值为 0。对不同的集群和网络性能需要可修改该值。
- 「加入后延迟」参数为在节点加入 fence 域之后，执行节点 fencing 前 fence 守护进程（fenced）要等待的秒数。「加入后延迟」的默认值为 3 通常将「加入后延迟」设定为 20-30 秒之间，但可根据集群和网络性能进行修改。

输入所需值并点击 应用 以便更改生效。

备注

有关「加入后延迟」和「失败后延迟」的详情请参考 fenced(8) man page。

3.4.3. 网络配置

点击「网络」标签显示「网络配置」页面，该页面提供配置网络传输类型的界面。

您可以使用这个标签选择以下选项之一：

- 「UDP 多播并让集群选择多播地址」

这是默认设置。如果您没有选择这个选项，则红帽高可用性附加组件软件会根据集群 ID 创建一个多播地址。它可生成地址的后 16 字节，并根据 IP 协议（IPv4 或者 IPv6）将其附加到前面的部分。

- 对于 IPv4 — 该地址格式为 239.192 加上红帽高可用性附加组件软件生成的后 16 字节。
- 对于 IPv6 — 该地址格式为 FF15:: 加上红帽高可用性附加组件软件生成的后 16 字节。

备注

集群 ID 是 cman 为每个集群生成的唯一设备符。要查看集群 ID，请在集群节点中运行 `cman_tool status` 命令。

- 「UDP 多播并手动指定多播地址」

如果您需要使用具体的多播地址，请选择这个选项并在文本框中输入多播地址。

如果您指定一个多播地址，您应该使用 cman 采用的 239.192.x.x 序列（IPv6 使用 FF15::）。否则，使用该范围外的多播地址可能导致无法预计的结果。例如：使用 224.0.0.x（“网络中的所有主机”）可能无法正常路由，有些硬件甚至根本无法路由。

**注意**

如果您指定了多播地址，请确定您检查了集群数据包所要经过的路由器。有些路由器需要较长的时间获得地址，这样会严重影响集群性能。

点击 应用。修改传输类型时，需要重启集群以便更改生效。

3.4.4. 仲裁磁盘配置

点击「QDisk」标签显示「仲裁磁盘配置」页面，该页面可在您需要使用仲裁磁盘时提供配置仲裁磁盘参数的界面。

**重要**

仲裁磁盘参数和试探法要根据具体环境和特殊要求而定。要了解如何使用仲裁磁盘以及试探法，请参考 `qdisk(5) man page`。如果您在理解和使用仲裁磁盘方面需要帮助，请联络授权的红帽支持代表。

默认启用「不使用仲裁磁盘」参数。如果您需要使用仲裁磁盘，请点击「使用仲裁磁盘」，输入仲裁磁盘参数，点击「应用」并重启该集群以便更改生效。

表 3.1 “仲裁磁盘参数” 描述了仲裁磁盘参数。

表 3.1. 仲裁磁盘参数

参数	描述
Specify physical device: By device label	指定 <code>mkqdisk</code> 程序生成的仲裁磁盘标签。如果使用该字段，则仲裁守护进程会读取 <code>/proc/partitions</code> 文件，并在每个找到的块设备中检查 <code>qdisk</code> 签名，与指定的标签进行对比。这在节点间使用不同仲裁设备名称时很有用。
Heuristics	「到程序的路径」 — 用来确定试探法是否可用的程序。它可以是任何可由 <code>/bin/sh -c</code> 执行的程序。返回值为 0 说明成功，返回其它任意值说明失败。需要这个字段。 「间隔」 — 调用试探法的频率（以秒为单位）。每次试探间的默认间隔为 2 秒。 「分数」 — 试探法的加权。请小心确定试探法分数。每次试探的默认分数为 1。 「TKO」 — 宣布这个试探法不可用前连续失败的次数。
Minimum total score	视节点为“活跃”所需的最小分数。如果忽略或者将其设定为 0，则使用模热功能 $\text{floor}((n+1)/2)$ ，其中 n 为试探法分数之和。「Minimum Score」值必须永远小于试探法分数之和，否则将无法使用仲裁磁盘。



备注

点击「仲裁磁盘配置」标签中的「应用」按钮将更改发布到每个集群节点的集群配置文件中（`/etc/cluster/cluster.conf`）。但如果要操作仲裁磁盘，您必须重启该集群（请参考第 4.3 节“启动、停止、刷新和删除集群”）。

3.4.5. 日志配置

点击「日志」标签显示「日志配置」页面，该页面提供配置日志设置的界面。

您可以为全局日志配置进行以下设置：

- 点击「记录 debugging 信息」可启用在日志文件中记录 debugging 信息。
- 点击「在 syslog 中记录信息」可启用在 syslog 中记录信息。您可以选择「syslog 信息工具」和「syslog 信息优先权」设置，表示会将所选级别以及更高级别中的信息发送到 syslog。
- 点击「在日志文件中记录信息」可启用在日志文件中记录信息。您可以指定日志文件的路径名。「日志文件信息优先权」设置表示会将所选级别以及更高级别中的信息写入日志文件。

您可以在「日志配置」页的底部选择一个守护进程为具体守护进程覆盖全局日志设置。选择守护进程后，您可以检查是否要为该具体守护进程记录 debugging 信息。您还可以为那个守护进程指定 syslog 和日志文件设置。

为日志配置点击 应用 以便更改生效。

3.5. 配置 Fence 设备

配置 fence 设备包括为集群创建、更新和删除 fence 设备。您可在集群中为节点配置 fencing 前必须在集群中配置 fence 设备。

创建 fence 设备包括选择 fence 设备类型以及为那个 fence 设备输入参数（例如：name、IP address、login 和 password）。更新 fence 设备包括选择现有 fence 设备并为那个 fence 设备更改参数。删除 fence 失败包括选择现有 fence 设备并删除它。

本小节提供以下任务的步骤：

- 创建 fence 设备 — 请参考第 3.5.1 节“创建 Fence 设备”。创建并命名 fence 设备后，您可以为集群中的每个节点配置 fence 设备，如第 3.6 节“为集群成员配置 Fencing”所述。
- 更新 fence 设备 — 请参考第 3.5.2 节“修改 Fence 设备”。
- 删除 fence 设备 — 请参考第 3.5.3 节“删除 Fence 设备”。

在具体集群页面中，您可以点击集群显示顶端的「Fence 设备」为那个集群配置 fence 设备。这样做可为集群显示 fence 设备，并显示 fence 设备配置菜单项：「添加」、「更新」和「删除」。这是以下小节中所有步骤描述的起点。



备注

如果是刚开始集群配置，则还没有创建 fence 设备，因此也没有显示任何 fence 设备。

图 3.5 “luci fence 设备配置页面” 演示在创建 fence 设备前的 fence 设备配置页面。

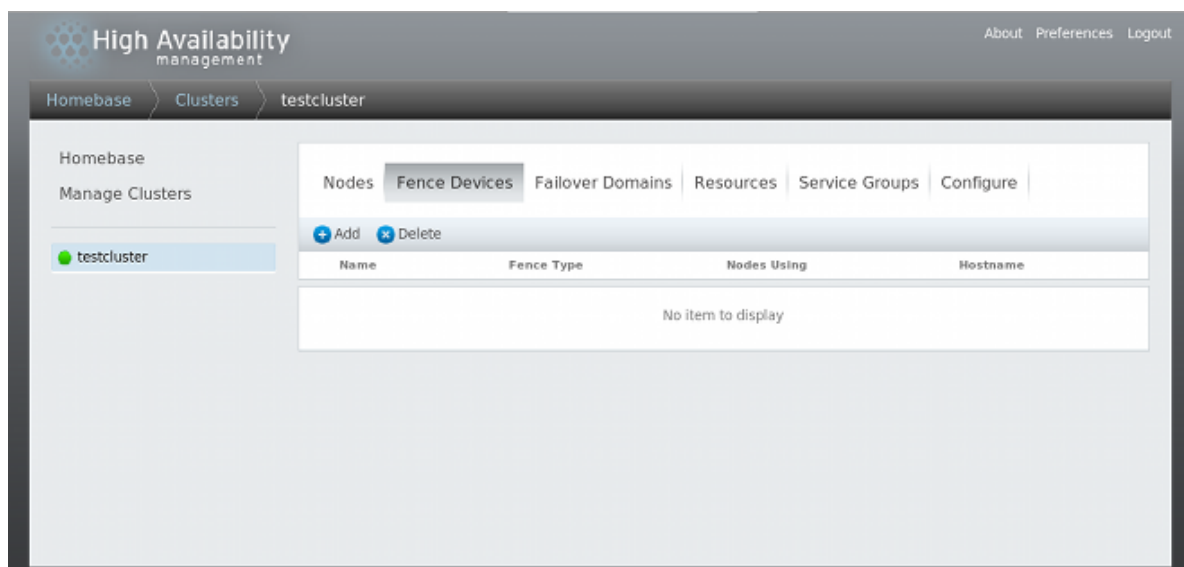


图 3.5. luci fence 设备配置页面

3.5.1. 创建 Fence 设备

要创建 fence 设备请按照以下步骤执行：

1. 在「Fence 设备」配置页面中，点击「添加」。点击「添加」显示「添加 Fence 设备（事务）」对话框。在这个对话框中选择要配置的 fence 设备类型。
2. 在「添加 Fence 设备（事务）」对话框中根据 fence 设备类型指定信息。有关 fence 设备参数详情请参考 [附录 A, Fence 设备参数](#)。在有些情况下您需要为该 fence 设备指定额外的节点具体参数，如 [第 3.6 节 “为集群成员配置 Fencing”](#) 所述。
3. 点击 提交。
4. 添加 fence 设备后，它会出现现在「Fence 设备」配置页面中。

3.5.2. 修改 Fence 设备

要修改 fence 设备，请按照以下步骤执行：

1. 在「Fence 设备」配置页面中点击要修改的 fence 设备名称。此时会出现那个 fence 设备的对话框，该对话框中应该有为该设备配置的值。
2. 要修改 fence 设备，修改显示的参数。有关详情请参考 [附录 A, Fence 设备参数](#)。
3. 点击 应用 并等待更新配置。

3.5.3. 删除 Fence 设备



注意

无法删除使用中的 fence 设备。要删除某个节点目前正在使用的 fence 设备，请首先为使用该设备的所有节点更新节点 fence 配置，然后删除该设备。

要删除 fence 设备，请按照以下步骤执行：

1. 在「Fence 设备」配置页面中选择 fence 设备左侧的复选框选择要删除的设备。
2. 点击 **删除** 并等待配置更新。此时会出现一条信息说明已经删除了该设备。
3. 当更新配置后，显示中不再会出现删除的 fence 设备。

3.6. 为集群成员配置 Fencing

您完成创建集群和创建集群 fence 设备的初始步骤后，您需要为集群节点配置 fencing。要在创建新集群后为节点配置 fencing 并为其配置 fence 设备，请按照本小节中的步骤执行。请注意：您必须为集群中的每个节点配置 fencing。

以下小节中提供了为节点配置单一 fence 设备、使用备份 fence 设备配置节点以及使用冗余电源配置节点的步骤：

- [第 3.6.1 节 “为节点配置单一 Fence 设备”](#)
- [第 3.6.2 节 “配置备份 Fence 设备”](#)
- [第 3.6.3 节 “配置使用冗余电源的节点”](#)

3.6.1. 为节点配置单一 Fence 设备

使用以下步骤配置有单一 fence 设备的节点。

1. 在具体集群页面中，您可以点击集群显示顶部的「节点」为集群中的节点配置 fencing。这样做会显示组成集群的节点。当您点击 `luci` 「Homebase」页面左侧菜单中的「管理集群」项下的集群名称时会出现这个默认页面。
2. 点击节点名称。点击节点链接会出现一个演示如何配置该节点的页面。

在具体节点页面中显示所有当前在该节点中运行的服务，同时还显示该节点所在故障切换域。您可以点击其名称修改现有故障切换域。有关配置故障切换域详情请参考 [第 3.7 节 “配置故障切换域”](#)。

3. 在具体节点页面中，请在「Fence 设备」下点击 **添加 fence 方法**。
4. 请输入为这个节点配置的 fencing 方法的「方法名」。这可以是红帽高可用性附加组件使用的任意名称。这与该设备的 DNS 名称不同。
5. 点击 **提交**。此时会显示具体节点页面，该页面中显示您刚刚在「Fence 设备」中添加的方法。
6. 点击 fence 方法下的 **添加 Fence 事务** 为这个方法配置 fence 事务。此时会出现「添加 Fence 设备（事务）」下拉菜单，您可从中选择您之前配置的 fence 设备，如 [第 3.5.1 节 “创建 Fence 设备”](#) 所述。

7. 为这个方法选择 fence 设备。如果这个 fence 设备需要您配置具体节点参数，则会显示要配置参数。有关 fencing 参数详情请参考 [附录 A, Fence 设备参数](#)。



注意

对于非电源 fence 方法（即 SAN/存储 fencing），会在具体节点参数显示中默认选择「Unfencing」。这可保证在重启该节点前不会重新启用被 fence 的节点对存储的访问。有关 unfencing 节点的详情请参考 `fence_node(8) man page`。

点击 提交。此时会返回显示 fence 方法和 fence 事务的具体节点页面。

3.6.2. 配置备份 Fence 设备

您可以为一个节点定义多个 fencing 方法。如果使用第一种方法对节点执行 fence 失败，系统会尝试使用第二种方法，之后是您配置的附加方法。

使用以下步骤为节点配置备份 fence 设备。

1. 使用 [第 3.6.1 节 “为节点配置单一 Fence 设备”](#) 所述步骤配置节点的主 fencing 方法。
2. 在您定义的主要方法下面点击 添加 Fence 方法。
3. 请您为这个节点配置的备份 fence 方法命名并点击 提交。此时会出现具体节点，该页面中显示您刚刚在主 fence 方法下添加的方法。
4. 点击 添加 Fence 事务 为这个方法配置 fence 事务。此时会出现下拉菜单，您可从中选择您之前配置的 fence 设备，如 [第 3.5.1 节 “创建 Fence 设备”](#) 所述。
5. 为这个方法选择 fence 设备。如果这个 fence 设备需要您配置具体节点参数，则会显示要配置参数。有关 fencing 参数详情请参考 [附录 A, Fence 设备参数](#)。

点击 提交。此时会返回显示 fence 方法和 fence 事务的具体节点页面。

您可以继续根据需要添加 fencing 方法。您可以点击「上移」和「下移」重新安排这个节点使用的 fencing 方法顺序。

3.6.3. 配置使用冗余电源的节点

如果将集群配置为在节点中使用冗余电源，您必须确定配置了 fencing，这样就可需要在对节点执行 fence 时将其完全关闭。如果您将每个电源配置为使用独立 fence 的方法，则会分别对每个电源执行 fence 操作。第二个电源可在第一个电源完全被 fence 后允许系统继续运行。要将系统配置为使用双电源，则必须配置您的 fence 设备以便关闭两个电源时可完全关闭系统。当将系统配置为使用 Conga 时要求您在单一 fencing 方法中配置两个事务。

要为有双电源的节点配置 fencing，请按照本小节中的步骤执行。

1. 在您为有冗余电源的节点中配置保护前，您必须将每个电源开关配置为集群的保护设备。有关配置 fence 设备的详情请参考 [第 3.5 节 “配置 Fence 设备”](#)。
2. 在具体集群页面中点击集群显示顶端的「节点」，此时会显示组成该集群的节点。这也是您点击 luci 「Homebase」页面左侧菜单中集群名称下的「管理集群」菜单时出现的默认显示此页面。
3. 点击节点名称。点击节点链接会出现一个演示如何配置该节点的页面。

4. 在具体节点页面中, 请点击 添加 Fence 方法。
5. 请输入您为这个节点配置的 fencing 方法名称。
6. 点击 提交。此时会显示具体节点页面, 该页面中显示您刚刚在「Fence 设备」中添加的方法。
7. 点击 添加 Fence 事务 将第一个电源供应配置为这个方法的 fence 事务。此时会显示一个下拉菜单, 您可从中选择您之前配置的电源 fencing 设备, 如 第 3.5.1 节 “创建 Fence 设备” 所示。
8. 为这个方法选择电源 fence 设备之一, 并为这个设备输入适当的参数。
9. 点击 提交。此时会返回显示 fence 方法和 fence 事务的具体节点页面。
10. 在您为第一个电源 fencing 设备配置的同一 fence 方法中点击 添加 Fence 事务。此时会出现一个下拉菜单, 您可从中选择您之前配置的第二个电源 fencing 设备, 如 第 3.5.1 节 “创建 Fence 设备” 所述。
11. 为这个方法选择第二电源 fence 设备, 并为这个设备输入适当的参数。
12. 点击 提交。此时会返回具体节点页面, 该页面中包括 fence 方法、显示的 fence 事务以及每个系统关闭和打开电源的顺序。如 图 3.6 “双电源 Fencing 配置” 所述。

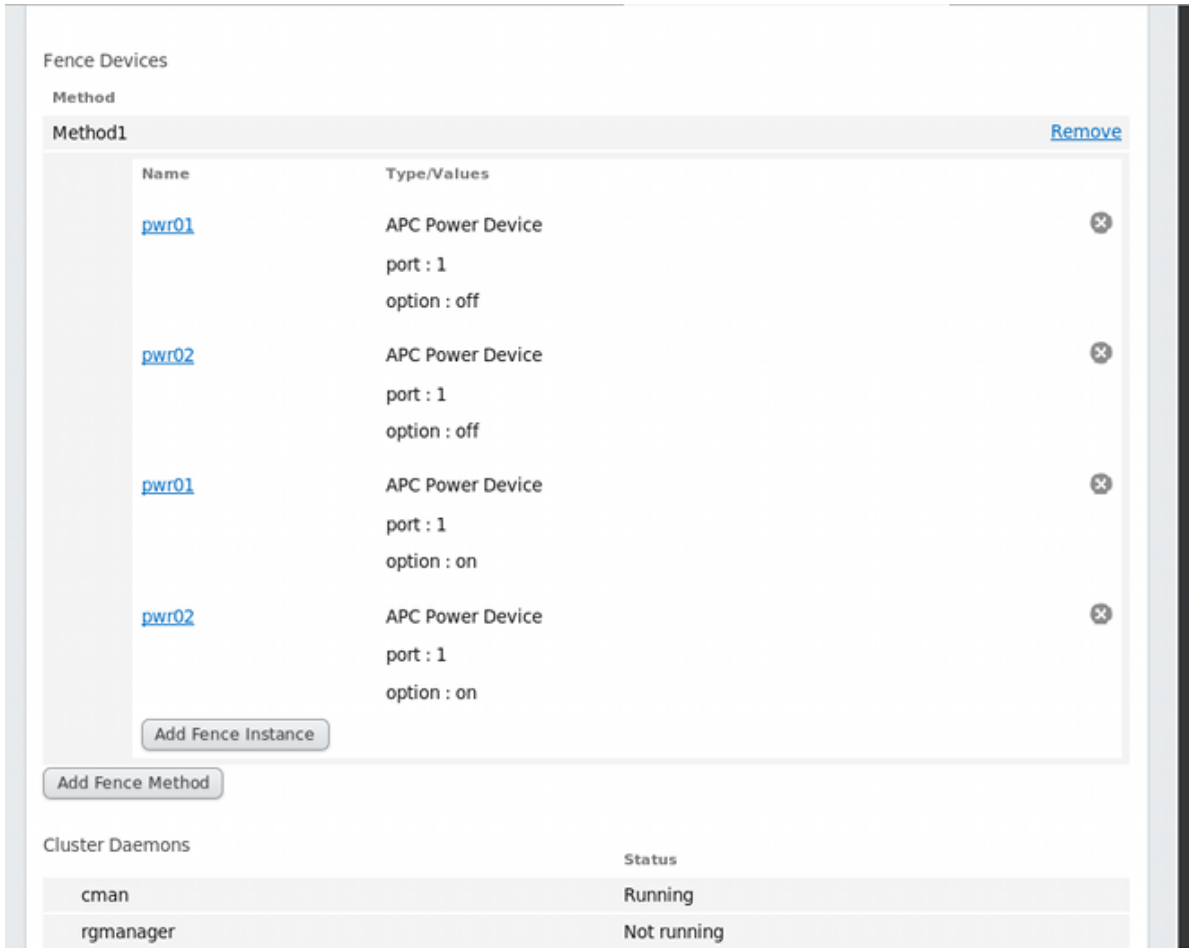


图 3.6. 双电源 Fencing 配置

3.7. 配置故障切换域

故障切换域是一个命名的集群节点子集，它可在节点失败事件中运行集群服务。故障切换域有以下特征：

- 无限制 — 允许您为在子集指定首选成员，但分配给这个域名的集群服务可在任意可用成员中运行。
- 限制 — 允许您限制可运行具体集群服务的成员。如果在限制故障切换域中没有可用成员，则无法启动集群服务（手动或者使用集群软件均不可行）。
- 无序 — 当将一个集群服务分配给一个无序故障切换域时，则可从可用故障切换域成员中随机选择运行集群服务的成员，没有优先顺序。
- 有序 — 允许您指定故障切换域成员的首选顺序。列表最顶端的成员为首选成员，接下来是列表中的第二个成员，以此类推。
- 故障恢复 — 允许您指定在故障切换域中的服务是否应该恢复到节点失败前最初运行的节点。配置这个特性在作为有序故障切换域一部分节点重复失败的环境中很有帮助。在那种情况下，如果某个节点是故障切换域中的首选节点，在可能在首选节点和其它节点间重复切换和恢复某个服务，从而不会对性能产生严重影响。



备注

故障恢复特性只适用于配置了有序故障切换的集群。



备注

更改故障切换域配置对目前运行中的服务无效。



备注

故障切换域不需要操作。

默认情况下故障切换域为无限制和无序的。

在由几个成员组成的集群中，使用限制故障切换域可最大程度降低设置集群以便运行集群服务的工作（比如 httpd），它要求您在运行该集群服务的所有成员中进行完全一致的配置。您不需要将整个集群设置为运行该集群服务，只要设置与该集群服务关联的限制故障切换域中的成员即可。

备注

要配置首选成员，您可以创建一个只由一个集群成员组成的无限故障切换域。这样可让该集群服务在那个主要集群成员（首选成员）中运行，但允许将集群服务切换到任意其它成员中。

以下小节描述了如何添加、修改和删除故障切换域：

- 第 3.7.1 节 “添加故障切换域”
- 第 3.7.2 节 “修改故障切换域”
- 第 3.7.3 节 “删除故障切换域”

3.7.1. 添加故障切换域

要添加故障切换域，请按照本小节中的步骤执行。

1. 在具体集群页面中，您可以点击集群显示顶部的「故障切换域」为那个集群配置故障切换域。此时会显示为这个集群配置的故障切换域。
2. 点击「添加」。点击「添加」时会显示「在集群中添加故障切换域」对话框，如 图 3.7 “luci 故障切换域配置对话框” 所示。

Add Failover Domain To Cluster

Name

☐

Prioritized

Order the nodes to which services failover.

☐

Restricted

Service can run only on nodes specified.

☐

No Failback

Do not send service back to 1st priority node when it becomes available again.

	Member	Priority
clusternode1.example.com	☐	
clusternode2.example.com	☐	
clusternode3.example.com	☐	

Create

Cancel

图 3.7. luci 故障切换域配置对话框

3. 在「在集群中添加故障切换域」对话框的「名称」文本框中指定故障切换域名称。

**注意**

该名称应该可以与集群中其它名称所显示的目的区别。

4. 要启用在故障切换域成员间设置故障切换优先权，请点击「优先的」复选框。选择「优先的」复选框后，您可以为选择作为故障切换域成员的每个节点设置优先值，「优先权」。
5. 要限制这个故障切换域成员的故障切换，请点击「有限」复选框。选择「有限」复选框后，分配给这个故障切换域的服务只能切换到这个故障切换域中的节点。
6. 要将那个节点指定为不在这个故障切换域中恢复，请点击「无故障恢复」复选框。选择「无故障恢复」后，如果从首选节点中恢复某个服务，则该服务不会切换到恢复它的节点中。
7. 配置这个故障切换域的成员。为每个要成为故障切换域成员的节点点击「成员」复选框。如果选择「优先的」复选框，则请为故障切换域每个成员在「优先权」文本框中设置优先权。
8. 点击 创建。此时会显示新创建故障切换域的「故障切换域」页面。出现一条信息显示创建了新的域。刷新该页面查看更新的状态。

3.7.2. 修改故障切换域

要修改故障切换域请按照本小节中的步骤执行。

1. 在具体集群页面中，您可以点击集群显示顶部的「故障切换域」为那个集群配置故障切换域。此时会显示为这个集群配置的故障切换域。
2. 点击故障切换域名称，此时会显示那个故障切换域的配置页面。
3. 要修改该故障切换域的「优先」、「有限」或者「无故障切换恢复」属性，请选择或者取消该属性旁的复选框，并点击 更新属性。
4. 要修改故障切换域成员，请选择或者取消集群成员旁的复选框。如果该故障切换域是有先的，您还可以为集群成员修改优先权设置。点击 更新设置。

3.7.3. 删除故障切换域

要删除故障切换域，请按照本小节中的步骤操作。

1. 在具体集群页面中，您可以点击集群显示顶部的「故障切换域」为那个集群配置故障切换域。此时会显示为这个集群配置的故障切换域。
2. 选择要删除的故障切换域前的复选框。
3. 点击「删除」。

3.8. 配置全局集群资源

您可以配置在集群中运行的任意服务所使用的全局资源，还可以配置只可用于具体服务的资源。

要添加全局集群资源，请按照本小节中的步骤操作。您可添加在配置服务时是属于本地资源的资源，如 [第 3.9 节 “在集群中添加集群服务”](#) 所述。

1. 在具体集群页面中，您可点击集群显示顶部的「资源」菜单在那个集群中添加资源。此时会显示已经为那个集群添加的资源。
2. 点击「添加」。此时会显示「在集群中添加资源」下拉菜单。
3. 点击「在集群中添加资源」中的下拉框并选择要配置的资源类型。
4. 输入您要添加资源的资源参数。资源参数请参考 [附录 B, HA 资源参数](#)。
5. 点击 提交。点击 提交 会返回显示「资源」显示的资源页面，此时该页面会显示添加的资源（和其它资源）。

要修改现有资源，请执行以下步骤。

1. 在 luci 「资源」页面中点击要修改的资源名称。此时会显示那个资源的参数。
2. 编辑该资源的参数。
3. 点击 应用。

要删除现有资源，请执行以下步骤。

1. 在 luci 「资源」页面中选择所有要删除资源。
2. 点击「删除」。

3.9. 在集群中添加集群服务

要在集群中添加集群服务，请按照本小节中的步骤执行。

1. 在具体集群页面中您可以点击集群显示顶部的「服务组」菜单在那个集群中添加服务。此时会显示已经为那个集群配置的服务。（在「服务」页面中，您还可以启动、重启以及禁用服务，如 [第 4.4 节 “管理高可用性服务”](#) 所述。）
2. 点击「添加」。此时会显示「在集群中添加服务」对话框。
3. 在「在集群中添加服务」对话框的「服务名称」文本框中输入该服务名称。



备注

请使用可明确与集群中的其它服务区别开来的描述性名称。

4. 如果您想在启动并运行集群时自动启动该服务，请选择「自动启动这个服务」复选框。如果没有选择这个复选框，则您必须在集群不处于停止状态时手动启动该服务。
5. 选择「独占运行」复选框设置策略，即该服务只在没有其它服务运行的节点中运行。
6. 如果您已经为该集群配置了故障切换域，您可以使用「故障切换域」参数的下拉菜单为该服务选择故障切换域。有关故障切换域的详情请参考 [第 3.7 节 “配置故障切换域”](#)。
7. 使用「恢复策略」下拉框为该服务选择恢复策略。选项包括 Relocate、Restart、Restart-Disable 或者 Disable 该服务。

选择 `Restart` 选项表示在重新定位该服务前系统应尝试重启失败的服务。选择 `Restart-Disable` 选项表示系统应在该服务失败的位置尝试重启该服务，但如果重启失败，则将禁用服务而不是移动到集群的其它主机中。

如果您选择 `Restart` 或者 `Restart-Disable` 作为该服务的恢复策略，您可以指定重新定位或者禁用该服务前最多重启失败的次数，您还可以在多少秒后不再重启。

8. 要在服务中添加资源，请点击 `添加资源`。点击 `添加资源` 会显示一个 `在服务中添加资源` 下拉菜单，您可从中选择要添加的现有全局资源，或者添加一个只可用于这个服务的新资源。

- 要添加现有全局资源，请在 `在服务中添加资源` 下拉框中点击现有资源名称。此时会显示在您所配置服务的「服务组」页面中的资源及其参数。有关添加或者修改全局资源的详情请参考 [第 3.8 节 “配置全局集群资源”](#)。
- 要添加只可用于这个服务的新资源，请在 `添加资源` 下拉框中选择要配置的资源类型并为您要添加的资源输入资源参数。有关资源参数请参考 [附录 B, HA 资源参数](#)。
- 当在服务中添加资源时，无论它是现有全局资源，还是只可用于这个服务的资源，您可将该资源指定为「独立子树」或者「非关键资源」。

如果您将资源指定为独立子树，那么如果该资源失败，则在系统尝试常规恢复前只会重启那个资源（而不是整个服务）。您可以指定在该节点中为该服务使用恢复策略前最多尝试重启该资源的次数。您还可以指定在多少秒后系统将为该服务使用恢复策略。

如果您将该资源指定为非关键资源，那么如果那个资源失败，则只需要重启该资源。同时如果该资源仍失败，那么只会禁用那个资源而不是整个服务。您可以指定在该节点中禁用该资源前最多重启该资源的次数。您还可以指定在多少秒后系统将禁用该资源。

9. 如果您要在您定义的资源中添加子资源，请点击 `添加子资源`。点击 `添加子资源` 后会显示「在服务中添加资源」下拉框，您可从中添加现有全局资源或者添加只可用于这个服务的新资源。您可以继续为这个资源添加子资源以适应您的要求。



备注

如果您要添加 Samba 服务资源，请将 Samba 服务资源直接连接到该服务，而不是服务中的资源。

10. 当您完成为该服务添加资源，并完成为资源添加子资源时，点击 `提交`。点击 `提交` 后会返回显示添加的服务（以及其它服务）的「服务组」页面。



备注

要确认在集群服务中使用现有的 IP 服务资源，您必须在集群节点中使用 `/sbin/ip addr list` 命令。以下为在运行集群服务的节点中执行 `/sbin/ip addr list` 命令的输出结果：

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
        valid_lft forever preferred_lft forever
```

要修改现有服务，请执行以下步骤。

1. 在 `luci` 「服务组」页面中点击要修改的服务名称。此时会显示您已经为那个服务配置的参数和资源。
2. 编辑该服务的参数。
3. 点击 提交。

要删除现有资源，请执行以下步骤。

1. 在 `luci` 「服务组」页面中选择要删除的所有服务。
2. 点击「删除」。

使用 Conga 管理 Red Hat 高可用性附加组件

本章论述了管理 Red Hat 高可用性附加组件的各种管理任务，它由以下小节组成：

- [第 4.1 节 “在 luci 界面中添加现有集群”](#)
- [第 4.2 节 “管理集群节点”](#)
- [第 4.3 节 “启动、停止、刷新和删除集群”](#)
- [第 4.4 节 “管理高可用性服务”](#)

4.1. 在 luci 界面中添加现有集群

如果您之前已创建了高可用性附加组件集群，则您可以在 luci 界面中轻松添加该集群以便使用 Conga 管理该集群。

要在 luci 界面中添加现有集群，请执行以下步骤：

1. 在 luci 「Homebase」页面左侧的菜单中点击「管理」。此时会出现「集群」页面。
2. 点击「添加」。此时会出现「添加现有集群」页面。
3. 在现有集群的所有节点中输入节点主机名和 ricci 密码。因为该集群的每个节点中都包含该集群的所有配置信息，这样就应该提供了在 luci 界面中添加集群的足够信息。
4. 点击 连接。此时「添加现有集群」页面会显示集群名称以及该集群中的其它节点。
5. 为该集群的每个节点输入独立的 ricci 密码，或者输入一个密码并选择「在所有节点中使用同一密码」。
6. 点击 添加集群。此时会在「管理集群」页面中出现之前配置的集群。

4.2. 管理集群节点

这部分记述了如何使用 Conga 的 luci 服务器组件执行以下节点管理功能：

- [第 4.2.1 节 “重启集群节点”](#)
- [第 4.2.2 节 “使节点离开或者加入集群”](#)
- [第 4.2.3 节 “在运行的集群中添加成员”](#)
- [第 4.2.4 节 “删除集群中的成员”](#)

4.2.1. 重启集群节点

要在集群中重启节点，请执行以下步骤：

1. 在具体集群页面中点击集群显示顶端的「节点」，此时会显示组成该集群的节点。这也是您点击 luci 「Homebase」页面左侧菜单中集群名称下的「管理集群」时出现的默认页面。
2. 点击要重启的节点旁的复选框选择该节点。

3. 选择该页面顶端菜单中的「重启」功能。此时会重启选择的节点并在该页面顶部出现一条信息表示正在重启该节点。
4. 刷新该页面查看该节点更新的状态。

您还可以在点击「重启」前选择所有要重启的节点，这样可以一次重启多个节点。

4.2.2. 使节点离开或者加入集群

您可以使用 Conga 的 luci 服务器组件，通过停止该节点中的所有集群服务让该节点离开集群。您还可以使用 Conga 的 luci 服务器组件让已经离开的节点重新加入该集群。

让集群离开节点并不会删除集群中该节点的配置信息，且该节点仍会出现在该集群节点显示中，只是状态为不是集群成员。有关从集群配置中完全删除节点的信息请参考 [第 4.2.4 节 “删除集群中的成员”](#)。

要让节点离开集群，请执行以下步骤，这样可关闭该节点中的集群软件。让节点离开集群可防止在重启时该节点自动加入集群。

1. 在具体集群页面中点击集群显示顶端的「节点」，此时会显示组成该集群的节点。这也是您点击 luci 「Homebase」页面左侧菜单中集群名称下的「管理集群」时出现的默认页面。
2. 点击节点旁的复选框选择您想要使其离开集群的节点。
3. 在该页面顶部菜单中选择「离开集群」功能，此时会在页面顶部出现一条信息表明已经停止该节点。
4. 刷新该页面查看该节点更新的状态。

您还可以在点击「离开节点」前选择所有您想要使其离开集群的节点，这样可一次让多个节点离开。

要让节点重新加入集群，请那些节点旁的复选框选择您想要重新加入到集群的节点并选择「加入集群」。这样可让选择的节点加入集群，并使其在重启时加入集群。

4.2.3. 在运行的集群中添加成员

要在运行的集群中添加成员，请按照本小节中的步骤执行。

1. 在具体集群页面中点击集群显示顶部的「节点」，此时会显示组成该集群的节点。这也是您点击 luci 「Homebase」页面左侧菜单中集群名称下的「管理集群」菜单时默认出现的页面。
2. 点击「添加」。点击「添加」会显示「在集群中添加节点」对话框。
3. 在「节点主机名」文本框中输入节点的名称，在「密码」文本框中输入 ricci 密码。如果您要在 ricci 代理中使用不同于默认 11111 的端口，您可以更改那个参数。
4. 如果需要集群的存储，请选择「启动共享存储支持」复选框下载支持集群存储的软件包并启用集群的 LVM。您应该只在可访问弹性存储附加组件或者可扩展文件系统附加组件时选择这个选项。
5. 如果您要添加多个节点，请点击「添加另一个节点」并为每个附加节点输入节点名称和密码。
6. 点击 添加节点。点击 添加节点 会导致以下动作：
 - a. 如果您选择「下载软件包」，则会在节点中下载集群软件包。
 - b. 在节点中安装集群软件（或者确认安装了正确的软件软件包）。

- c. 更新集群配置并在集群的每个节点中使用更新的集群配置 — 包括添加的节点。
- d. 添加的节点加入集群。

「节点」页面出现一条信息表示正在该集群中添加节点。刷新该页面更新状态。

7. 当添加节点进程完成后，点击新添加的节点名称为这个节点配置 fencing，如 [第 3.5 节 “配置 Fence 设备”](#) 所述。

4.2.4. 删除集群中的成员

要从目前处于操作状态的现有集群中删除成员，请按照本小节中的步骤执行。请注意：必须在删除节点前停止它们，除非您要同时删除该集群中的所有节点。

1. 在具体集群页面中点击集群显示顶部的「节点」，此时会显示组成该集群的节点。这也是您点击 luci 「Homebase」页面左侧菜单中集群名称下的「管理集群」菜单时默认出现的页面。



备注

要在删除节点时让所有在该节点中运行的服务执行故障切换，请跳过下一步。

2. 禁用或者重新定位要删除节点中运行的所有服务。有关禁用和重新定位服务的详情请参考 [第 4.4 节 “管理高可用性服务”](#)。
3. 选择要删除的节点。
4. 点击「删除」。「节点」页面显示正在删除该节点。刷新该页面查看当前状态。

4.3. 启动、停止、刷新和删除集群

您可以通过在集群的每个节点中执行以下动作启动、停止、重启某个集群。在具体集群页面中点击集群显示顶部的「节点」，此时会显示组成该集群的节点。

要停止集群，请执行以下步骤。这样会关闭节点中的集群软件，但不会删除节点中的集群配置信息，且该节点仍会出现在该集群节点显示中，只是状态为不是集群成员。

1. 点击每个集群旁的复选框选择集群中的所有节点。
2. 在该页面顶部的菜单中选择「离开集群」，此时会在页面顶部出现一条信息表示正在停止每个节点。
3. 刷新该页面查看节点更新的状态。

要启动集群，请执行以下步骤：

1. 点击每个集群旁的复选框选择集群中的所有节点。
2. 在该页面顶部的菜单中选择「加入集群」功能。
3. 刷新该页面查看节点更新的状态。

要重启运行中的集群，首先请停止集群中的所有节点，然后启动集群中的所有节点，如上所述。

要从 `luci` 界面中整个删除集群请执行以下步骤。这样会删除节点中的集群配置信息，并从集群显示中删除这些节点。



重要

删除集群是一个破坏操作，且无法取消。要在删除集群后进行恢复要求您从头开始重新创建并重新定义该集群。

1. 点击每个集群旁的复选框选择集群中的所有节点。
2. 在该页面顶部的菜单中选择「删除」功能。

4.4. 管理高可用性服务

除在 [第 3.9 节 “在集群中添加集群服务”](#) 中所述的添加和修改服务外，您还可以使用 Conga 的 `luci` 服务器组件为高可用性服务执行以下管理功能：

- 启动服务
- 重启服务
- 禁用服务
- 删除服务
- 重新定位服务

在具体集群页面中，您可以点击集群显示顶部的「服务组」为集群管理服务。此时会显示为该集群配置的服务。

- 启动服务 — 要启动任何当前没有运行的服务，请点击该服务旁的复选框选择您要启动的所有服务并点击「启动」。
- 重启服务 — 要重启任何当前运行的服务，请点击该服务旁的复选框选择您要启动的所有服务并点击「重启」。
- 「禁用服务」 — 要禁用任何当前运行的服务，请点击该服务旁的复选框选择您要启动的所有服务并点击「禁用」。
- 「删除服务」 — 要删除任何当前运行的服务，请点击该服务旁的复选框选择您要启动的所有服务并点击「删除」。
- 「重新定位服务」 — 要重新定位运行的服务，请在服务显示中点击该服务的名称。此时会显示该服务配置页面，并显示该服务目前在哪个节点中运行。

在「在节点中启动.....」下拉框中选择您想要将服务重新定位的节点，并点击「启动」图标。此时会在页面顶部显示一条信息说明正在重启该服务。您可以刷新该页面查看新显示，在该显示中说明该服务正在您选择的节点中运行。



注意

您还可以点击「服务」页面中的服务名称启动、重启、禁用或者删除独立服务。此时会显示服务配置页面。在服务配置页面右上角有一些图标:「启动」、「重启」、「禁用」和「删除」。

使用 ccs 命令配置红帽高可用性附加组件

从红帽企业版 Linux 6.1 发行本开始，红帽高可用性附加组件支持 ccs 集群配置命令。ccs 命令可让管理员创建、修改和查看 cluster.conf 集群配置文件。您可以使用 ccs 命令在本地文件系统或者远程节点中配置集群配置文件。管理员还可以使用 ccs 命令在配置的集群的一个或者全部节点中启动或者停止集群服务。

本章论述了如何使用 ccs 命令配置红帽高可用性附加组件集群配置文件。有关使用 ccs 命令管理运行的集群的详情请参考 第 6 章 使用 ccs 管理 Red Hat 高可用性附加组件。

本章由以下小节组成：

- 第 5.1 节 “操作概述”
- 第 5.2 节 “配置任务”
- 第 5.3 节 “启动 ricci”
- 第 5.4 节 “创建集群”
- 第 5.5 节 “配置 Fence 设备”
- 第 5.6 节 “为集群成员配置 Fencing”
- 第 5.7 节 “配置故障切换域”
- 第 5.8 节 “配置全局集群资源”
- 第 5.9 节 “在集群中添加集群服务”
- 第 5.10 节 “配置仲裁磁盘”
- 第 5.11 节 “其它集群配置”
- 第 5.12 节 “在集群节点中传播配置文件”



重要

请确定您部署的高可用性附加组件符合您的要求，且可被支持。请在部署前咨询授权红帽代表确认您的配置。另外还要预留一定的时间进行配置的失败模型测试。



重要

本章通常使用 cluster.conf 元素和属性作为参考。有关 cluster.conf 元素和属性的完整列表，请参考 /usr/share/cluster/cluster.rng 中的集群方案以及 /usr/share/doc/cman-X.Y.ZZ/cluster_conf.html 中有注释的方案（例如：/usr/share/doc/cman-3.0.12/cluster_conf.html）。

5.1. 操作概述

本小节论述了以下使用 ccs 命令配置集群的常规操作：

- [第 5.1.1 节 “在本地系统中创建集群配置文件”](#)
- [第 5.1.2 节 “查看当前集群配置”](#)
- [第 5.1.3 节 “使用 ccs 命令指定 ricci 密码”](#)
- [第 5.1.4 节 “修改集群配置组件”](#)

5.1.1. 在本地系统中创建集群配置文件

使用 ccs 命令，您可以在集群节点中创建集群配置文件，或者您可以在本地文件系统中创建集群配置文件然后将其发送到集群的主机中。这可让您在本地机器中完成文件，使用版本控制进行维护或者根据您的需要标记该文件。使用 ccs 命令不需要 root 特权。

当您在集群节点中使用 ccs 命令创建并编辑集群配置文件时，请使用 -h 选项指定主机名。这样就可在该主机中创建并编辑 cluster.conf 文件：

```
ccs -h host [options]
```

要在本地系统中创建并编辑集群配置文件，请使用 ccs 命令的 -f 选项指定您要执行集群操作的配置文件名称。您可以使用任意名称命名该文件。

```
ccs -f file [options]
```

您在本地创建该文件后，可使用 ccs 命令的 --setconf 选项将其发送到集群节点中。在集群的一台主机中会将您发送的文件命名为 cluster.conf，会将其保存在 /etc/cluster 目录中。

```
ccs -h host -f file --setconf
```

有关使用 ccs 命令的 --setconf 选项的详情请参考 [第 5.12 节 “在集群节点中传播配置文件”](#)。

5.1.2. 查看当前集群配置

本章论述了如何创建集群配置文件。您可以使用以下命令指定该集群中的一个节点作为主机，随时为集群输出当前文件：

```
ccs -h host --getconf
```

如果您在本地系统中创建了您的配置文件，您可以使用 -f 选项而不是 -h 选项，如 [第 5.1.1 节 “在本地系统中创建集群配置文件”](#) 所述。

5.1.3. 使用 ccs 命令指定 ricci 密码

执行 ccs 命令在集群的节点中发布 cluster.conf 文件的副本需要在该集群节点中安装并运行 ricci，如 [第 2.11 节 “ricci 注意事项”](#) 所述。使用 ricci 时，第一次在任意具体机器中与 ricci 互动时要求输入密码。

如果您还没有在您要使用的具体机器中为 ricci 事务输入密码，则会在 ccs 命令需要密码时要求您输入。另外，您可以使用 -p 选项在该命令中指定 ricci 密码。


```
ccs -h host -p password --sync --activate
```

当您在集群的所有节点中使用 `ccs` 命令的 `--sync` 选项传播 `cluster.conf` 文件，并为该命令指定 `ricci` 密码时，`ccs` 命令将在该集群的每个节点中使用那个密码。如果您需要为 `ricci` 在独立节点中设定不同密码，您可以使用 `--setconf` 命令的 `-p` 选项每次在一个节点中发布配置文件。

5.1.4. 修改集群配置组件

您使用 `ccs` 命令配置集群配置文件中的集群组件及其属性。您在该文件中添加集群组件后，要修改那个组件的属性，您必须删除您定义的组件并使用修改的属性再次添加该组件。本章的各个小节提供了如何对每个组件进行此操作的信息。

`cman` 集群组件的属性提供修改集群组件过程的一个例外选择。要修改这些属性，您可执行 `ccs` 命令的 `--setcman` 选项，执行新属性。

5.2. 配置任务

使用 `ccs` 配置红帽高可用性附加组件软件包括以下步骤：

1. 确定在该集群的所有节点中运行 `ricci`，请参考 [第 5.3 节 “启动 ricci”](#)。
2. 创建集群。请参考 [第 5.4 节 “创建集群”](#)。
3. 配置 fence 设备。请参考 [第 5.5 节 “配置 Fence 设备”](#)。
4. 为集群成员配置 fencing。请参考 [第 5.6 节 “为集群成员配置 Fencing”](#)。
5. 创建故障切换域。请参考 [第 5.7 节 “配置故障切换域”](#)。
6. 创建资源。请参考 [第 5.8 节 “配置全局集群资源”](#)。
7. 创建集群服务。请参考 [第 5.9 节 “在集群中添加集群服务”](#)。
8. 需要时配置仲裁磁盘。请参考 [第 5.10 节 “配置仲裁磁盘”](#)。
9. 配置全局集群属性。请参考 [第 5.11 节 “其它集群配置”](#)。
10. 将该集群配置文件传播到所有集群节点中。请参考 [第 5.12 节 “在集群节点中传播配置文件”](#)。

5.3. 启动 ricci

要在该集群的节点中创建并部署集群配置文件，必须在每个节点中运行 `ricci` 服务。在启动 `ricci` 前，您应该确定您将系统配置如下：

1. 应在您的集群节点中为 `ricci` 启用 IP 断口。有关在集群节点中启用 IP 端口的详情请参考 [第 2.3.1 节 “在集群节点中启用 IP 端口”](#)。
2. 在该集群的所有节点中安装 `ricci` 服务，并分配 `ricci` 密码，如 [第 2.11 节 “ricci 注意事项”](#) 所示。

在每个节点中安装并配置 `ricci` 后，在每个节点中启动 `ricci` 服务：

```
# service ricci start
Starting ricci: [ OK ]
```

5.4. 创建集群

本小节论述了如何使用 ccs 命令，在没有 fencing、故障切换域和 HA 服务的情况下创建、修改并删除集群配置框架。随后的小节论述了如何配置该配置文件的那些部分。

要创建集群配置文件框架，请首先创建并命名该集群，然后在该集群中添加节点，如下所示：

1. 在该集群中的节点中执行 ccs 命令创建集群配置文件，使用 -h 参数指定创建该文件的节点，并使用 createcluster 选项指定该集群名称：

```
ccs -h host --createcluster clustername
```

例如：以下命令在 node-01.example.com 中创建了名为 mycluster 的配置文件：

```
ccs -h node-01.example.com --createcluster mycluster
```

集群名称不能超过 15 个字符。

如果该主机中已经存在 cluster.conf 文件，执行这个命令将替换现有文件。

如果您要在本地系统中创建集群配置文件，您可以指定 -f 选项，而不是 -h 选项。有关在本地创建该文件的详情请参考 [第 5.1.1 节 “在本地系统中创建集群配置文件”](#)。

2. 要配置该集群包含的节点，请在该集群的每个节点中执行以下命令：

```
ccs -h host --addnode node
```

例如：以下三个命令可在 node-01.example.com 的配置文件中添加 node-01.example.com、node-02.example.com 和 node-03.example.com。

```
ccs -h node-01.example.com --addnode node-01.example.com
ccs -h node-01.example.com --addnode node-02.example.com
ccs -h node-01.example.com --addnode node-03.example.com
```

要查看集群中已经配置的节点列表，请执行以下命令：

```
ccs -h host --lsnodes
```

例 5.1 “添加了三个节点后的 cluster.conf 文件” 演示了您创建的包含节点 node-01.example.com、node-02.example.com 和 node-03.example.com 的集群 mycluster 的 cluster.conf 配置文件。

例 5.1. 添加了三个节点后的 cluster.conf 文件

```
<cluster name="mycluster" config_version="2">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
```

```
<clusternode name="node-02.example.com" nodeid="2">
  <fence>
  </fence>
</clusternode>
<clusternode name="node-03.example.com" nodeid="3">
  <fence>
  </fence>
</clusternode>
</clusternodes>
<fencedevices>
</fencedevices>
<rm>
</rm>
</cluster>
```

当您在该集群中添加节点时，您可以该节点贡献的票数以便决定是否达到法定票数。请使用以下命令为集群节点设定票数：

```
ccs -h host --addnode host --votes votes
```

添加节点时，ccs 会为该节点分配一个唯一证书作为该节点的识别符。如果您要在创建节点时手动指定节点识别符，请使用以下命令：

```
ccs -h host --addnode host --nodeide nodeid
```

要从集群中删除节点，请执行以下命令：

```
ccs -h host --rmnode node
```

您完成配置集群的所有组件后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.5. 配置 Fence 设备

配置 fence 设备包括为该集群创建、更新和删除 fence 设备。您在为集群的节点配置 fencing 前必须在集群中创建并命名 fence 设备。有关在集群的独立节点中配置 fencing 的详情请参考 [第 5.6 节 “为集群成员配置 Fencing”](#)。

在配置您的 fence 设备前，您可能想要为您的系统修改 fence 守护进程属性，替换默认值。您为 fence 守护进程配置的值应该是集群的常规值。您要为集群修改的常规 fencing 属性如下：

- `post_fail_delay` 属性是在节点失败后，fencing 节点前 fence 守护进程（fenced）要等待的秒数。• `post_fail_delay` 的默认值为 0。可使用不同的数值以适应集群和网络性能。

请执行以下命令为 `post_fail_delay` 属性分配数值：

```
ccs -h host --setfencedaemon post_fail_delay=value
```

- `post-join_delay` 属性是节点加入 fence 域后，fencing 节点前 fence 守护进程（fenced）等待的秒数。• `post-join_delay` 默认值为 3。通常将 `post-join_delay` 设定为 20 - 30 秒之间，但根据集群和网络性能有所不同。

请执行以下命令为 `post_join_delay` 属性分配数值：

```
ccs -h host --setfencedaemon post_join_delay=value
```



注意

有关 `post_join_delay` 和 `post_fail_delay` 属性以及您可以修改的附加 fence 守护进程的详情请参考 `fenced(8)` man page, `/usr/share/cluster/cluster.rng` 中的集群方案以及 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中注释的方案。

请执行以下命令为集群配置 fence 设备：

```
ccs -h host --addfencedev devicename [fencedeviceoptions]
```

例如：执行以下命令在集群节点 `node1` 中配置一个 `apc` fence 设备，名为 `myfence`，IP 地址为 `apc_ip_example`，登录为 `login_example`，密码为 `password_example`：

```
ccs -h node1 --addfencedev myfence agent=fence_apc ipaddr=apc_ip_example login=login_example  
passwd=password_example
```

下面的示例演示了您添加这个 `apc` fence 设备后 `cluster.conf` 配置文件的 `fencedevices` 部分：

```
<fencedevices>  
  <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"  
    passwd="password_example"/>  
</fencedevices>
```

请执行以下命令输出当前为您的集群配置的 fence 设备列表：

```
ccs -h host --lsfencedev
```

请执行以下命令从您的集群配置中删除 fence 设备：

```
ccs -h host --rmfencedev fence device name
```

例如：执行以下命令从集群节点 `node1` 的集群配置文件中删除名为 `myfence` 的 fence 设备：

```
ccs -h node1 --rmfencedev myfence
```

如果您要修改已经配置的 fence 设备的属性，您必须首先删除那个 fence 设备，然后使用修改的属性再次添加该设备。

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.6. 为集群成员配置 Fencing

您完成创建集群以及创建 fence 设备的初始步骤后，您需要为集群节点配置 fencing。请按照本小节提供的步骤在创建新集群并为该集群配置 fencing 设备后为节点配置 fencing。请注意：您必须为该集群中的每个节点配置 fencing。

本小节记录了以下步骤：

- [第 5.6.1 节 “为节点配置使用单一电源的 Fence 设备”](#)
- [第 5.6.2 节 “为节点配置单一存储 Fence 设备”](#)
- [第 5.6.3 节 “配置备用 Fence 设备”](#)
- [第 5.6.4 节 “配置使用冗余电源的节点”](#)
- [第 5.6.5 节 “删除 Fence 方法和 Fence 事务”](#)

5.6.1. 为节点配置使用单一电源的 Fence 设备

请使用以下步骤配置使用单一电源 fence 设备的节点，fence 设备名为 apc，使用 fence_apc fencing 代理。

1. 在该节点中添加 fence 方法，并为该 fence 方法提供名称。

```
ccs -h host --addmethod method node
```

例如：执行以下命令在集群节点 node-01.example.com 的配置文件中为节点 node-01.example.com 配置名为 APC 的 fence 方法：

```
ccs -h node01.example.com --addmethod APC node01.example.com
```

2. 为该方法添加 fence 事务。您必须为该节点指定要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体在这个节点中这个方法的所有选项：

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中配置 fence 事务，该节点使用该 fence 设备中名为 apc 的 APC 电源切换端口 1 使用名为 APC 的方法 fence 集群节点 node-01.example.com：

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
```

您需要在该集群的每个节点中添加 fence 方法。下面的命令使用名为 APC 方法为每个节点配置 fence 方法。使用该 fence 方法的设备指定 apc 作为设备名称，即之前使用 --addfencedev 选项指定的设备，如 [第 5.5 节 “配置 Fence 设备”](#) 所述。每一个节点都是使用唯一的 APC 切换电源端口号指定

: node-01.example.com 的端口号为 1, node-02.example.com 的端口号为 2, node-03.example.com 的端口号为 3。

```
ccs -h node01.example.com --addmethod APC node01.example.com
ccs -h node01.example.com --addmethod APC node02.example.com
ccs -h node01.example.com --addmethod APC node03.example.com
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
ccs -h node01.example.com --addfenceinst apc node02.example.com APC port=2
ccs -h node01.example.com --addfenceinst apc node03.example.com APC port=3
```

例 5.2 “cluster.conf 添加使用电源的 Fence 方法后” 演示了您在集群的每个节点中添加这些 fencing 方法和事务后的 cluster.conf 配置文件：

例 5.2. cluster.conf 添加使用电源的 Fence 方法后

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
    passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>
```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 第 5.12 节 “在集群节点中传播配置文件” 所述。

5.6.2. 为节点配置单一存储 Fence 设备

当使用非电源 fencing 方法（即 SAN/存储 fencing）fence 某个节点时，您必须为该 fence 设备配置 unfencing。这样可保证被 fence 的节点在重启该节点前不会重新启用。当您为某个节点配置 unfencing 时，您可以指定一个与对应 fence 设备成镜像的设备，该设备是您明确使用 on 或者 enable 为该节点配置的。

有关 unfencing 某个节点的详情请参考 `fence_node(8) man page`。

使用以下步骤配置使用单一存储 fence 设备的节点，该 fence 设备名为 `sanswitch1`，使用 `fence_sanbox2 fencing` 代理。

1. 在该节点中添加 fence 方法，并为该 fence 方法提供名称。

```
ccs -h host --addmethod method node
```

例如：请执行以下命令在集群节点 `node-01.example.com` 的配置文件中为节点 `node-01.example.com` 配置名为 SAN 的 fence 方法：

```
ccs -h node01.example.com --addmethod SAN node01.example.com
```

2. 为该方法添加 fence 事务。您必须为该节点指定要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体在这个节点中这个方法的所有选项：

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

例如：请执行以下命令在集群节点 `node-01.example.com` 的配置文件中配置 fence 事务，该节点使用名为 `sanswitch1` 的 fence 设备的 SAN 切换电源端口 11 fence 使用名为 SAN 方法的集群节点 `node-01.example.com`：

```
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN port=11
```

3. 请执行以下命令为这个节点中基于存储的 fence 设备配置 unfencing：

```
ccs -h host --addunfence fencedevicename node action=on|off
```

您需要为该集群中的每个节点添加 fence 方法。下面的命令为每个节点配置了名为 SAN 的 fence 方法。使用该 fence 方法的设备指定 `sanswitch` 为设备名称，它是之前使用 `--addfencedev` 选项配置的设备，如 [第 5.5 节“配置 Fence 设备”](#) 所示。为每个节点配置一个唯一 SAN 物理端口号：`node-01.example.com` 的端口号为 11，`node-02.example.com` 的端口号为 12，`node-03.example.com` 的端口号为 13。

```
ccs -h node01.example.com --addmethod SAN node01.example.com
ccs -h node01.example.com --addmethod SAN node02.example.com
ccs -h node01.example.com --addmethod SAN node03.example.com
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN port=11
ccs -h node01.example.com --addfenceinst sanswitch1 node02.example.com SAN port=12
ccs -h node01.example.com --addfenceinst sanswitch1 node03.example.com SAN port=13
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com port=11 action=on
ccs -h node01.example.com --addunfence sanswitch1 node02.example.com port=12 action=on
ccs -h node01.example.com --addunfence sanswitch1 node03.example.com port=13 action=on
```

例 5.3 “cluster.conf 添加基于存储的 Fence 方法后” 演示了在该集群的每个节点中添加 fencing 方法、fencing 事务以及 unfencing 后的配置文件。

例 5.3. cluster.conf 添加基于存储的 Fence 方法后

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="12"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="12" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="13"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="13" action="on"/>
      </unfence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
  </fencedevices>
</rm>
</rm>
</cluster>

```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.6.3. 配置备用 Fence 设备

您可以为一个节点定义多个 fencing 方法。如果使用第一个方法 fencing 失败，则系统会尝试使用第二个方法 fence 该节点，然后是其它您配置的方法。要为节点配置备用 fencing 方法，您需要为一个节点配置两个方法，并为每个节点配置一个 fence 事务。

**注意**

系统使用您所配置的 fencing 方法的顺序与其在配置文件中的顺序一致。使用 ccs 命令配置的第一个方法是首选 fencing 方法，您配置的第二个方法是备用 fencing 方法。要更改顺序，您可以从配置文件中删除首选 fencing 方法，然后再将其添加回配置文件中。

备注：您可在任何时候执行以下命令输出当前为某个节点配置的 fence 方法和事务列表。如果您没有指定节点，这个命令将列出为所有节点当前配置的 fence 方法和事务。

```
ccs -h host --lsfenceinst [node]
```

使用以下步骤为某个节点配置首选 fencing 方法，该方法使用名为 apc 的 fence 设备，该设备使用 fence_apc fencing 代理，并配置使用名为 sanswitch1 的 fence 设备作为备用 fencing 设备，该设备使用 fence_sanbox2 fencing 代理。因为 sanswitch1 设备是基于存储的 fencing 代理，所以您还需要为那个失败配置 unfencing。

1. 为该节点添加首选 fence 方法，并为该 fence 方法命名。

```
ccs -h host --addmethod method node
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中为节点 node-01.example.com 将名为 APC 的 fence 方法配置为首选方法：

```
ccs -h node01.example.com --addmethod APC node01.example.com
```

2. 为首选方法添加 fence 事务。您必须指定要该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体到这个节点该方法的所有选项：

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中配置 fence 事务，该节点使用该 fence 设备中名为 apc 的 APC 电源切换端口 1 使用名为 APC 的方法 fence 集群节点 node-01.example.com：

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
```

3. 为该节点添加备用 fence 方法，并提供该 fence 方法的名称。

```
ccs -h host --addmethod method node
```

例如：请执行以下命令为集群节点 node-01.example.com 配置文件中的节点 node-01.example.com 配置名为 SAN 的 fence 方法：

```
ccs -h node01.example.com --addmethod SAN node01.example.com
```

4. 为该备用方法添加 fence 事务。您必须指定该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体在这个节点中该方法要使用的所有选项：

```
ccs -h host --addfenceinst fencedevicename node method [optionsr]
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中配置 fence 事务，该节点使用名为 sanswitch1 的 fence 设备的 SAN 切换电源端口 11 fence 使用名为 SAN 方法的集群节点 node-01.example.com：

```
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN port=11
```

5. 因为 sanswitch1 设备是基于存储的设备，您必须为这个设备指定 unfencing。

```
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com port=11 action=on
```

您可以根据需要继续添加 fencing 方法。

这个过程为集群中的一个节点配置 fence 设备和备用 fence 设备。您还需要为该集群中的其它节点配置 fencing。

例 5.4 “cluster.conf 添加备用 Fence 方法后” 演示了在该集群的每个节点中添加使用电源的主要备用 fencing 方和基于存储的备用 fencing 方法后的 cluster.conf 配置文件。

例 5.4. cluster.conf 添加备用 Fence 方法后

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="12"/>
        </method>
      </fence>
      <unfence>
```

```

        <device name="sanswitch1" port="12" action="on"/>
    </unfence>
</clusternode>
<clusternode name="node-03.example.com" nodeid="3">
    <fence>
        <method name="APC">
            <device name="apc" port="3"/>
        </method>
        <method name="SAN">
            <device name="sanswitch1" port="13"/>
        </method>
    </fence>
    <unfence>
        <device name="sanswitch1" port="13" action="on"/>
    </unfence>
</clusternode>
</clusternodes>
<fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"
    </fencedevices>
</rm>
</rm>
</cluster>

```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。



注意

系统使用您配置的 fencing 方法的顺序与其在集群配置文件中的顺序一致。您配置的第一个方法就是首选 fencing 方法，您配置的第二个方法就是备用 fencing 方法。要更改顺序，您可以从配置文件中删除首选 fencing 方法，然后再添加回来。

5.6.4. 配置使用冗余电源的节点

如果您将集群配置为使用冗余电源供应您的节点，则您必须确定配置 fencing 以便在节点需要被 fence 时可完全关闭它们。如果您将每个电源供应都设定为独立的 fence 方法，则会分别 fence 每个电源供应；第二个电源供应可在 fence 了第一个电源供应时让该系统继续运行，同时根本不会 fence 该系统。要将系统配置为双电源供应，您必须配置 fence 设备以便可关闭两个电源并完全关闭系统。这要求您在一个 fencing 方法中配置两个事务，且在您配置的每个事务中，要在为设备配置 action 属性 on 之前配置 action 属性 off。

请按照本小节中的步骤为有双电源供应的节点配置 fencing。

1. 在您可以为使用冗余电源的节点配置 fencing 强，您必须为该集群将每个电源开关配置为 fence 设备。有关配置 fence 设备的详情请参考 [第 5.5 节 “配置 Fence 设备”](#)。

请执行以下命令输出当前为您的集群配置的 fence 设备列表：

```
ccs -h host --lsfencedev
```

2. 在该节点中添加 fence 方法，并为该 fence 方法提供名称。

```
ccs -h host --addmethod method node
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中为节点 node-01.example.com 配置名为 APC-dual 的 fence 方法：

```
ccs -h node01.example.com --addmethod APC-dual node01.example.com
```

3. 为该 fence 方法的第一个电源添加 fence 事务。您必须指定该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体在这个节点中可用于这个方法的所有选项。此时您可将 action 属性配置为 on。

```
ccs -h host --addfenceinst fencedevicename node method [options] action=off
```

例如：您可以执行以下命令在集群节点 node-01.example.com 的配置文件中配置 fence 事务，该事务在名为 apc1 的 fence 设备中使用 APC 切换电源端口 1 fence 集群节点 node-01.example.com，fence 方法名为 APC-dual，并将 action 属性设定为 on：

```
ccs -h node01.example.com --addfenceinst apc1 node01.example.com APC-dual port=1 action=off
```

4. 为该 fence 方法的第二电源供应添加 fence 事务。您必须指定该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体在这个节点中这个方法的所有选项。此时您也可将这个事务将 action 属性配置为 off：

```
ccs -h host --addfenceinst fencedevicename node method [options] action=off
```

例如：请执行以下命令在集群节点 node-01.example.com 的配置文件中配置第二 fence 事务，该事务在名为 apc2 的 fence 设备中使用 APC 切换电源端口 1 fence 集群节点 node-01.example.com，该节点使用您为第一个事务所指定的同一方法，即 APC-dual，并将 action 属性设定为 off：

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com APC-dual port=1 action=off
```

5. 此时，您可为给 fence 方法的第一电源供应添加另一个 fence 事务，将 action 属性配置为 on。您必须指定该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体到这个节点这个方法的所有选项，并将 action 属性指定为 off：

```
ccs -h host --addfenceinst fencedevicename node method [options] action=on
```

例如：请执行以下命令为集群节点 node-01.example.com 的配置文件配置 fence 事务，该事务使用名为 apc1 的 fence 设备的 APC 切换电源端口 1 fence 集群节点 node-01.example.com，使用的方法名为 APC-dual，并将 action 属性设定为 on：

```
ccs -h node01.example.com --addfenceinst apc1 node01.example.com APC-dual port=1 action=on
```

6. 您可为给 fence 方法的第二电源供应添加另一个 fence 事务，将 action 属性指定为 on。您必须指定该节点要使用的 fence 设备，应用这个事务的节点，该方法的名称以及具体到这个节点这个方法的所有选项，并将 action 属性指定为 on:

```
ccs -h host --addfenceinst fencedevicename node method [options] action=on
```

例如：请执行以下命令为集群节点 node-01.example.com 的配置文件配置第二个 fence 事务，该事务使用名为 apc2 的 fence 设备的 APC 切换电源端口 1 fence 集群节点 node-01.example.com，使用的方法名为 APC-dual1，并将 action 属性设定为 off:

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com APC-dual port=1 action=on
```

例 5.5 “cluster.conf 添加双电源 Fencing” 演示了在集群的每个端口中使用两个电源供应时添加 fencing 后的 cluster.conf 配置文件。

例 5.5. cluster.conf 添加双电源 Fencing

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="1"action="off"/>
          <device name="apc2" port="1"action="off"/>
          <device name="apc1" port="1"action="on"/>
          <device name="apc2" port="1"action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="2"action="off"/>
          <device name="apc2" port="2"action="off"/>
          <device name="apc1" port="2"action="on"/>
          <device name="apc2" port="2"action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="3"action="off"/>
          <device name="apc2" port="3"action="off"/>
          <device name="apc1" port="3"action="on"/>
          <device name="apc2" port="3"action="on"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc1"
    passwd="password_example"/>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc2"
    passwd="password_example"/>
  </fencedevices>
</cluster>
```

```
</fencedevices>
</rm>
</rm>
</cluster>
```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.6.5. 删除 Fence 方法和 Fence 事务

请执行以下命令从您的配置文件中删除 fence 方法：

```
ccs -h host --rmmethod method node
```

例如：请执行以下命令在集群节点 `node01.example.com` 的集群配置文件中删除您为 `node01.example.com` 配置的名为 APC 的 fence 方法：

```
ccs -h node01.example.com --rmmethod APC node01.example.com
```

请执行以下命令从 fence 方法中删除某个 fence 设备的所有 fence 事务：

```
ccs -h host --rmfenceinst fencedevicename node method
```

例如：请执行以下命令在集群节点 `node01.example.com` 的集群配置文件中从为 `node01.example.com` 配置的名为 APC-dual 的方法中删除 fence 设备名为 `apc1` 的所有事务：

```
ccs -h node01.example.com --rmfenceinst apc1 node01.example.com APC-dual
```

5.7. 配置故障切换域

故障切换域是节点集群的一个命名子集，可在节点失败事件中运行集群服务。故障切换域有以下特征：

- 无限制 — 可让您指定您喜欢的子集成员，但分配给这个域的集群服务可在任意成员中运行。
- 限制的 — 可让您限制可运行具体集群服务的成员。如果在限制的故障切换域中没有任何成员可用，则无法启动该集群服务（手动或者使用集群软件都不行）。
- 无序的 — 当为某个无序的故障切域分配集群服务时，运行该集群服务的域的成员是从可用故障切换域成员中随即挑选的。
- 有序的 — 可让您在故障切换域的成员间指定顺序。该列表顶端的成员是首选成员，接下来是列表中的第二个成员，依此类推。
- 返回 — 可让您指定是否让故障切换域中的服务返回节点失败前运行该服务的节点中。配置这个特性对节点会反复失败的环境很有用，且该节点是有序故障切换域的一部分。在那个环境中，如果某个节点是故障切换域中的首选节点，那么某个服务就可能不断在首选节点和另一个节点间进行故障切换和返回，从而影响服务器性能。



备注

只有在配置了有序故障切换时方可应用返回属性。



注意

更改故障切换域配置对当前运行中的服务没有影响。



备注

操作不需要故障切换域。

默认情况下故障切换域是无限制且无序的。

在有几个成员的集群中，使用限制的故障切换域可尽量减少配置集群运行集群服务（比如 httpd）的工作，这些配置要求您在运行该集群服务的所有成员中设定完全相同的配置。与其设定整个集群运行该集群服务，您可以只在与该集群服务关联的限制故障切换域成员中进行设定。



备注

要配置首选成员，您可以创建只有一个集群成员的无限制故障切换域。这样做就让集群服务主要在那个集群成员（首选成员）中运行，但允许将该集群服务故障切换到任意其它成员中。

请执行以下步骤配置故障切换域：

1. 请执行以下命令添加故障切换域：

```
ccs -h host --addfailoverdomain name [restricted] [ordered] [nofailback]
```



注意

该名称应可显示出与您集群中使用的其它名称在使用目的上有区别。

例如：以下命令在无限制、有序且允许返回的 node-01.example.com 中配置名为 example_pri 的故障切换域：

```
ccs -h node-01.example.com --addfailoverdomain example_pri ordered
```

2. 请执行以下命令在故障切换域中添加一个节点:

```
ccs -h host --addfailoverdomainnode failoverdomain node priority
```

例如: 请执行以下命令在 node-01.example.com 的配置文件中配置故障切换域 example_pri, 这样它就包括优先权为 1 的 node-01.example.com, 优先权为 2 的 node-02.example.com 和优先权为 3 的 node-03.example.com:

```
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-01.example.com 1
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-02.example.com 2
ccs -h node-01.example.com --addfailoverdomainnode example_pri node-03.example.com 3
```

您可使用以下命令列出在集群中配置的所有故障切换域和故障切换域节点:

```
ccs -h host --lsfailoverdomain
```

请执行以下命令删除故障切换域:

```
ccs -h host --rmfailoverdomain name
```

请执行以下命令从故障切换域中删除节点:

```
ccs -h host --rmfailoverdomainnode failoverdomain node
```

请注意: 您完成集群的所有组件配置后, 需要在所有节点中同步该集群配置文件, 如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.8. 配置全局集群资源

您可以配置两类资源:

- 全局 — 可在集群的任意服务中使用的资源。
- 具体服务 — 只可用于一个服务的资源。

请执行以下命令查看当前在该集群中配置的资源和服务列表:

```
ccs -h host --lsservices
```

请执行以下命令添加全局集群资源。您可在配置该资源时添加某个具体服务的本地资源, 如 [第 5.9 节 “在集群中添加集群服务”](#) 所述。

```
ccs -h host --addresource resourcetype [resource options] ...
```


例如：下面的命令在 `node01.example.com` 的集群配置文件中添加了一个全局文件系统资源。该资源的名称为 `web_fs`，该文件系统设备为 `/dev/sdd2`，挂载点为 `/var/www`，类型为 `ext3`。

```
ccs -h node01.example.com --addresource fs name=web_fs device=/dev/sdd2 mountpoint=/var/www fstype=ext3
```

有关可用资源类型和资源选项的详情请参考 [附录 B, HA 资源参数](#)。

请执行以下命令删除全局资源：

```
ccs -h host --rmresource resourcetype [resource options]
```

如果您要修改现有全局资源参数，您可以删除该资源然后再重新配置。

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.9. 在集群中添加集群服务

请执行以下步骤在集群中配置集群服务：

1. 请使用以下命令在集群中添加服务：

```
ccs -h host --addservice servicename [service options]...
```



注意

为该服务使用可明确与集群中其它服务区分开来的说明性名称。

请在集群配置中添加服务时配置以下属性：

- `autostart` — 指定在集群启动时是否自动启动该服务。
- `domain` — 指定故障切换域（如果要求）。
- `exclusive` — 指定服务只能在没有其它服务运行的节点中运行的策略。
- `recovery` — 为该服务指定恢复策略。其选项为 `relocate`、`restart`、`disable` 或者 `restart-disable` 该服务。有关服务恢复策略的详情请参考 [表 B.18 “服务”](#)。

例如：请执行以下命令在集群节点 `node-01.example.com` 的配置文件中添加名为 `example_apache` 的服务，该服务使用故障切换域 `example_pri`，恢复策略为 `relocate`：

```
ccs -h node-01.example.com --addservice example_apache domain=example_pri recovery=relocate
```

2. 请使用以下命令在服务中添加资源：

```
ccs -h host --addsubservice servicename subservice [service options]...
```

请根据您想要使用的资源类型使用全局或者具体服务资源传播服务。请在添加资源时使用 `ccs` 命令的 `--addsubservice` 选项添加全局资源。例如：请执行以下命令在 `node-01.example.com` 集群配置文件中为名为 `example_apache` 的服务添加名为 `web_fs` 的全局文件系统资源：

```
ccs -h node01.example.com --addsubservice example_apache fs ref=web_fs
```

要在服务中添加具体服务资源，您需要指定所有服务选项。例如：如果您之前没有将 `web_fs` 定义为全局服务，则您可以使用以下命令将其作为具体服务资源：

```
ccs -h node01.example.com --addsubservice example_apache fs name=web_fs device=/dev/sdd2 mountpoint=/var/www fstype=ext3
```

3. 要在该服务中添加子服务，您还可以使用 `ccs` 命令的 `--addsubservice` 选项指定服务选项。

如果您需要在相依性树状结构中添加服务，请使用冒号（`:`）分隔元素，并使用括号区分同一类型的子服务。下面的示例添加了第三个 `nfscient` 服务作为 `nfscient` 服务的子服务，它本身是 `nfscient` 服务的子服务，而后者又是 `service_a` 服务的子服务：

```
ccs -h node01.example.com --addsubservice service_a nfscient[1]:nfscient[2]:nfscient
```



注意

如果您要添加 Samba 服务资源，请直接在该服务中添加，不要将其作为另一个资源的子资源使用。



备注

要确认在目前在集群服务中使用的 IP 服务资源，您必须在集群节点中运行 `/sbin/ip addr list` 命令。下面的输出演示了在运行集群服务的节点中执行该命令的输出：

```
1: lo: <LOOPBACK,UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP> mtu 1356 qdisc pfifo_fast qlen 1000
    link/ether 00:05:5d:9a:d8:91 brd ff:ff:ff:ff:ff:ff
    inet 10.11.4.31/22 brd 10.11.7.255 scope global eth0
    inet6 fe80::205:5dff:fe9a:d891/64 scope link
    inet 10.11.4.240/22 scope global secondary eth0
        valid_lft forever preferred_lft forever
```

请执行以下命令删除服务及其所有子服务：

```
ccs -h host --rmservice servicename
```

请执行以下命令删除子服务：

```
ccs -h host --rmsubservice servicename subservice [service options]...
```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

5.10. 配置仲裁磁盘

 **重要**

Quorum-disk 参数以及探测法根据网站环境以及特殊需要有所不同。要了解如何使用 Quorum-disk 参数和探测法，请参考 `qdisk(5) man page`。如果您在理解和使用仲裁磁盘方面需要帮助，请联络授权的红帽支持代表。

请使用以下命令将您的系统配置为使用仲裁磁盘：

```
ccs -h host --setquorumd [quorumd options] ...
```

[表 5.1 “仲裁磁盘选项”](#) 总结了您可能需要设置的仲裁磁盘选项的含义。有关仲裁磁盘参数的完整列表请参考 `/usr/share/cluster/cluster.rng` 中的集群方案以及 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中的注释方案。

表 5.1. 仲裁磁盘选项

参数	描述
interval	读取/写入循环的频率，单位为秒。
votes	仲裁守护进程告知 cman 它有足够分数时的票数。
tko	宣布节点死亡时节点必须错过的循环数。
min_score	将节点视为 "alive" 的最小分数。如果省略或者设为 0，则使用默认功能 $\text{floor}((n+1)/2)$ ，其中 n 为探测法分数之和。Minimum Score 值永远不能超过探测法分数之和，否则将无法使用该仲裁磁盘。
device	制裁守护进程使用的存储设备。在所有节点中该设备必须相同。
label	指定由 mkqdisk 工具创建的制裁磁盘标签。如果这个字段中包含一个条目，则该标签将覆盖 Device 字段。如果这个字段已经被使用，则仲裁守护进程会在每个找到的块设备中读取 <code>/proc/partitions</code> 并检查 qdisk 签名，根据指定的标签对比该标签。这在节点间使用不同仲裁设备名称配置时有用。

请使用以下命令为仲裁磁盘配置探测法：

```
ccs -h host --addheuristic [heuristic options] ...
```

表 5.2 “仲裁磁盘探测法” 总结了您可能需要设置的仲裁磁盘探测法的含义。

表 5.2. 仲裁磁盘探测法

参数	描述
program	用来确定这个探测法是否可用的程序路径。它可以是任意可由 <code>/bin/sh -c</code> 的路径。返回值 0 表示成功，返回任意其它值表示失败。使用仲裁磁盘需要这个参数。
interval	调用探测法的频率（单位为秒）。每个探测法间的默认间隔为 2 秒。
score	这个探测法的加权。决定探测法分数时请小心。每个探测法的默认分数为 1。
tko	在宣布这个探测法不可用前连续失败的次数。

您可执行以下命令查看系统配置的仲裁磁盘选项和探测法：

```
ccs -h host --lsquorum
```

您可执行以下命令删除探测法选项指定的探测法：

```
ccs -h host rmheuristic [heuristic options]
```

请注意：您完成集群的所有组件配置后，需要在所有节点中同步该集群配置文件，如 第 5.12 节 “在集群节点中传播配置文件” 所述。

 **备注**

同步和激活可传播并激活更新的集群配置文件。但对于要操作的仲裁磁盘，您必须重启该集群（请参考 第 6.2 节 “启动和停止集群”）。

5.11. 其它集群配置

本小节描述了如何使用 `ccs` 命令配置以下内容：

- 第 5.11.1 节 “集群配置版本”
- 第 5.11.2 节 “多播配置”
- 第 5.11.3 节 “配置双节点集群”

您还可以使用 `ccs` 命令设定高级集群配置参数，其中包括 `totem` 选项、`d1m` 选项、`rm` 选项和 `cman` 选项。有关设定这些参数的详情请参考 `ccs(8)` man page 以及 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中注释的集群配置文件方案。

请执行以下命令查看为集群配置的其它集群属性：

```
ccs -h host --lsmisc
```

5.11.1. 集群配置版本

集群配置版本包括集群配置版本值。默认在创建集群配置文件时将配置版本值设定为 1，并在每次您修改集群配置时自动增加。但如果您将其设定为其它数值，您可以使用以下命令指定该数值：

```
ccs -h host --setversion n
```

您可使用以下命令在获得现有集群配置文件的当前配置版本值：

```
ccs -h host --getversion
```

请执行以下命令将集群中每个节点的集群配置文件中的当前配置版本值加 1：

```
ccs -h host --incversion
```

5.11.2. 多播配置

如果您没有在集群配置文件中指定多播地址，红帽高可用性附加组件软件可根据集群 ID 创建一个。它可生成地址的后 16 位数字，并根据所使用的 IP 协议（IPv4 或者 IPv6）将其附加到该地址中：

- 对于 IPv4 — 地址格式为 239.192. 加上由红帽高可用性附加组件软件生成的最后 16 位数字。
- 对于 IPv6 — 地址格式为 FF15:: 加上由红帽高可用性附加组件软件生成的最后 16 位数字。



注意

集群 ID 是 cman 为每个集群生成的唯一识别符。请在集群节点中运行 `cman_tool status` 命令查看集群 ID。

您可使用以下命令在集群配置文件中手动指定多播地址：

```
ccs -h host --setmulticast multicastaddress
```

如果您指定了多播地址，您应该使用 cman 使用的 239.192.x.x 系列（IPv6 使用 FF15::）。使用该范围以外的多播地址将导致不可预测的结果。例如：使用 224.0.0.x（即“网络中的所有主机”）可能无法正确路由，或者在有些硬件中根本无法路由。



注意

如果您指定了多播地址，请确定检查集群数据包通过的路由器配置。有些路由器可能需要较长时间了解这些地址，这样会严重影响集群性能。

要删除配置文件中的多播地址，请使用 `ccs` 的 `--setmulticast` 选项，但不要指定多播地址：

```
ccs -h host --setmulticast
```

5.11.3. 配置双节点集群

如果您要配置双节点集群，您可执行以下命令允许单一节点维护仲裁（例如：如果一个节点失败）：

```
ccs -h host --setcman two_node=1 expected_votes=1
```

5.12. 在集群节点中传播配置文件

您在集群节点之一创建或者编辑集群配置文件后，需要将同一文件传播到所有集群节点并激活该配置。

使用以下命令传播并激活集群配置文件：

```
ccs -h host --sync --activate
```

请执行以下命令确定主机集群配置文件中指定的所有节点有相同的集群配置文件：

```
ccs -h host --checkconf
```

如果您在本地节点创建或者编辑了配置文件，请执行以下命令将其发送到该集群的一个节点中：

```
ccs -f file -h host --setconf
```

请执行以下命令验证该本地文件中指定的所有节点有相同的集群配置文件：

```
ccs -f file --checkconf
```

使用 ccs 管理 Red Hat 高可用性附加组件

本章论述了使用 ccs 工具管理 Red Hat 高可用性附加组件的各种管理任务，在红帽企业版 Linux 6.1 以及之后的版本中支持这个工具。本章由以下小节组成：

- [第 6.1 节 “管理集群节点”](#)
- [第 6.2 节 “启动和停止集群”](#)
- [第 6.3 节 “诊断并修正集群中的问题”](#)

6.1. 管理集群节点

本小节记录了如何使用 ccs 命令执行以下节点管理功能：

- [第 6.1.1 节 “使节点离开或者加入集群”](#)
- [第 6.1.2 节 “在运行的集群中添加成员”](#)

6.1.1. 使节点离开或者加入集群

您可以使用 ccs 命令通过停止在那个节点中的集群服务让节点离开集群。让节点离开集群不会从那个节点中删除集群配置信息。让节点离开集群可防止在重启是将其自动加入该集群。

请执行以下命令让节点离开集群，指定 -h 选项可停止该节点中的集群服务：

```
ccs -h host --stop
```

您停止节点中的集群服务时，会故障切换所有在该节点中运行的服务。

要从集群配置文件中整个删除一个节点，请使用 ccs 命令的 --rmnode 选项，如 [第 5.4 节 “创建集群”](#) 所述。

请执行以下命令让节点重新加入集群，可指定 -h 选项启动该节点中的集群服务：

```
ccs -h host --start
```

6.1.2. 在运行的集群中添加成员

要在运行的集群中添加成员，可如 [第 5.4 节 “创建集群”](#) 所述在集群中添加节点。更新配置文件后，请将其传播到该集群的所有节点中，并确定激活了新的集群配置文件，如 [第 5.12 节 “在集群节点中传播配置文件”](#) 所述。

6.2. 启动和停止集群

您可以使用 ccs 停止集群，并使用以下命令停止集群中所有节点中的集群服务：

```
ccs -h host --stopall
```

您可以使用 `ccs` 启动集群，并使用以下命令启动集群中所有节点中的集群服务：

```
ccs -h host --startall
```

6.3. 诊断并修正集群中的问题

有关诊断并修正集群中问题的详情请参考 [第 9 章 诊断并修正集群中的问题](#)。这里有一些您可以使用 `ccs` 命令执行的简单检查。

请执行以下命令验证主机集群配置文件中的所有节点是否有相同的集群配置文件：

```
ccs -h host --checkconf
```

如果您在本地节点中创建或者编辑配置文件，您可以使用以下命令验证在本地文件中指定的所有节点是否有相同的集群配置文件：

```
ccs -f file --checkconf
```


使用命令行工具配置红帽高可用性附加组件

本章论述了如何通过直接编辑集群配置文件（`/etc/cluster/cluster.conf`）以及使用命令行工具配置红帽高可用性附加组件软件。本章分节提供了有关构建配置文件的步骤，从本章提供的简单文件开始。另外，您可以使用这里提供的简单文件，从 `cluster.conf man page` 复制配置文件框架。但这样做就无法与本章之后所提供步骤的信息对应。还有其它方法可创建并配置集群文件。本章分节提供有关构建配置文件的步骤。另外请记住这只是编写适合您集群所需配置文件的起点。

本章由以下节组成：

- 第 7.1 节 “配置任务”
- 第 7.2 节 “配置基本集群配置文件”
- 第 7.3 节 “配置 Fencing”
- 第 7.4 节 “配置故障切换域”
- 第 7.5 节 “配置 HA 服务”
- 第 7.6 节 “确认配置”



重要

请确定您部署的高可用性附加组件满足您的需要并可被支持。部署前请咨询授权红帽代表确认您的配置。另外预留足够的时间测试配置的失败模式。



重要

本章通常使用 `cluster.conf` 元素和属性作为参考。有关 `cluster.conf` 元素和属性的完整列表，请参考 `/usr/share/cluster/cluster.rng` 中的集群方案，注释的方案位于 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。



重要

本章的某些步骤调用 `cman_tool -r` 命令在整个集群中传播集群配置。使用该命令需要运行 `ricci`。使用 `ricci` 要求您第一次从某台具体机器与 `ricci` 互动时输入密码。有关 `ricci` 服务的详情请参考 第 2.11 节 “[ricci 注意事项](#)”。



备注

本章中的步骤可能包括某些命令行工具的具体命令，如 [附录 D，命令行工具小结](#) 所示。有关所有命令和变量的详情请参考每个命令行工具的 `man page`。

7.1. 配置任务

使用命令行工具配置红帽高可用性附加组件步骤如下：

1. 创建一个集群。请参考 [第 7.2 节 “配置基本集群配置文件”](#)。
2. 配置 fencing。请参考 [第 7.3 节 “配置 Fencing”](#)。
3. 配置故障切换域。请参考 [第 7.4 节 “配置故障切换域”](#)。
4. 配置 HA 服务。请参考 [第 7.5 节 “配置 HA 服务”](#)。
5. 验证配置。请参考 [第 7.6 节 “确认配置”](#)。

7.2. 配置基本集群配置文件

如果安装了集群硬件、红帽企业版 Linux 和高可用性附加组件软件，您就可以创建一个集群配置文件（`/etc/cluster/cluster.conf`）并开始运行高可用性附加组件。本小节描述了如何创建没有 fencing、故障切换以及 HA 服务的配置文件框架，这只是个起点。后续的小节将描述如何在配置文件中配置那些部分。



重要

这只是创建集群配置文件的临时步骤，所生成的文件不包含任何 fencing，也不会将其视为支持的配置。

以下步骤描述了如何创建并配置集群配置文件框架。最终，您集群的配置文件在节点数、fencing 类型、HA 服务的类型及数目以及其它具体位置要求方面会有所不同。

1. 在集群中的任意节点使用 [例 7.1 “cluster.conf 示例：基本配置”](#) 中的示例模板创建 `/etc/cluster/cluster.conf`。
2. （可选），如果您要配置有两个节点的集群，您可以在配置文件中添加以下行以便允许单节点维持定额（例如：如果一个节点失败）：

```
<cman two_node="1" expected_votes="1"/>
```

请参考 [例 7.2 “cluster.conf 示例：基本双节点配置”](#)。

3. 使用 `cluster` 属性：`name` 和 `config_version` 指定集群名称以及配置版本号（请参考 [例 7.1 “cluster.conf 示例：基本配置”](#) 或者 [例 7.2 “cluster.conf 示例：基本双节点配置”](#)）。
4. 在 `clusternodes` 部分，请使用 `clusternode` 属性：`name` 和 `nodeid` 为每个节点指定节点名称和节点 ID。

5. 保存 `/etc/cluster/cluster.conf`。
6. 根据集群方案 (`cluster.rng`) 通过运行 `ccs_config_validate` 命令验证该文件。例如:

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. 将配置文件放在每个集群节点的 `/etc/cluster/`。例如: 您可以使用 `scp` 命令将该文件放在其它集群节点中。



备注

在首次创建集群时需要使用这个方法传播集群配置文件。安装并运行集群后, 可使用 `cman_tool version -r` 传播集群配置文件。可以使用 `scp` 传播更新的配置文件, 但使用 `scp` 命令时必须所有节点中停止集群软件。另外, 如果您使用 `scp` 传播更新的配置文件, 您应该运行 `ccs_config_validate`。



备注

当在同一配置文件中有其它元素和属性时 (例如: `fence` 和 `fencedevices`), 现在则不需要传播它们。本章后面的步骤提供有关指定其它元素和属性的信息。

8. 启动集群。在每个集群节点中运行以下命令:

```
service cman start
```

例如:

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...          [ OK ]
  Global setup...                     [ OK ]
  Loading kernel modules...           [ OK ]
  Mounting configfs...                 [ OK ]
  Starting cman...                     [ OK ]
  Waiting for quorum...                [ OK ]
  Starting fenced...                   [ OK ]
  Starting dlm_controld...              [ OK ]
  Starting gfs_controld...              [ OK ]
  Unfencing self...                    [ OK ]
  Joining fence domain...               [ OK ]
```

9. 在任意集群节点中运行 `cman_tools nodes` 以确认那些节点作为集群的成员运行 (在状态列 "Sts" 中被标记为 "M")。例如:

```
[root@example-01 ~]# cman_tool nodes
```

Node	Sts	Inc	Joined	Name
1	M	548	2010-09-28 10:52:21	node-01.example.com
2	M	548	2010-09-28 10:52:21	node-02.example.com
3	M	544	2010-09-28 10:52:21	node-03.example.com

10. 如果集群正在运行，请执行 第 7.3 节 “配置 Fencing”。

基本配置示例

例 7.1 “cluster.conf 示例：基本配置”和 例 7.2 “cluster.conf 示例：基本双节点配置”（双节点集群）提供最基本的集群配置示例。本章的后续步骤将提供有关配置 fencing 和 HA 服务的信息。

例 7.1. cluster.conf 示例：基本配置

```
<cluster name="mycluster" config_version="2">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
  </fencedevices>
</rm>
</rm>
</cluster>
```

例 7.2. cluster.conf 示例：基本双节点配置

```
<cluster name="mycluster" config_version="2">
  <cman two_node="1" expected_votes="1"/>
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
  </fencedevices>
</rm>
</rm>
</cluster>
```

双节点集群中 totem 的 consensus 值

您创建双节点集群且不想之后为该集群添加附加节点时，您应该忽略 `cluster.conf` 文件 `totem` 标签的 `consensus` 值，以便根据以下规则自动计算 `consensus` 值：

- 如果有两个或者少于两个节点，`consensus` 值将在 200 毫秒到 2000 毫秒之间 ($\text{token} * 0.2$)。
- 如果有三个或者三个以上节点，`consensus` 值为 $\text{token} + 2000$ 毫秒。

如果您让 `cman` 工具以这种方式配置您的 `consensus` 超时，那么将来您从两个节点移动到三个（或者更多）节点中移开，则需要重启集群，因为 `consensus` 超时需要根据 `token` 超时更改较大的值。

如果您要配置一个双节点集群并想要在将来升级到两个以上节点，您可以覆盖这个 `consensus` 超时以便当从两个节点移动到三个（或者更多）节点时不需要重启集群。您可以在 `cluster.conf` 中进行如下操作以达到此目的：

```
<totem token="X" consensus="X + 2000" />
```

请注意：这个配置解析程序不会自动计算 $X + 2000$ 。您在此必须使用整数而不是等式。

在双节点集群中使用优化的 `consensus` 超时的优点是总体上降低了双节点故障切换耗时，因为 `consensus` 不是 `token` 超时的功能。

备注：在 `cman` 的双节点自动探测中，物理节点数是那些有用的节点而不是 `cluster.conf` 文件中出现的 `two_node=1` 指令。

7.3. 配置 Fencing

配置 fencing 包括 (a) 在一个集群中配置一个或者多个 fence 设备；(b) 为每个节点配置一个或者多个 fence 方法（使用指定的一个或者多个 fence 设备）。

根据您的配置所需 fence 设备类型和 fence 方法配置 `cluster.conf`，如下：

1. 在 `fencedevices` 部分，使用 `fencedevice` 元素和 fence 设备独立属性指定每个 fence 设备。
例 7.3 “添加到 `cluster.conf` 中的 APC Fence 设备” 演示添加了 APC fence 设备的配置文件示例。
2. 在 `clusternodes` 部分，每个 `clusternodes` 部分的 fence 元素中，指定每个节点的 fence 方法。使用 `method` 属性 `name` 指定 fence 方法名称。使用 `device` 元素及其属性 `name` 和具体 fence 设备参数为每个 fence 方法指定 fence 设备。例 7.4 “添加到 `cluster.conf` 的 fence 方法” 演示了集群中每个节点一个 fence 设备的 fence 方法示例。
3. 在非电源 fence 方法（即 SAN/存储 fencing）的 `clusternodes` 部分添加 `unfence` 字段。这可保证在重启该节点前不会重新启用被 fence 的节点。有关 unfencing 节点的详情请参考 `fence_node(8)` man page。

与 fence 部分不同，`unfence` 部分不包含 `method`。它直接包含 `device` 参考，使用 `"on"` 或者 `"enable"` 的明确动作 (action) 成为 fence 对应设备部分的镜像。fence 和 unfence device 行都参考同一 `fencedevice`，并应在每个节点中重复同样的参数。

将 `action` 属性指定为 `"on"` 或者 `"enable"` 可在重启时启用该节点。例 7.4 “添加到 `cluster.conf` 的 fence 方法” 和 例 7.5 “`cluster.conf`：每个节点中有多种 Fence 方法” 中包含 `unfence` 元素和属性示例。

有关 unfence 详情请参考 `fence_node` man page。

4. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。
5. 保存 `/etc/cluster/cluster.conf`。
6. （可选），运行 `ccs_config_validate` 命令，确认根据集群方案（`cluster.rng`）更新的文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. 请运行 `cman_tool version -r` 命令在剩余的所有集群节点中传播。这还将运行附加验证。需要在每个集群节点中都运行 `ricci` 方可传播更新的集群配置信息。
8. 确认传播了更新的文件。
9. 执行 第 7.4 节 “配置故障切换域”。

如果需要，您可以通过在每个节点中使用多 `fence` 方法和在每个 `fence` 方法中使用多个 `fence` 设备配置复杂配置。当在每个节点中指定多个 `fence` 方法时，如果使用第一个方法 `fenced` 执行 `fence` 操作失败，则 `fence` 守护进程会尝试下一个方法，然后继续循环尝试所有方法直到成功为止。

有时 `fencing` 节点需要禁用两个 I/O 路径或者两个电源端口。这可通过在 `fence` 方法中指定两个或者多个设备完成。`fenced` 为每个 `fence` 设备行运行一次 `fence` 代理；只有全部成功方可认为是成功。

“[Fencing 配置示例](#)”一节中演示了更复杂的配置。

您可以在 `fence` 设备代理 man page 中找到有关配置具体 `fence` 设备的更多信息（例如：`fence_apc` man page）。另外，您可以在 [附录 A, Fence 设备参数](#) 中获得有关 `fence` 参数的更多信息，在 `/usr/sbin/` 中获得 `fence` 代理的更多信息，在 `/usr/share/cluster/cluster.rng` 中获得有关集群方案的更多信息，在 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中获得有关注释方案的更多信息（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。

Fencing 配置示例

以下示例演示了每个节点一个 `fence` 方法以及每个 `fence` 方法一个 `fence` 设备的简单配置：

- [例 7.3 “添加到 `cluster.conf` 中的 APC Fence 设备”](#)
- [例 7.4 “添加到 `cluster.conf` 的 `fence` 方法”](#)

以下示例演示了较复杂的配置：

- [例 7.5 “`cluster.conf`：每个节点中有多种 Fence 方法”](#)
- [例 7.6 “`cluster.conf`：Fencing，多路径多端口”](#)
- [例 7.7 “`cluster.conf`：使用双电源 Fencing 节点”](#)



备注

本节中的示例并不完全，即还有其它方法根据您的要求配置 fencing。

例 7.3. 添加到 cluster.conf 中的 APC Fence 设备

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
    passwd="password_example"/>
  </fencedevices>
</rm>
</rm>
</cluster>
```

在这个示例中，已将 fence 设备（fencedevice）添加到 fencedevices 元素，指定 fence 代理（agent）为 fence_apc，IP 地址（ipaddr）为 apc_ip_example，登录（login）为 login_example，fence 设备名称（name）为 apc，以及密码（passwd）为 password_example。

例 7.4. 添加到 cluster.conf 的 fence 方法

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
```

```

        <method name="APC">
            <device name="apc" port="3"/>
        </method>
    </fence>
</clusternode>
</clusternodes>
<fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
</fencedevices>
<rm>
</rm>
</cluster>

```

在这个示例中为每个节点添加了 fence 方法 (method)。每个节点的 fence 方法名称 (name) 为 APC。将用于每个节点中 fence 方法的设备 (device) 名称 (name) 指定为 apc, 并为每个节点指定唯一 APC 切换电源端口号 (port)。例如: node-01.example.com 的端口号为 1 (port="1")。每个节点的设备名称 (device name="apc") 根据 fencedevices 元素 apc 行中的名称 (name) 指向 fence 设备: fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc" passwd="password_example"/。

例 7.5. cluster.conf: 每个节点中有多种 Fence 方法

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="12"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="12" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="13"/>
        </method>
      </fence>
      <unfence>

```



```

        <device name="sanswitch1" port="13" action="on"/>
    </unfence>
</clusternode>
</clusternodes>
<fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"
    </fencedevices>
</rm>
</rm>
</cluster>

```

例 7.6. cluster.conf: Fencing, 多路径多端口

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="SAN-multi">
          <device name="sanswitch1" port="11"/>
          <device name="sanswitch2" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
        <device name="sanswitch2" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="SAN-multi">
          <device name="sanswitch1" port="12"/>
          <device name="sanswitch2" port="12"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="12" action="on"/>
        <device name="sanswitch2" port="12" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="SAN-multi">
          <device name="sanswitch1" port="13"/>
          <device name="sanswitch2" port="13"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="13" action="on"/>
        <device name="sanswitch2" port="13" action="on"/>
      </unfence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example" >
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch2" passwd="password_example"
  </fencedevices>

```

```

    </rm>
  </rm>
</cluster>

```

例 7.7. cluster.conf: 使用双电源 Fencing 节点

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="1" action="off"/>
          <device name="apc2" port="1" action="off"/>
          <device name="apc1" port="1" action="on"/>
          <device name="apc2" port="1" action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="2" action="off"/>
          <device name="apc2" port="2" action="off"/>
          <device name="apc1" port="2" action="on"/>
          <device name="apc2" port="2" action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="3" action="off"/>
          <device name="apc2" port="3" action="off"/>
          <device name="apc1" port="3" action="on"/>
          <device name="apc2" port="3" action="on"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc1"
    passwd="password_example"/>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc2"
    passwd="password_example"/>
  </fencedevices>
  </rm>
</rm>
</cluster>

```

当使用电源切换到带双电源的 fence 节点时，必须告知代理在恢复两个电源端口前关闭两个电源端口。默认代理开关行为将导致永远无法在该节点完全禁用该电源。

7.4. 配置故障切换域

故障切换域是一个命名的集群节点子集，它可在节点失败事件中运行集群服务。故障切换域有以下特征：

- 无限制 — 允许您在子集指定首选成员，但分配给这个域名的集群服务可在任意可用成员中运行。
- 限制 — 允许您限制可运行特定集群服务的成员。如果在限制故障切换域中没有可用成员，则无法启动集群服务（手动或者使用集群软件均不可行）。
- 无序 — 当将一个集群服务分配给一个无序故障切换域时，则可从可用故障切换域成员中随机选择运行集群服务的成员，没有优先顺序。
- 有序 — 允许您在故障切换域成员中指定首选顺序。有序故障切换域从优先数字最小的节点开始选择。即故障切换域中优先数字为“1”的节点具有最高的优先权，因此它是故障切换域中的首选节点。那个节点之后，第二首选节点应为含有下一个最高优先权数字的节点，以此类推。
- 故障恢复 — 允许您指定在故障切换域中的服务是否应该恢复到节点失败前最初运行的节点。配置这个特性在作为有序故障切换域一部分节点重复失败的环境中很有帮助。在那种情况下，如果某个节点是故障切换域中的首选节点，在可能在首选节点和其它节点间重复切换和恢复某个服务，从而不会对性能产生严重影响。



备注

故障恢复特性只适用于配置了有序故障切换的集群。



备注

更改故障切换域配置对目前运行中的服务无效。



备注

故障切换域不需要操作。

默认情况下故障切换域为无限制和无序的。

在由几个成员组成的集群中，使用限制故障切换域可最大程度降低设置集群以便运行集群服务的工作（比如 httpd），它要求您在运行该集群服务的所有成员中进行完全一致的配置。您不需要将整个集群设置为运行该集群服务，您只要在与集群服务关联的限制故障切换域中的成员即可。



备注

要配置首选成员，您可以创建一个只由一个集群成员组成的无限故障切换域。这样可让该集群服务在那个主要集群成员（首选成员）中运行，但允许将集群服务切换到任意其它成员中。

1. 在集群的任意节点中打开 `/etc/cluster/cluster.conf`。
2. 为每个要使用的故障切换域 `rm` 元素中添加以下框架部分：

```
<failoverdomains>
  <failoverdomain name="" nofailback="" ordered="" restricted="">
    <failoverdomainnode name="" priority=""/>
    <failoverdomainnode name="" priority=""/>
    <failoverdomainnode name="" priority=""/>
  </failoverdomain>
</failoverdomains>
```



备注

`failoverdomainnode` 属性的数目由故障切换域中的节点数决定。前面文本中的框架 `failoverdomainnode` 部分有三个 `failoverdomainnode` 元素（没有指定节点名称），表示在故障切换域中有三个节点。

3. 在 `failoverdomainnode` 部分提供了元素和属性值。有关元素和属性描述请参考只是集群方案的 `failoverdomain` 部分。注释集群方案位于任意集群节点中的 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。有关 `failoverdomains` 部分示例请参考 例 7.8 “在 `cluster.conf` 中添加故障切换域”。
4. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。
5. 保存 `/etc/cluster/cluster.conf`。
6. （可选），运行 `ccs_config_validate` 命令，根据集群方案（`cluster.rng`）验证该文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. 运行 `cman_tool version -r` 命令在其它集群节点中传播该配置。
8. 执行 第 7.5 节 “配置 HA 服务”。

例 7.8 “在 `cluster.conf` 中添加故障切换域” 演示使用有序、无限故障切换域配置示例。

例 7.8. 在 `cluster.conf` 中添加故障切换域

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
```

```

    </fence>
  </clusternode>
  <clusternode name="node-02.example.com" nodeid="2">
    <fence>
      <method name="APC">
        <device name="apc" port="2"/>
      </method>
    </fence>
  </clusternode>
  <clusternode name="node-03.example.com" nodeid="3">
    <fence>
      <method name="APC">
        <device name="apc" port="3"/>
      </method>
    </fence>
  </clusternode>
</clusternodes>
<fencedevices>
  <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
</fencedevices>
<rm>
  <failoverdomains>
    <failoverdomain name="example_pri" nofailback="0" ordered="1" restricted="0">
      <failoverdomainnode name="node-01.example.com" priority="1"/>
      <failoverdomainnode name="node-02.example.com" priority="2"/>
      <failoverdomainnode name="node-03.example.com" priority="3"/>
    </failoverdomain>
  </failoverdomains>
</rm>
</cluster>

```

failoverdomains 部分包含集群中每个故障切换域中的 failoverdomains 部分。这个示例中有一个故障切换域。在 failoverdomains 行中，名称 (name) 为 example_pri。另外，它指定无故障恢复 (failback="0")，即故障切换为有序的 (ordered="1")，故障切换域为无限的 (restricted="0")。

7.5. 配置 HA 服务

配置 HA（高可用性）服务包括配置资源以及为服务分配资源。

以下小节描述了如何编辑 /etc/cluster/cluster.conf 添加资源和服务。

- 第 7.5.1 节 “添加集群资源”
- 第 7.5.2 节 “在集群中添加集群服务”



重要

配置高可用性资源和服务有很多可能性。要更好了解资源参数和资源行为，请参考 [附录 B，HA 资源参数](#) 和 [附录 C，HA 资源行为](#)。为优化性能并保证您的配置可被支持，请联络授权红帽支持代表。

7.5.1. 添加集群资源

您可配置两种类型的资源：

- 全局 — 集群中的任何服务都可用的资源。这些在配置文件的 resources 部分配置 (rm 元素中)。

- 具体服务 — 只在一个服务中可用的资源。这些在配置文件的每个 `service` 部分配置（在 `rm` 元素中）。

这小节描述了如何添加全局资源。有关配置具体服务资源的步骤请参考 [第 7.5.2 节 “在集群中添加集群服务”](#)。

要添加全局集群资源，请按照本小节中的步骤执行。

1. 在集群的任意节点中打开 `/etc/cluster/cluster.conf`。
2. 在 `rm` 元素中添加 `resources` 部分。例如：

```
<rm>
  <resources>

  </resources>
</rm>
```

3. 根据您要创建的服务为其部署资源。例如：这里是 Apache 服务中要使用的资源。它们包括一个文件系统（`fs`）资源、一个 IP（`ip`）资源和一个 Apache（`apache`）资源。

```
<rm>
  <resources>
    <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
    <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
    <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
shutdown_wait="0"/>
  </resources>
</rm>
```

例 7.9 “`cluster.conf` 添加了资源的文件” 演示了添加 `resources` 部分的 `cluster.conf` 文件示例。

4. 通过增加其值更新 `config_version` 属性（例如：从 `config_version="2"` 改为 `config_version="3"`）。
5. 保存 `/etc/cluster/cluster.conf`。
6. （可选），运行 `ccs_config_validate` 命令，根据集群方案（`cluster.rng`）验证该文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

7. 运行 `cman_tool version -r` 命令在其它集群节点中传播该配置。
8. 确认传播了更新的文件。
9. 执行 [第 7.5.2 节 “在集群中添加集群服务”](#)。

例 7.9. `cluster.conf` 添加了资源的文件

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
    passwd="password_example"/>
  </fencedevices>
  <rm>
    <failoverdomains>
      <failoverdomain name="example_pri" nofailback="0" ordered="1" restricted="0">
        <failoverdomainnode name="node-01.example.com" priority="1"/>
        <failoverdomainnode name="node-02.example.com" priority="2"/>
        <failoverdomainnode name="node-03.example.com" priority="3"/>
      </failoverdomain>
    </failoverdomains>
    <resources>
      <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
      <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
      <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
      shutdown_wait="0"/>
    </resources>
  </rm>
</cluster>

```

7.5.2. 在集群中添加集群服务

要在集群中添加集群服务，请按照本小节中的步骤执行。

1. 在集群的任意节点中打开 `/etc/cluster/cluster.conf`。
2. 为每个服务在 `rm` 元素中添加 `service` 部分。例如：

```

<rm>
  <service autostart="1" domain="" exclusive="0" name="" recovery="restart">

  </service>
</rm>

```

3. 在 `service` 元素中配置以下参数（属性）：

- `autostart` — 指定是否在集群启动时自动启动该服务。
- `domain` — 指定故障切换域（如果需要）。
- `exclusive` — 指定该服务只在没有其它服务运行的节点中的策略。
- `recovery` — 为该服务指定恢复策略。选项为 `relocate`（重新定位）、`restart`（重启）或者 `disable`（禁用）该服务。

4. 根据您要使用的资源类型使用全局或者具体服务资源部署该服务。

例如：这里是使用全局资源的 Apache 服务：

```
<rm>
  <resources>
    <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
    <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
    <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
shutdown_wait="0"/>
  </resources>
  <service autostart="1" domain="example_pri" exclusive="0" name="example_apache"
recovery="relocate">
    <fs ref="web_fs"/>
    <ip ref="127.143.131.100"/>
    <apache ref="example_server"/>
  </service>
</rm>
```

例如：这里是具体服务资源使用的 Apache 服务：

```
<rm>
  <service autostart="0" domain="example_pri" exclusive="0" name="example_apache2"
recovery="relocate">
    <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www2" fstype="ext3"/>
    <ip address="127.143.131.101" monitor_link="on" sleeptime="10"/>
    <apache config_file="conf/httpd.conf" name="example_server2" server_root="/etc/httpd"
shutdown_wait="0"/>
  </service>
</rm>
```

例 7.10 “添加服务的 `cluster.conf`：一个使用全局资源，一个使用具体服务资源” 演示有两个服务的 `cluster.conf` 文件示例：

- `example_apache` — 这个服务使用全局资源 `web_fs`、`127.143.131.100` 和 `example_server`。
- `example_apache2` — 这个服务使用具体服务资源 `web_fs2`、`127.143.131.101` 和 `example_server2`。

5. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。

6. 保存 `/etc/cluster/cluster.conf`。
7. (可选)，运行 `ccs_config_validate` 命令，确认根据集群方案 (`cluster.rng`) 更新的文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

8. 运行 `cman_tool version -r` 命令在其它集群节点中传播该配置。
9. 确认传播了更新的文件。
10. 执行 第 7.6 节 “确认配置”。

例 7.10. 添加服务的 `cluster.conf`：一个使用全局资源，一个使用具体服务资源

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
  </fencedevices>
  <rm>
    <failoverdomains>
      <failoverdomain name="example_pri" nofailback="0" ordered="1" restricted="0">
        <failoverdomainnode name="node-01.example.com" priority="1"/>
        <failoverdomainnode name="node-02.example.com" priority="2"/>
        <failoverdomainnode name="node-03.example.com" priority="3"/>
      </failoverdomain>
    </failoverdomains>
    <resources>
      <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
      <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
      <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
shutdown_wait="0"/>
    </resources>
    <service autostart="1" domain="example_pri" exclusive="0" name="example_apache" recovery="relocate">
      <fs ref="web_fs"/>
    </service>
  </rm>
</cluster>
```

```

        <ip ref="127.143.131.100"/>
        <apache ref="example_server"/>
    </service>
    <service autostart="0" domain="example_pri" exclusive="0" name="example_apache2" recovery="relocate">
        <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www2" fstype="ext3"/>
        <ip address="127.143.131.101" monitor_link="on" sleeptime="10"/>
        <apache config_file="conf/httpd.conf" name="example_server2" server_root="/etc/httpd"
shutdown_wait="0"/>
    </service>
</rm>
</cluster>

```

7.6. 确认配置

您创建您的集群配置文件后，执行以下步骤确认其正常运行：

1. 在每个节点中重启集群软件。这个动作确保在运行的配置中包括只在启动时检查的配置添加。您可以运行 `service cman restart` 命令重启集群软件。例如：

```

[root@example-01 ~]# service cman restart
Stopping cluster:
  Leaving fence domain...           [ OK ]
  Stopping gfs_controld...          [ OK ]
  Stopping dlm_controld...          [ OK ]
  Stopping fenced...                [ OK ]
  Stopping cman...                  [ OK ]
  Waiting for corosync to shutdown: [ OK ]
  Unloading kernel modules...       [ OK ]
  Unmounting configfs...            [ OK ]
Starting cluster:
  Checking Network Manager...       [ OK ]
  Global setup...                   [ OK ]
  Loading kernel modules...         [ OK ]
  Mounting configfs...              [ OK ]
  Starting cman...                   [ OK ]
  Waiting for quorum...              [ OK ]
  Starting fenced...                [ OK ]
  Starting dlm_controld...          [ OK ]
  Starting gfs_controld...          [ OK ]
  Unfencing self...                 [ OK ]
  Joining fence domain...           [ OK ]

```

2. 如果使用 CLVM 创建集群的卷，则运行 `service clvmd start`。例如：

```

[root@example-01 ~]# service clvmd start
Activating VGs:                     [ OK ]

```

3. 如果您使用 Red Hat GFS2，请运行 `service gfs2 start`。例如：

```

[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB): [ OK ]

```

4. 如果您使用高可用性（HA）服务，请运行 `service rgmanager start`。例如：

```
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager: [ OK ]
```

- 在任意集群节点中运行 `cman_tools nodes` 以确认那些节点作为集群的成员运行（在状态列 "Sts" 中被标记为 "M"）。例如：

```
[root@example-01 ~]# cman_tool nodes
Node Sts Inc Joined Name
  1  M  548  2010-09-28 10:52:21 node-01.example.com
  2  M  548  2010-09-28 10:52:21 node-02.example.com
  3  M  544  2010-09-28 10:52:21 node-03.example.com
```

- 在任意节点中使用 `clustat` 程序确认那些 HA 服务正常运行。另外 `clustat` 可显示集群节点状态。例如：

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

Member Name                ID  Status
-----
node-03.example.com        3 Online, rgmanager
node-02.example.com        2 Online, rgmanager
node-01.example.com        1 Online, Local, rgmanager

Service Name                Owner (Last)                State
-----
service:example_apache      node-01.example.com         started
service:example_apache2     (none)                      disabled
```

- 如果集群正常运行，则您完成了配置文件创建。您可使用命令行工具管理集群，如 [第 8 章 使用命令行工具管理红帽高可用性附加组件](#) 所述。

使用命令行工具管理红帽高可用性附加组件

本章论述了管理红帽高可用性附加组件的各种管理任务，它由以下小节组成：

- 第 8.1 节 “启动和停止集群软件”
- 第 8.2 节 “删除或者添加节点”
- 第 8.3 节 “管理高可用性服务”
- 第 8.4 节 “更新配置”



重要

确定您部署的红帽高可用性附加组件满足您的需要并可支持。部署前联络授权红帽代表确认您的配置。另外预留充分时间测试失败模式。



重要

本章通常使用 `cluster.conf` 元素和属性作为参考。有关 `cluster.conf` 元素和属性的完整列表，请参考 `/usr/share/cluster/cluster.rng` 中的集群方案，注释的方案位于 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。



重要

本章的某些步骤调用 `cman_tool -r` 命令在集群中传播集群配置。使用该命令需要运行 `ricci`。



备注

本章中的步骤可能包括某些命令行工具的具体命令，如 [附录 D，命令行工具小结](#) 所示。有关所有命令和变量的详情请参考每个命令行工具的 `man page`。

8.1. 启动和停止集群软件

您可以如 [第 8.1.1 节 “启动集群软件”](#) 和 [第 8.1.2 节 “停止集群软件”](#) 所述在某个节点中启动或者停止集群软件。在节点中启动集群软件可让该软件加入集群，在节点中停止集群软件则会让该软件离开集群。

8.1.1. 启动集群软件

要在节点中启动集群软件，请按以下顺序输入命令：

1. `service cman start`
2. `service clvmd start`，如果使用 CLVM 创建集群的卷
3. `service gfs2 start`，如果您使用 Red Hat GFS2
4. `service rgmanager start`，如果您使用高可用性（HA）服务（`rgmanager`）。

例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...          [ OK ]
  Global setup...                     [ OK ]
  Loading kernel modules...           [ OK ]
  Mounting configfs...                 [ OK ]
  Starting cman...                     [ OK ]
  Waiting for quorum...                [ OK ]
  Starting fenced...                   [ OK ]
  Starting dlm_control...               [ OK ]
  Starting gfs_control...               [ OK ]
  Unfencing self...                   [ OK ]
  Joining fence domain...              [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                        [ OK ]
Activating VG(s):  2 logical volume(s) in volume group "vg_example" now active
[ OK ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):  [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB):  [ OK ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:      [ OK ]
[root@example-01 ~]#
```

8.1.2. 停止集群软件

要在节点中停止集群软件，请按以下顺序输入命令：

1. `service rgmanager stop`，如果您使用高可用性（HA）服务（`rgmanager`）。
2. `service gfs2 stop`，如果您使用 Red Hat GFS2
3. `umount -at gfs2`，如果您将 `rgmanager` 与 Red Hat GFS2 一同使用，可保证同时卸载了在启动 `rgmanager` 过程中（但不在关机过程中卸载）挂载的所有 GFS2 文件。
4. `service clvmd stop`，如果使用 CLVM 创建集群的卷
5. `service cman stop`

例如：

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:      [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB): [ OK ]
```

```
[root@example-01 ~]# umount -at gfs2
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit [ OK ]
clvmd terminated [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
  Leaving fence domain... [ OK ]
  Stopping gfs_controld... [ OK ]
  Stopping dlm_controld... [ OK ]
  Stopping fenced... [ OK ]
  Stopping cman... [ OK ]
  Waiting for corosync to shutdown: [ OK ]
  Unloading kernel modules... [ OK ]
  Unmounting configfs... [ OK ]
[root@example-01 ~]#
```



备注

在节点中停止集群软件可将其 HA 服务切换到另一个节点中。备选的方法是在停止集群软件前将 HA 服务重新定位或者迁移到另一个节点中。有关管理 HA 服务的详情请参考 [第 8.3 节 “管理高可用性服务”](#)。

8.2. 删除或者添加节点

本章论述了如何在集群中删除节点或者从集群中删除节点。您可以从集群中删除节点，如 [第 8.2.1 节 “从集群中删除节点”](#) 所述，或者在集群中添加节点，如 [第 8.2.2 节 “在集群中添加节点”](#) 所述。

8.2.1. 从集群中删除节点

从集群中删除节点包括在节点中关闭要删除的集群软件并更新集群配置以反应此变化。



重要

如果从集群中删除节点让该集群中只剩下两个节点，则您必须在更新集群配置文件后在每个节点中重启该集群软件。

要从集群中删除节点，请按照以下步骤操作：

1. 在任意节点中使用 `clusvcadm` 程序重新定位、迁移或者停止要从该集群删除的节点中的运行的所有 HA 服务。有关使用 `clusvcadm` 的详情请参考 [第 8.3 节 “管理高可用性服务”](#)。
2. 在您要删除的节点中停止集群软件，如 [第 8.1.2 节 “停止集群软件”](#) 所述。例如：

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager: [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB): [ OK ]
[root@example-01 ~]# service clvmd stop
```

```

Signaling clvmd to exit          [ OK ]
clvmd terminated                 [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
  Leaving fence domain...        [ OK ]
  Stopping gfs_controld...       [ OK ]
  Stopping dlm_controld...       [ OK ]
  Stopping fenced...            [ OK ]
  Stopping cman...               [ OK ]
  Waiting for corosync to shutdown: [ OK ]
  Unloading kernel modules...    [ OK ]
  Unmounting configfs...         [ OK ]
[root@example-01 ~]#

```

3. 在集群的任意节点中编辑 `/etc/cluster/cluster.conf`，删除要删除节点的 `clusternode` 部分。例如：在 例 8.1 “三节点配置” 中，如果要删除 `node-03.example.com`，则要删除该节点的 `clusternode` 部分。如果删除节点让该集群只剩下两个节点，则您可在配置文件中添加以下行以便单一节点可维护仲裁（例如：如果一个节点失败）：

```
<cman two_node="1" expected_votes="1"/>
```

有关三节点配置和双节点配置对比请参考 第 8.2.3 节 “三节点和双节点配置示例”。

4. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。
5. 保存 `/etc/cluster/cluster.conf`。
6. （可选），运行 `ccs_config_validate` 命令，确认根据集群方案（`cluster.rng`）更新的文件。例如：

```

[root@example-01 ~]# ccs_config_validate
Configuration validates

```

7. 运行 `cman_tool version -r` 命令在其它集群节点中传播该配置。
8. 确认传播了更新的文件。
9. 如果集群节点计数从大于 2 变为等于 2，则您必须重启集群软件，如下：
 - a. 在每个节点中如 第 8.1.2 节 “停止集群软件” 所述停止集群软件。例如：

```

[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager: [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB): [ OK ]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit          [ OK ]
clvmd terminated                 [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
  Leaving fence domain...        [ OK ]
  Stopping gfs_controld...       [ OK ]
  Stopping dlm_controld...       [ OK ]
  Stopping fenced...            [ OK ]
  Stopping cman...               [ OK ]
  Waiting for corosync to shutdown: [ OK ]
  Unloading kernel modules...    [ OK ]

```



```
Unmounting configfs... [ OK ]
[root@example-01 ~]#
```

- b. 在每个节点中如 第 8.1.1 节 “启动集群软件” 所述启动集群软件。例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager... [ OK ]
  Global setup... [ OK ]
  Loading kernel modules... [ OK ]
  Mounting configfs... [ OK ]
  Starting cman... [ OK ]
  Waiting for quorum... [ OK ]
  Starting fenced... [ OK ]
  Starting dlm_controld... [ OK ]
  Starting gfs_controld... [ OK ]
  Unfencing self... [ OK ]
  Joining fence domain... [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd: [ OK ]
Activating VG(s): 2 logical volume(s) in volume group "vg_example" now active
[ OK ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB): [ OK ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager: [ OK ]
[root@example-01 ~]#
```

- c. 在任意集群节点中运行 `cman_tools nodes` 以确认那些节点作为集群的成员运行（在状态列 "Sts" 中被标记为 "M"）。例如：

```
[root@example-01 ~]# cman_tool nodes
Node Sts Inc Joined Name
  1 M 548 2010-09-28 10:52:21 node-01.example.com
  2 M 548 2010-09-28 10:52:21 node-02.example.com
```

- d. 在任意节点中使用 `clustat` 程序确认那些 HA 服务正常运行。另外 `clustat` 可显示集群节点状态。例如：

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

Member Name          ID  Status
-----
node-02.example.com   2  Online, rgmanager
node-01.example.com   1  Online, Local, rgmanager

Service Name          Owner (Last)      State
-----
service:example_apache node-01.example.com started
service:example_apache2 (none)            disabled
```

8.2.2. 在集群中添加节点

在集群中添加节点包括更新集群配置；在要添加的节点中使用更新的配置；以及在该节点中启动集群软件。要在集群中添加节点，请执行以下步骤：

1. 在集群的任意节点中编辑 `/etc/cluster/cluster.conf`，为要添加的节点添加 `clusternode` 部分。例如：在 [例 8.2 “双节点配置”](#) 中，如果要添加 `node-03.example.com`，则请为该节点添加 `clusternode` 部分。如果添加节点导致群集从双节点变为三个或者三个以上节点，请在 `/etc/cluster/cluster.conf` 中删除 `cman` 属性：

```
• cman two_node="1"

• expected_votes="1"
```

有关三节点配置和双节点配置对比请参考 [第 8.2.3 节 “三节点和双节点配置示例”](#)。

2. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。
3. 保存 `/etc/cluster/cluster.conf`。
4. （可选），运行 `ccs_config_validate` 命令，确认根据集群方案（`cluster.rng`）更新的文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

5. 运行 `cman_tool version -r` 命令在其它集群节点中传播该配置。
6. 确认传播了更新的文件。
7. 在每个要添加到集群的节点的 `/etc/cluster/` 中使用更新的配置文件。例如：`scp` 命令会将更新的配置文件发送到要添加到集群的每个节点中。
8. 如果节点计数从 2 变为大于 2，则您必须在现有集群节点中重启集群软件，如下：
 - a. 在每个节点中如 [第 8.1.2 节 “停止集群软件”](#) 所述停止集群软件。例如：

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager: [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB): [ OK ]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit [ OK ]
clvmd terminated [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
  Leaving fence domain... [ OK ]
  Stopping gfs_controld... [ OK ]
  Stopping dlm_controld... [ OK ]
  Stopping fenced... [ OK ]
  Stopping cman... [ OK ]
  Waiting for corosync to shutdown: [ OK ]
  Unloading kernel modules... [ OK ]
  Unmounting configs... [ OK ]
[root@example-01 ~]#
```

- b. 在每个节点中如 第 8.1.1 节 “启动集群软件” 所述启动集群软件。例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...          [ OK ]
  Global setup...                      [ OK ]
  Loading kernel modules...            [ OK ]
  Mounting configs...                  [ OK ]
  Starting cman...                      [ OK ]
  Waiting for quorum...                 [ OK ]
  Starting fenced...                    [ OK ]
  Starting dlm_controld...              [ OK ]
  Starting gfs_controld...              [ OK ]
  Unfencing self...                     [ OK ]
  Joining fence domain...               [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                          [ OK ]
Activating VG(s):  2 logical volume(s) in volume group "vg_example" now active
                                                [ OK ]

[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):    [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB):    [ OK ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:        [ OK ]
[root@example-01 ~]#
```

9. 在每个要添加到集群的节点中如 第 8.1.1 节 “启动集群软件” 所述启动集群软件。例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...          [ OK ]
  Global setup...                      [ OK ]
  Loading kernel modules...            [ OK ]
  Mounting configs...                  [ OK ]
  Starting cman...                      [ OK ]
  Waiting for quorum...                 [ OK ]
  Starting fenced...                    [ OK ]
  Starting dlm_controld...              [ OK ]
  Starting gfs_controld...              [ OK ]
  Unfencing self...                     [ OK ]
  Joining fence domain...               [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                          [ OK ]
Activating VG(s):  2 logical volume(s) in volume group "vg_example" now active
                                                [ OK ]

[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):    [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB):    [ OK ]

[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:        [ OK ]
[root@example-01 ~]#
```

10. 在任意节点中使用 `clustat` 程序确认每个添加的节点正作为集群的一部分运行。例如：

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate
```

Member Name	ID	Status
node-03.example.com	3	Online, rgmanager
node-02.example.com	2	Online, rgmanager
node-01.example.com	1	Online, Local, rgmanager
Service Name	Owner (Last)	State
service:example_apache	node-01.example.com	started
service:example_apache2	(none)	disabled

有关使用 `clustat` 的详情请参考 [第 8.3 节 “管理高可用性服务”](#)。

另外，您可以使用 `cman_tool status` 确认节点投票、节点计数以及仲裁计数。例如：

```
[root@example-01 ~]#cman_tool status
Version: 6.2.0
Config Version: 19
Cluster Name: mycluster
Cluster Id: 3794
Cluster Member: Yes
Cluster Generation: 548
Membership state: Cluster-Member
Nodes: 3
Expected votes: 3
Total votes: 3
Node votes: 1
Quorum: 2
Active subsystems: 9
Flags:
Ports Bound: 0 11 177
Node name: node-01.example.com
Node ID: 3
Multicast addresses: 239.192.14.224
Node addresses: 10.15.90.58
```

11. 在任意节点中您可以使用 `clusvcadm` 程序将运行中的服务重新定位或者迁移到新添加的节点中。另外，您可以启用所有禁用的服务。有关使用 `clusvcadm` 详情请参考 [第 8.3 节 “管理高可用性服务”](#)。

8.2.3. 三节点和双节点配置示例

以下是三节点和双节点配置示例对比。

例 8.1. 三节点配置

```
<cluster name="mycluster" config_version="3">
  <cman/>
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
```

```

        <device name="apc" port="2"/>
    </method>
</fence>
</clusternode>
<clusternode name="node-03.example.com" nodeid="3">
    <fence>
        <method name="APC">
            <device name="apc" port="3"/>
        </method>
    </fence>
</clusternode>
</clusternodes>
<fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
</fencedevices>
<rm>
    <failoverdomains>
        <failoverdomain name="example_pri" nofailback="0" ordered="1" restricted="0">
            <failoverdomainnode name="node-01.example.com" priority="1"/>
            <failoverdomainnode name="node-02.example.com" priority="2"/>
            <failoverdomainnode name="node-03.example.com" priority="3"/>
        </failoverdomain>
    </failoverdomains>
    <resources>
        <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
        <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
        <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
shutdown_wait="0"/>
    </resources>
    <service autostart="0" domain="example_pri" exclusive="0" name="example_apache" recovery="relocate">
        <fs ref="web_fs"/>
        <ip ref="127.143.131.100"/>
        <apache ref="example_server"/>
    </service>
    <service autostart="0" domain="example_pri" exclusive="0" name="example_apache2" recovery="relocate">
        <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www" fstype="ext3"/>
        <ip address="127.143.131.101" monitor_link="on" sleeptime="10"/>
        <apache config_file="conf/httpd.conf" name="example_server2" server_root="/etc/httpd"
shutdown_wait="0"/>
    </service>
</rm>
</cluster>

```

例 8.2. 双节点配置

```

<cluster name="mycluster" config_version="3">
    <cman two_node="1" expected_votes="1"/>
    <clusternodes>
        <clusternode name="node-01.example.com" nodeid="1">
            <fence>
                <method name="APC">
                    <device name="apc" port="1"/>
                </method>
            </fence>
        </clusternode>
        <clusternode name="node-02.example.com" nodeid="2">
            <fence>
                <method name="APC">
                    <device name="apc" port="2"/>
                </method>
            </fence>
        </clusternode>
    </clusternodes>

```

```

</clusternodes>
<fencedevices>
  <fencedevice agent="fence_apc" ipaddr="apc_ip_example" login="login_example" name="apc"
passwd="password_example"/>
</fencedevices>
<rm>
  <failoverdomains>
    <failoverdomain name="example_pri" nofailback="0" ordered="1" restricted="0">
      <failoverdomainnode name="node-01.example.com" priority="1"/>
      <failoverdomainnode name="node-02.example.com" priority="2"/>
    </failoverdomain>
  </failoverdomains>
  <resources>
    <fs name="web_fs" device="/dev/sdd2" mountpoint="/var/www" fstype="ext3"/>
    <ip address="127.143.131.100" monitor_link="on" sleeptime="10"/>
    <apache config_file="conf/httpd.conf" name="example_server" server_root="/etc/httpd"
shutdown_wait="0"/>
  </resources>
  <service autostart="0" domain="example_pri" exclusive="0" name="example_apache" recovery="relocate">
    <fs ref="web_fs"/>
    <ip ref="127.143.131.100"/>
    <apache ref="example_server"/>
  </service>
  <service autostart="0" domain="example_pri" exclusive="0" name="example_apache2" recovery="relocate">
    <fs name="web_fs2" device="/dev/sdd3" mountpoint="/var/www" fstype="ext3"/>
    <ip address="127.143.131.101" monitor_link="on" sleeptime="10"/>
    <apache config_file="conf/httpd.conf" name="example_server2" server_root="/etc/httpd"
shutdown_wait="0"/>
  </service>
</rm>
</cluster>

```

8.3. 管理高可用性服务

您可以使用 Cluster Status Utility `clustat` 和 Cluster User Service Administration Utility `clusvcadm` 管理高可用性服务。`clustat` 显示集群的状态，`clusvcadm` 提供管理高可用性服务的工具。

本小节提供有关使用 `clustat` 和 `clusvcadm` 管理 HA 服务的信息，它包括以下部分：

- 第 8.3.1 节 “使用 `clustat` 显示 HA 服务”
- 第 8.3.2 节 “使用 `clusvcadm` 管理 HA 服务”

8.3.1. 使用 `clustat` 显示 HA 服务

`clustat` 显示集群状态。它为您提供成员信息、仲裁查看、所有高可用性服务的状态，并提示运行 `clustat` 命令的节点（本地）。表 8.1 “服务状态” 描述运行 `clustat` 是会出现并显示的服务。例 8.3 “`clustat` 显示” 给出 `clustat` 显示示例。有关运行 `clustat` 命令的详情请参考 `clustat man page`。

表 8.1. 服务状态

服务状态	描述
「Started」	该服务资源已被配置并可用于拥有该服务的集群系统。
「Recovering」	该服务正在另一个节点中等待启动。
「Disabled」	该服务已经被禁用，且没有分配拥有者。该集群永远不能自动重启禁用的服务。

服务状态	描述
「Stopped」	在停止状态中，将评价该服务以便在下一个服务或者节点过渡后启动。这是一个临时状态。您可以在这个状态中禁用或者启用该服务。
「Failed」	假设该服务已死。无论何时，当资源的 stop 操作失败时，服务都会处于这个状态。服务处于这个状态后，您在发出 disable 请求前必须确认没有为其分配任何资源（例如挂载的文件系统）。当某个服务处于这种状态是，唯一可行的操作就是 disable。
「Uninitialized」	在启动和运行 clustat -f 的过程中，有些时候可以出现这个状态。

例 8.3. clustat 显示

```
[root@example-01 ~]#clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:15 2010
Member Status: Quorate

Member Name                                ID  Status
-----
node-03.example.com                        3 Online, rgmanager
node-02.example.com                        2 Online, rgmanager
node-01.example.com                        1 Online, Local, rgmanager

Service Name                                Owner (Last)                                State
-----
service:example_apache                     node-01.example.com                        started
service:example_apache2                     (none)                                    disabled
```

8.3.2. 使用 clusvcadm 管理 HA 服务

您可以使用 clusvcadm 命令管理 HA 服务。使用它您可以执行以下操作:

- 启用并启动服务。
- 禁用服务。
- 停止服务。
- 冻结服务
- 解冻服务
- 迁移服务（只用于虚拟机服务）
- 重新定位服务。
- 重启服务。

表 8.2 “服务操作” 详细描述了这些操作。有关如何执行那些操作的完整论述请参考 clusvcadm 程序 man page。

表 8.2. 服务操作

服务操作	描述	命令语法
「启用」	有条件地在首选对象中并根据故障切换域规则启动服务。二者缺一，则运行 clusvcadm 的本地主机将会启动该服务。	clusvcadm -e <service_name> 或者 clusvcadm -e <service_name> -m

服务操作	描述	命令语法
	如果原始启动失败，则该服务的行为会类似重新定位请求（请参考本表格中的「重新定位」）。如果操作成功，则该服务会处于启动的状态。	<member>（使用 -m 选项指定要启动该服务的首选目标成员。）
「禁用」	停止该服务使其处于禁用状态。当某个服务处于失败状态时，这是唯一允许的操作。	clusvcadm -d <service_name>
「重新定位」	将该服务移动到另一个节点中。您可以任选首选节点接受此服务，但如果在那个主机中无法运行该服务（例如：如果服务无法启动或者主机离线），则无法阻止重新定位，并选择另一个节点。rgmanager 试图启动该服务，重新定位失败并尝试在原始拥有者中启动该服务。如果原始拥有者无法重启该服务，则该服务会处于停止的状态。	clusvcadm -r <service_name> 或者 clusvcadm -r <service_name> -m <member>（使用 -m 选项指定要启动该服务的首选目标成员。）
「停止」	停止该服务使其处于停止状态。	clusvcadm -s <service_name>
「冻结」	在运行某个服务的节点中冻结该服务。这会阻止服务状态检查以及节点失败事件或者 rgmanager 停止时的故障切换。这可用于挂起服务以便在基础资源维护。有关使用冻结和解冻操作的重要信息请参考 “使用冻结和解冻操作的注意事项” 一节。	clusvcadm -Z <service_name>
「解冻」	解冻会使服务脱离冻结状态。这会重新启用状态检查。有关使用冻结和解冻操作的重要信息请参考 “使用冻结和解冻操作的注意事项” 一节。	clusvcadm -U <service_name>
「迁移」	将虚拟机迁移到另一个节点中。您必须指定目标节点。根据失败的情况，迁移失败可能导致虚拟机处于 failed 状态，或者在原始拥有者中处于 started 的状态。	clusvcadm -M <service_name> -m <member>  重要 在迁移操作中您必须使用 -m <member> 选项指定目标节点。
「重启」	在当前运行该服务的节点中重启服务。	clusvcadm -R <service_name>

使用冻结和解冻操作的注意事项

使用冻结操作可维护部分 rgmanager 服务。例如：如果您有一个数据库和一个网页服务器使用 rgmanager 服务，您可能要冻结 rgmanager 服务，停止数据库，执行维护，重启数据库，并解冻该服务。

当冻结某个服务时，它会有以下动作：

- 禁用状态检查。
- 禁用启动操作。
- 禁用停止操作。
- 不会执行故障切换（即使关闭该服务的拥有者）。



重要

不按照以下步骤执行将导致将在多个主机中重新分配资源:

- 在冻结服务时您一定不能停止 rgmanager 的所有事务，除非您要在重启 rgmanager 前重启该主机。
- 在报告的拥有者重新加入集群并重启 rgmanager 前您一定不能解冻该服务。

8.4. 更新配置

更新集群配置包括编辑集群配置文件（/etc/cluster/cluster.conf）并在集群的每个节点中使用该文件。您可以使用以下方法之一更新配置:

- [第 8.4.1 节 “使用 cman_tool version -r 更新配置”](#)
- [第 8.4.2 节 “使用 scp 更新配置”](#)

8.4.1. 使用 cman_tool version -r 更新配置

要使用 cman_tool version -r 命令更新配置，请执行以下步骤:

1. 在集群的任意节点中编辑 /etc/cluster/cluster.conf 文件。
2. 增加 config_version 属性参数即可更新该参数（例如：从 config_version="2" 改为 config_version="3">）。
3. 保存 /etc/cluster/cluster.conf。
4. 运行 cman_tool version -r 命令在其它集群节点中传播该配置。需要在每个集群节点中都运行 ricci 方可传播更新的集群配置信息。
5. 确认传播了更新的文件。
6. 如果您只更改以下配置，则您可以跳过这一步（重启集群软件）：
 - 从集群配置中删除节点 — 除非节点计数从大于 2 到等于 2。有关从集群中删除节点，并从两个以上节点变为两个节点的详情请参考 [第 8.2 节 “删除或者添加节点”](#)。
 - 在集群配置中添加节点 — 除非节点计数从 2 到大于 2。有关在集群中添加节点且从两个节点增加到两个以上节点的详情请参考 [第 8.2.2 节 “在集群中添加节点”](#)。
 - 更改记录守护进程信息的方法。
 - HA 服务/VM 维护（添加、编辑或者删除）。

- 资源维护（添加、编辑或者删除）。
- 故障切换域维护（添加、编辑或者删除）。

否则，您必须重启该集群软件，如下：

- a. 在每个节点中如 [第 8.1.2 节 “停止集群软件”](#) 所述停止集群软件。例如：

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager:           [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA):      [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB):      [ OK ]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit                      [ OK ]
clvmd terminated                             [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
  Leaving fence domain...                     [ OK ]
  Stopping gfs_controld...                    [ OK ]
  Stopping dlm_controld...                    [ OK ]
  Stopping fenced...                          [ OK ]
  Stopping cman...                            [ OK ]
  Waiting for corosync to shutdown:           [ OK ]
  Unloading kernel modules...                 [ OK ]
  Unmounting configfs...                      [ OK ]
[root@example-01 ~]#
```

- b. 在每个节点中如 [第 8.1.1 节 “启动集群软件”](#) 所述启动集群软件。例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...                 [ OK ]
  Global setup...                             [ OK ]
  Loading kernel modules...                   [ OK ]
  Mounting configfs...                        [ OK ]
  Starting cman...                            [ OK ]
  Waiting for quorum...                       [ OK ]
  Starting fenced...                          [ OK ]
  Starting dlm_controld...                    [ OK ]
  Starting gfs_controld...                    [ OK ]
  Unfencing self...                           [ OK ]
  Joining fence domain...                     [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                               [ OK ]
Activating VG(s):  2 logical volume(s) in volume group "vg_example" now active
[ OK ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):        [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB):        [ OK ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:             [ OK ]
[root@example-01 ~]#
```

停止和启动集群软件可确保所有只在启动时检查的配置更改都包含在运行的配置中。

7. 在任意集群节点中运行 `cman_tools nodes` 以确认那些节点作为集群的成员运行（在状态列 "Sts" 中被标记为 "M"）。例如：

```
[root@example-01 ~]# cman_tool nodes
Node Sts Inc Joined Name
1 M 548 2010-09-28 10:52:21 node-01.example.com
2 M 548 2010-09-28 10:52:21 node-02.example.com
3 M 544 2010-09-28 10:52:21 node-03.example.com
```

8. 在任意节点中使用 `clustat` 程序确认那些 HA 服务正常运行。另外 `clustat` 可显示集群节点状态。例如：

```
[root@example-01 ~]# clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

Member Name                      ID Status
-----
node-03.example.com              3 Online, rgmanager
node-02.example.com              2 Online, rgmanager
node-01.example.com              1 Online, Local, rgmanager

Service Name                      Owner (Last) State
-----
service:example_apache            node-01.example.com started
service:example_apache2           (none) disabled
```

9. 如果集群如预期运行，则您完成了配置更新。

8.4.2. 使用 scp 更新配置

要使用 `scp` 命令更新配置，请执行以下步骤：

1. 在每个节点中如 第 8.1.2 节 “停止集群软件” 所述停止集群软件。例如：

```
[root@example-01 ~]# service rgmanager stop
Stopping Cluster Service Manager: [ OK ]
[root@example-01 ~]# service gfs2 stop
Unmounting GFS2 filesystem (/mnt/gfsA): [ OK ]
Unmounting GFS2 filesystem (/mnt/gfsB): [ OK ]
[root@example-01 ~]# service clvmd stop
Signaling clvmd to exit [ OK ]
clvmd terminated [ OK ]
[root@example-01 ~]# service cman stop
Stopping cluster:
Leaving fence domain... [ OK ]
Stopping gfs_controld... [ OK ]
Stopping dlm_controld... [ OK ]
Stopping fenced... [ OK ]
Stopping cman... [ OK ]
Waiting for corosync to shutdown: [ OK ]
Unloading kernel modules... [ OK ]
Unmounting configfs... [ OK ]
[root@example-01 ~]#
```

2. 在集群的任意节点中编辑 `/etc/cluster/cluster.conf` 文件。
3. 增加 `config_version` 属性参数即可更新该参数（例如：从 `config_version="2"` 改为 `config_version="3">`）。

4. 保存 `/etc/cluster/cluster.conf`。
5. 运行 `ccs_config_validate` 命令，根据集群方案 (`cluster.rng`) 确认更新的文件。例如：

```
[root@example-01 ~]# ccs_config_validate
Configuration validates
```

6. 如果更新的文件有效，则请使用 `scp` 将其部署到每个集群节点的 `/etc/cluster/` 文件中。
7. 确认传播了更新的文件。
8. 在每个节点中如 第 8.1.1 节 “启动集群软件” 所述启动集群软件。例如：

```
[root@example-01 ~]# service cman start
Starting cluster:
  Checking Network Manager...          [ OK ]
  Global setup...                      [ OK ]
  Loading kernel modules...            [ OK ]
  Mounting configfs...                  [ OK ]
  Starting cman...                      [ OK ]
  Waiting for quorum...                 [ OK ]
  Starting fenced...                    [ OK ]
  Starting dlm_control...                [ OK ]
  Starting gfs_control...                [ OK ]
  Unfencing self...                     [ OK ]
  Joining fence domain...               [ OK ]
[root@example-01 ~]# service clvmd start
Starting clvmd:                          [ OK ]
Activating VG(s):  2 logical volume(s) in volume group "vg_example" now active
                                                [ OK ]
[root@example-01 ~]# service gfs2 start
Mounting GFS2 filesystem (/mnt/gfsA):    [ OK ]
Mounting GFS2 filesystem (/mnt/gfsB):    [ OK ]
[root@example-01 ~]# service rgmanager start
Starting Cluster Service Manager:        [ OK ]
[root@example-01 ~]#
```

9. 在任意集群节点中运行 `cman_tools nodes` 以确认那些节点作为集群的成员运行（在状态列 "Sts" 中被标记为 "M"）。例如：

```
[root@example-01 ~]# cman_tool nodes
Node Sts  Inc   Joined           Name
  1   M    548   2010-09-28 10:52:21 node-01.example.com
  2   M    548   2010-09-28 10:52:21 node-02.example.com
  3   M    544   2010-09-28 10:52:21 node-03.example.com
```

10. 在任意节点中使用 `clustat` 程序确认那些 HA 服务正常运行。另外 `clustat` 可显示集群节点状态。例如：

```
[root@example-01 ~]# clustat
Cluster Status for mycluster @ Wed Nov 17 05:40:00 2010
Member Status: Quorate

Member Name                      ID   Status
-----
node-03.example.com              3   Online, rgmanager
node-02.example.com              2   Online, rgmanager
```

```
node-01.example.com          1 Online, Local, rgmanager

Service Name                  Owner (Last)                  State
-----
service:example_apache       node-01.example.com          started
service:example_apache2      (none)                       disabled
```

11. 如果集群如预期运行，则您完成了配置更新。

诊断并修正集群中的问题

集群问题的故障配置通常比较困难。这是因为由于系统集群产生的负载性的增大使之与诊断单一系统的问题完全不同。但是有一些问题是管理员在部署或者管理集群时会经常遇到的。了解如何处理那些常见问题可对部署和管理集群有所帮助。

本章提供有关集群的常见问题以及如何对其进行故障排除。您可以在我们的知识库中获得更多信息，也可联络授权红帽支持代表寻求帮助。如果您的问题于 GFS2 文件系统有关，您可以在《全局文件系统 2: 配置和管理》一书中找到故障排除常见 GFS2 问题的信息。

9.1. 没有形成集群

如果您无法形成新的集群，请检查以下方面：

- 确定正确设置了名称解析。`cluster.conf` 文件中的集群节点名称应该与用来解析该集群通过网络进行沟通的集群地址的名称对应。例如：如果您的集群节点名为 `nodea` 和 `nodeb`，请确定在 `/etc/cluster/cluster.conf` 文件中有这两个节点的条目，且 `/etc/hosts` 文件与那些名称匹配。
- 因为集群使用多播在节点间进行沟通，请确定多播流量没有被阻断、延迟，或者被该集群用来沟通的网络影响。请注意：有些 Cisco 开关的功能可能导致多播流量延迟。
- 使用 `telnet` 或者 `SSH` 确认您是否可连接到远程节点。
- 执行 `ethtool eth1 | grep link` 命令检查该以太网链接是否可用。
- 在每个节点中使用 `tcpdump` 命令坚持网络流量。
- 确定您没有设定防火墙规则阻断节点间的沟通。
- 请确定您忽略集群流量的接口不使用任何 0 以外的绑定模式，也不使用 VLAN 标记。

9.2. 无法在 Fence 或者重启后重新加入集群的节点

如果您从节点在 fence 或者重启后无法重新加入该集群，请检查以下方面：

- 使用 Cisco Catalyst 切换通过其流量的集群可能会有这个问题。
- 请确定所有集群节点都使用同一版本的 `cluster.conf` 文件。如果在任何一个节点中的 `cluster.conf` 文件有所不同，则那些节点在 fence 后就无法加入该集群。

从红帽企业版 Linux 6.1 发行本开始，您可以使用以下命令确认在主机的集群配置文件中指定的所有节点都有相同的集群配置文件：

```
ccs -h host --checkconf
```

有关 `ccs` 命令的详情请参考 [第 5 章 使用 ccs 命令配置红帽高可用性附加组件](#) 和 [第 6 章 使用 ccs 管理 Red Hat 高可用性附加组件](#)。

- 请确定您要在要加入该集群的节点中为集群服务配置了 `chkconfig on`。
- 请确定没有阻断该节点与集群中的其它节点沟通的防火墙规则。

9.3. 集群服务挂起

当集群服务尝试 fence 某个节点时，该集群服务会停止直到成功完成 fence 操作。因此，如果您使用集群控制的存储或者服务挂起，且集群节点显示不同的集群成员；或者当您尝试 fence 某个节点时集群挂起，您需要重启几点进行恢复时，请检查以下方面：

- 该集群可能尝试 fence 某个节点，且 fence 操作可能已经失败。
- 查看所有节点中的 `/var/log/messages` 文件，看看是否有失败的 fence 信息。如果有，重启集群中的那些节点，并正确配置 fencing。
- 确认没有出现如 第 9.6 节 “双节点集群的每个节点都报告第二个节点无法工作” 所示的网络分区。同时确认节点间可进行沟通，网络正常工作。
- 如果有节点离开该集群，剩余的节点可能不足构成集群。集群需要有一定量的节点方可操作。如果删除节点导致该节点没有足够量的节点，则服务和存储将会挂起。您可以调整预期的票数或者在该集群中保持所需节点数。



注意

您可以使用 `fence_node` 命令或者 Conga 手动 fence 某个节点。详情请查看 第 4.2.2 节 “使节点离开或者加入集群”。

9.4. 无法启动集群服务

如果无法启动某个集群控制的服务，请检查以下方面。

- 可能在 `cluster.conf` 文件的服务配置部分有语法错误。您可以使用 `rg_test` 命令验证您配置文件中的语法。如果有任何配置或者语法错误，`rg_test` 会告诉您哪里出了问题。

```
$ rg_test test /etc/cluster/cluster.conf start service servicename
```

有关 `rg_test` 命令的详情请参考 第 C.5 节 “调整并测试服务和资源顺序”。

如果配置无误，可提高资源组管理器的日志级别，然后阅读日志信息确定是什么导致无法启动该服务。您可以在 `cluster.conf` 文件的 `rm` 标签中添加 `loglevel="7"` 参数提高日志级别。然后您可以根据启动、停止、迁移集群的服务增加信息日志的详细程度。

9.5. 无法迁移集群控制的服务

如果无法将集群控制的服务转移到另一个节点，但可在某些具体节点中启动，请检查以下方面。

- 请确定该集群的所有需要运行给定服务的节点中都有运行那个服务需要的资源。例如：如果您的集群服务假设某个脚本文件位于某具体位置，或者某个文件系统挂载于某个具体挂载点，那么您必须确定那些资源位于该集群的所有节点的预期位置。
- 请确定没有将故障切换域、服务相依性以及服务专有权配置为您无法将服务迁移到您需要的节点中。
- 如果所需服务是一个虚拟机资源，请查看文档意确定完成了所有正确配置工作。
- 如 第 9.4 节 “无法启动集群服务” 所述提高资源组管理器日志级别，然后阅读信息日志以确定是什么导致无法迁移该服务。

9.6. 双节点集群的每个节点都报告第二个节点无法工作

如果您的集群是一个双节点集群，且每个节点都报告它可工作，但另一个无法工作，这说明您的集群节点无法通过集群心跳网络（heartbeat network）的多播进行沟通。我们称之为“脑裂（split brain）”或者“网络分区（network partition）”。要解决这个问题，请检查 [第 9.1 节 “没有形成集群”](#) 中所述的情况。

9.7. 在 LUN 路径失败中 Fence 的节点

如果无论何时您有 LUN 路径失败时，您集群中的一个或者多个节点被 fence，这可能是由于比多路径的存储优先使用仲裁磁盘的结果。如果您使用仲裁磁盘，且您的仲裁磁盘优先于多路径存储，请确定您正确配置了所有承受路径失败的时限。

9.8. 仲裁磁盘不作为集群成员出现

如果您将系统配置为使用仲裁磁盘，但该仲裁磁盘没有作为您集群的成员出现，请检查以下情况。

- 请确定您为 qdisk 服务设定了 chkconfig on。
- 请确定您启动了 qdisk 服务。
- 请注意：在该集群中注册仲裁磁盘可能需要几分钟，这很正常。

9.9. 异常故障切换行为

集群服务器的常见问题之一是异常故障切换行为。当启动其它服务时服务会停止，或者服务在故障切换时拒绝启动。这可能是由于故障切换域、服务相依性以及服务排他性造成故障切换系统比较复杂。尝试使用简单一些的服务或者故障切换域配置，看看问题是否还存在。避免使用类似服务排他性和相依性功能，除非您完全掌握这些功能在所有情况下对故障切换造成的影响。

9.10. 随机发生 fencing

如果您发现某个节点随机发生 fence，请检查以下情况。

- 造成 fence 的根本原因总是因为某个节点丢失 token，就是说它无法与集群中的其它节点沟通，并停止返回心跳。
- 任何情况下如果某个系统不在指定的 token 间隔中返回心跳都会导致出现 fence。默认情况下 token 间隔为 10 秒。您可以在 cluster.conf 文件 totem 标签的 token 参数中以毫秒为单位将其指定为所需值（例如：设定为 totem token="30000" 即为 30 秒）。
- 请确定网络可正常工作。
- 请确定集群没有在内部节点沟通的界面中使用奇怪的绑定模式和 VLAN 标记。
- 设法决定该系统是否为 "freezing" 或者内核 panicking。设置 kdump 程序看看您是否可以从这些 fence 之一中得到一个 core。
- 确定不会出现错误地归咎于某个 fence 的情况，例如因为某个存储失败弹出仲裁磁盘，或者由于一些外部因素造成类似 Oracle RAC 的第三方产品重启某个节点。这些信息日志通常对确定这样的问题非常有帮助。无论何时 fence 节点或者重启节点，标准的操作都应该是查看从发生重启/ fence 时起该集群中所有节点的信息日志。
- 对于那些可导致系统不按照预期相应心跳的硬件错误应该全面检查该系统。

使用红帽高可用性附加组件进行 SNMP 配置

从红帽企业版 Linux 6.1 发行本开始，红帽高可用性附加组件支持 SNMP 陷阱。本章论述了如何为 SNMP 配置您的系统，并附带红帽高可用性附加组件为具体集群事件使用的陷阱总结。

10.1. SNMP 和红帽高可用性附加组件

红帽高可用性附加组件 SNMP 的子代理为 foghorn，它可发出 SNMP 陷阱。foghorn 代理与 snmpd 守护进程通过 AgentX 协议进行对话。foghorn 代理知生成 SNMP 陷阱，不支持其它 SNMP 操作，比如 get 或者 set。

目前 foghorn 子代理没有 config 选项。无法将其配置为使用具体插槽，目前只支持默认的 AgentX 插槽。

10.2. 使用红帽高可用性附加组件配置 SNMP

要使用红帽高可用性附加组件配置 SNMP，请在该集群的每个节点中执行以下步骤，保证启用并运行了所需服务。

1. 要在红帽高可用性附加组件中使用 SNMP 陷阱，则要求将 snmpd 作为主代理运行。因为 foghorn 服务是子代理，且使用 AgentX 协议，您必须在 /etc/snmp/snmpd.conf 文件中添加以下行启用 AgentX 支持：

```
master agentx
```

2. 要指定发送 SNMP 陷阱通知的主机，请在 /etc/snmp/snmpd.conf 文件中添加以下行：

```
trap2sink host
```

有关通知处理的详情请参考 snmpd.conf man page。

3. 执行以下命令确定启用并运行了 snmpd 守护进程：

```
% chkconfig snmpd on
% service snmpd start
```

4. 如果没有启用并运行 messagebus，请执行以下命令：

```
% chkconfig messagebus on
% service messagebus start
```

5. 执行以下命令确定启用并运行 foghorn 守护进程：

```
% chkconfig foghorn on
% service foghorn start
```

6. 执行以下命令配置您的系统以便 COROSYNC-MIB 生成 SNMP 陷阱并确定启用并运行 `corosync-notifyd` 守护进程:

```
$ echo "OPTIONS=\"-d\" " > /etc/sysconfig/corosync-notifyd
$ chkconfig corosync-notifyd on
$ service corosync-notifyd start
```

您在集群的每个节点中配置 SNMP 并确定所需服务都在运行后, `foghorn` 服务会受到 D-bus 信号, 并将其转换为 SNMPv2 陷阱。会将这些陷阱发送到使用 `trapsink` 条目定义的主机中以便接收 SNMPv2 陷阱。

10.3. 转发 SNMP 陷阱

可以将 SNMP 陷阱转发到不是该集群一部分的机器中, 您可在外部机器中使用 `snmptrapd` 守护进程并自定义如何响应通知。

执行以下步骤在集群中将 SNMP 陷阱转发到不是该集群节点的机器中:

1. 在该集群的每个节点中如 [第 10.2 节 “使用红帽高可用性附加组件配置 SNMP”](#) 所述执行以下步骤在 `/etc/snmp/snmpd.conf` 文件中设置 `trap2sink host` 条目来指定将要运行 `snmptrapd` 守护进程的外部主机。
2. 在将要接收陷阱的外部主机中编辑 `/etc/snmp/snmptrapd.conf` 配置文件指定您的社区字符串。例如: 使用以下条目可让 `snmptrapd` 守护进程使用 `public` 社区字符串处理通知。

```
authCommunity log.execute.net public
```

3. 执行以下命令在接收陷阱的外部主机中确定启用并正在运行 `snmptrapd` 守护进程:

```
% chkconfig snmptrapd on
% service snmptrapd start
```

有关处理 SNMP 通知的详情请参考 `snmptrapd.conf man page`。

10.4. 红帽高可用性附加组件产生的 SNMP 陷阱

`foghorn` 守护进程生成以下陷阱:

- `fenceNotifyFenceNode`

在被 `fence` 的节点尝试 `fence` 另一个节点时会出现这个陷阱。请注意: 只会在一个节点中生成这个陷阱 -- 即在尝试执行 `fence` 操作的节点中生成。这个通知包括以下字段:

- `fenceNodeName` -- 被 `fence` 的节点名称
- `fenceNodeID` -- 被 `fence` 的节点 id
- `fenceResult` -- `fence` 操作结果 (0 表示成功, -1 表示有问题, -2 表示没有定义 `fencing` 方法)
- `rgmanagerServiceStateChange`

集群服务状态更改时会出现这个陷阱。该通知包括以下字段:

- `rgmanagerServiceName` -- 该服务名称，其中包括服务类型（例如：`service:foo` 或者 `vm:foo`）。
- `rgmanagerServiceState` -- 该服务的状态。这包括过渡状态，比如 `starting` 和 `stopping`，以减小陷阱中的杂乱程度。
- `rgmanagerServiceFlags` -- 服务标签。以下是目前支持的两个标签：`frozen` 表示已经使用 `clusvcadm -Z` 冻结服务；`partial` 表示在该服务中将失败的资源标记为 `non-critical`，这样可在该资源失败并手动重启其组件时不影响整个服务。
- `rgmanagerServiceCurrentOwner` -- 服务拥有者。如果该服务没有运行，则该字段为 `(none)`。
- `rgmanagerServicePreviousOwner` -- 如果知道，则列出最后的服务拥有者。如果不知道最后的拥有者，该字段会显示 `(none)`。

`corosync-nodifyd` 守护进程生成以下陷阱：

- `corosyncNoticesNodeStatus`

当节点加入或者离开集群时会出现这个陷阱。该通知包括以下字段：

- `corosyncObjectsNodeName` -- 节点名称
- `corosyncObjectsNodeID` -- 节点 id
- `corosyncObjectsNodeAddress` -- 节点 IP 地址
- `corosyncObjectsNodeStatus` -- 节点状态 (`joined` 或者 `left`)

- `corosyncNoticesQuorumStatus`

仲裁状态更改时会出现这个陷阱。该通知包括以下字段：

- `corosyncObjectsNodeName` -- 节点名称
- `corosyncObjectsNodeID` -- 节点 id
- `corosyncObjectsQuorumStatus` -- 仲裁的新状态 (`quorate` 或者 `NOT quorate`)

- `corosyncNoticesAppStatus`

客户端程序连接到 Corosync 或者与之断开连接时会出现这个陷阱。

- `corosyncObjectsNodeName` -- 节点名称
- `corosyncObjectsNodeID` -- 节点 id
- `corosyncObjectsAppName` -- 应用程序名称
- `corosyncObjectsAppStatus` -- 该应用程序的新状态 (`connected` 或者 `disconnected`)

附录 A. Fence 设备参数

本附录提供 fence 设备及其参数描述表，同时还提供那些设备的 fence 代理名称。



备注

使用 fence 设备的「Name」参数为红帽高可用性附加组件使用的设备指定任意名称。这与该设备的 DNS 名称不同。



备注

某些 fence 设备有可选的「Password Script」参数。「Password Script」参数可让您使用脚本而不是「Password」参数提供 fence 设备密码。使用「Password Script」参数可取代「Password」参数，允许在集群配置文件（/etc/cluster/cluster.conf）不显示密码。

表 A.1. APC 电源开关 (telnet/SSH)

字段	描述
Name	连接到集群的 APC 设备名称，在该集群中通过 telnet/ssh 记录 fence 守护进程。
IP Address	为该设备分配的 IP 地址或者主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
Switch (可选)	当您使用多个菊花链 (daisy-chained) 开关时连接到该节点的 APC 开关的开关数。
Use SSH	表示系统将使用 SSH 访问该设备。
Path to the SSH identity file	SSH 的识别文件。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_apc	APC 通过 telnet/SSH 的 fence 代理。

表 A.2. Brocade 光纤开关

字段	描述
Name	连接到该集群的 Brocade 设备名称。
IP Address	为该设备分配的 IP 地址。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。

字段	描述
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	开关插座数。
fence_brocade	Brocade FC 开关的 fence 代理。

表 A.3. 使用 SNMP 的 APC 电源开关

字段	描述
Name	连接到集群的 APC 设备名称，在该集群中您通过 SNMP 协议记录 fence 守护进程。
IP Address	为该设备分配的 IP 地址或者主机名。
UDP/TCP port	用来与该设备连接的 UDP/TCP 端口，默认值为 161。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
Switch (可选)	当您使用多个菊花链 (daisy-chained) 开关时连接到该节点的 APC 开关的开关数。
SNMP version	要使用的 SNMP 版本 (1, 2c, 3)，默认值为 1。
SNMP community	SNMP 社区字符串。默认值为 private。
SNMP security level	SNMP 安全等级 (noAuthNoPriv、authNoPriv、authPriv)。
SNMP authentication protocol	SNMP 认证协议 (MD5、SHA)。
SNMP privacy protocol	SNMP 隐私协议 (DES、AES)
SNMP privacy protocol password	SNMP 隐私协议密码。
SNMP privacy protocol script	该脚本为 SNMP 隐私协议提供密码。使用这个参数替代「SNMP privacy protocol password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_apc_snmp	使用 SNMP 协议记录 SNP 设备日志的 APC 的 fence 代理。

表 A.4. Cisco UCS

字段	描述
Name	Cisco UCS 设备名称。
IP Address	为该设备分配的 IP 地址或者主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
SSL	SSL 连接。
IP port (可选)	用来连接到该设备的 TCP 端口。

字段	描述
Port	物理插件数或者虚拟机名称。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
Power timeout	执行 power off 或者 power on 命令后测试状态更改要等待的秒数。
Shell timeout	执行命令后要等待出现命令提示符的秒数。
Retry on	重试 power on 的次数。
fence_cisco_ucs	Cisco UCS 的 fence 代理。

表 A.5. Cisco MDS

字段	描述
Name	启用 Cisco MDS 9000 系列设备的名称。
IP Address	为该设备分配的 IP 地址或者主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
SNMP version	要使用的 SNMP 版本 (1、2c、3)。
SNMP community	SNMP 社区字符串。
SNMP authentication protocol	SNMP 认证协议 (MD5、SHA)。
SNMP security level	SNMP 安全等级 (noAuthNoPriv、authNoPriv、authPriv)。
SNMP privacy protocol	SNMP 隐私协议 (DES、AES)
SNMP privacy protocol password	SNMP 隐私协议密码。
SNMP privacy protocol script	该脚本为 SNMP 隐私协议提供密码。使用这个参数替代「SNMP privacy protocol password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_cisco_mds	Cisco MDS 的 fence 代理。

表 A.6. Dell DRAC 5

字段	描述
Name	分配给 DRAC 的名称。
IP Address	分配给 DRAC 的 IP 地址或者主机名。
IP port (可选)	用来连接到该设备的 TCP 端口。
Login	访问 DRAC 的登录名
Password	验证到 DRAC 的连接密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Module name	(可选) 当您有多个 DRAC 模块时用于这个 DRAC 的模块名。
Use SSH	表示系统将使用 SSH 访问该设备。

字段	描述
Path to the SSH identity file	SSH 的识别文件。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_drac5	Dell DRAC 5 的 fence 代理。

表 A.7. Egenera SAN 控制器

字段	描述
Name	连接到该集群的 eGenera 刀片设备的名称。
CServer	分配给该设备的主机名（以及可选用户名，格式为 username@hostname）。详情请参考 fence_egenera(8) man page。
ESh Path (可选)	到 eserver 中 esh 命令的路径（默认为 /opt/panmgr/bin/esh）
lpan	该设备的局域网逻辑进程（LPAN）。
pserver	该设备的处理刀片（pserver）。
fence_egenera	eGenera 刀片机的 fence 代理。

表 A.8. ePowerSwitch

字段	描述
Name	连接到该集群的 ePowerSwitch 设备名称。
IP Address	为该设备分配的 IP 地址或者主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
Hidden page	为该设备隐藏的页面名称。
fence_eps	ePowerSwitch 的 fence 代理。

表 A.9. 富士通-西门子远程查看服务栏 (RSB)

字段	描述
Name	作为 fence 设备使用的 RSB 名称。
Hostname	分配给该设备的主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
TCP port	Telnet 服务侦听的端口号。
fence_rsb	富士通-西门子 RSB 的 fence 代理。

表 A.10. Fence virt

字段	描述
Name	Fence virt fence 设备的名称。
Port	要 fence 的虚拟机（域 UUID 或者名称）。

字段	描述
Serial device	在主机中，串口设备必须与每个域的配置文件的映射。有关详情请参考 <code>fence_virt.conf</code> man page。如果指定了该字段，则 <code>fence_virt</code> fencing 代理会使用串口模式操作。不指定该值，则 <code>fence_virt</code> fencing 代理会使用 VM 通道模式操作。
Serial parameters	串口参数。默认为 115200, 8N1。
VM channel IP address	通道 IP。默认值为 10.0.2.179。
Channel port	通道端口。默认值为 1229。
fence_virt	Fence 虚拟 fence 设备的 fence 代理。

表 A.11. 惠普 iLO/iLO2 (Integrated Lights Out)

字段	描述
Name	支持惠普 iLO 支持的服务器名称。
IP Address	为该设备分配的 IP 地址或者主机名。
IP port (可选)	用来连接该设备的 TCP 端口。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_ilo	惠普 iLO 设备的 fence 代理。

表 A.12. 惠普 iLO (Integrated Lights Out) MP

字段	描述
Name	支持惠普 iLO 支持的服务器名称。
Hostname	分配给该设备的主机名。
IP port (可选)	用来连接该设备的 TCP 端口。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
SSH	说明该系统将使用 SSH 访问该设备。
Path to the SSH identity file	SSH 的识别文件。
Force command prompt	提示要使用的命令。默认值为 ' MP>' , ' hpiLO->' 。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_ilo_mp	惠普 iLO MP 设备的 fence 代理。

表 A.13. IBM 刀片服务器

字段	描述
Name	连接该集群的 IBM 刀片服务器设备名称。
IP Address	为该设备分配的 IP 地址或者主机名。

字段	描述
IP port (可选)	用来连接该设备的 TCP 端口。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
Use SSH	表示系统将使用 SSH 访问该设备。
Path to the SSH identity file	SSH 的识别文件。
fence_bladecenter	IBM 刀片服务器的 fence 代理。

表 A.14. IBM 刀片服务器 SNMP

字段	描述
Name	连接该集群的 IBM 刀片服务器 SNMP 名称。
IP Address	为该设备分配的 IP 地址或者主机名。
UDP/TCP port (可选)	用来连接该设备的 UDP/TCP 端口，默认值为 161。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
SNMP version	要使用的 SNMP 版本 (1, 2c, 3)，默认值为 1。
SNMP community	SNMP 社区字符串。
SNMP security level	SNMP 安全等级 (noAuthNoPriv、authNoPriv、authPriv)。
SNMP authentication protocol	SNMP 认证协议 (MD5、SHA)。
SNMP privacy protocol	SNMP 隐私协议 (DES、AES)
SNMP privacy protocol password	SNMP 隐私协议密码。
SNMP privacy protocol script	该脚本为 SNMP 隐私协议提供密码。使用这个参数替代「SNMP privacy protocol password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_bladecenter	IBM 刀片服务器的 fence 代理。

表 A.15. IF MIB

字段	描述
Name	连接该集群的 IF MIB 设备名称。
IP Address	为该设备分配的 IP 地址或者主机名。
UDP/TCP 端口 (可选)	用来与该设备连接的 UDP/TCP 端口，默认值为 161。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。

字段	描述
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
SNMP version	要使用的 SNMP 版本 (1, 2c, 3), 默认值为 1。
SNMP community	SNMP 社区字符串。
SNMP security level	SNMP 安全等级 (noAuthNoPriv、authNoPriv、authPriv)。
SNMP authentication protocol	SNMP 认证协议 (MD5、SHA)。
SNMP privacy protocol	SNMP 隐私协议 (DES、AES)
SNMP privacy protocol password	SNMP 隐私协议密码。
SNMP privacy protocol script	该脚本为 SNMP 隐私协议提供密码。使用这个参数替代「SNMP privacy protocol password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
Port	物理插件数或者虚拟机名称。
fence_ifmib	IF-MIB 设备的 fence 代理。

表 A.16. Intel 模块化

字段	描述
Name	连接到该集群的 Intel 模块化设备名称。
IP Address	为该设备分配的 IP 地址或者主机名。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
SNMP version	要使用的 SNMP 版本 (1, 2c, 3), 默认值为 1。
SNMP community	SNMP 社区字符串。默认值为 private。
SNMP security level	SNMP 安全等级 (noAuthNoPriv、authNoPriv、authPriv)。
SNMP authentication protocol	SNMP 认证协议 (MD5、SHA)。
SNMP privacy protocol	SNMP 隐私协议 (DES、AES)
SNMP privacy protocol password	SNMP 隐私协议密码。
SNMP privacy protocol script	该脚本为 SNMP 隐私协议提供密码。使用这个参数替代「SNMP privacy protocol password」参数。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
fence_intelmodular	APC 的 fence 代理。

表 A.17. IPMI (智能平台管理界面) LAN

字段	描述
Name	连接到该集群的 IPMI LAN 名称。
IP Address	为该设备分配的 IP 地址或者主机名。

字段	描述
Login	可为给定 IPMI 端口发出 power on/off 命令的用户登录名。
Password	用来验证到 IPMI 端口连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Authentication Type	none、password、md2 或者 md5
Use Lanplus	True 或者 1。如果空白, 则该值为 False。
Ciphersuite to use	用于 IPMIv2 lanplus 连接的远程服务器验证、完整性以及加密算法。
fence_ipmilan	IPMI 控制的机器的 fence 代理。

表 A.18. SCSI Fencing

字段	描述
Name	SCSI fence 设备名称。
Node name	要 fence 的节点名称。详情请参考 fence_scsi(8) man page。
fence_scsi	SCSI 长久保存的 fence 代理。



备注

支持使用 SCSI 永久保留作为 fence 方法但有以下限制:

- 当使用 SCSI fencing 时, 集群中的所有节点必须使用同一设备注册, 这样每个节点都可删除另一个节点注册的所有设备的注册密钥。
- 用于集群卷的设备应该是完整的 LUN, 不是分区。SCSI 永久保留适用于整个 LUN, 即控制到每个 LUN 的访问, 而不是对独立分区的访问。

表 A.19. WTI 电源开关

字段	描述
Name	连接到集群的 WTI 电源开关的名称。
IP Address	分配给该设备的 IP 或者主机名地址。
IP port (可选)	用来连接到该设备的 TCP 端口。
Login	访问该设备的登录名称。
Password	用来验证到该设备的连接的密码。
Password Script (可选)	为访问该 fence 设备提供密码的脚本。使用这个参数替代「Password」参数。
Port	物理插件数或者虚拟机名称。
Force command prompt	使用的命令提示。默认值为 [' RSM>' , ' >MPC' , ' IPS>' , ' TPS>' , ' NBB>' , ' NPS>' , ' VMR>']。
Power wait	执行 power off 或者 power on 命令后要等待的秒数。
Use SSH	表示系统将使用 SSH 访问该设备。
Path to the SSH identity file	SSH 的识别文件。
fence_wti	WTI 网络电源开关的 fence 代理。

附录 B. HA 资源参数

本附录提供 HA 资源参数的描述。您可以使用 Luci 或者编辑 `etc/cluster/cluster.conf` 配置参数。表 B.1 “HA 资源小结” 列出了这些资源、其对应的资源代理以及对参考其它包含参数描述的表格。有关资源代理详情请查看任意集群节点中的 `/usr/share/cluster` 文件。

有关 `cluster.conf` 元素和属性的完整列表及描述，请参考 `/usr/share/cluster/cluster.rng` 中的集群方案，以及 `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` 中的注释方案（例如：`/usr/share/doc/cman-3.0.12/cluster_conf.html`）。

表 B.1. HA 资源小结

资源	资源代理	参数描述参考
Apache	apache.sh	表 B.2 “Apache 服务器”
文件系统	fs.sh	表 B.3 “文件系统”
GFS2 文件系统	clusterfs.sh	表 B.4 “GFS2”
IP Address	ip.sh	表 B.5 “IP Address”
LVM	lvm.sh	表 B.6 “LVM”
MySQL	mysql.sh	表 B.7 “MySQL®”
NFS 客户端	nfscient.sh	表 B.8 “NFS 客户端”
NFS 导出	nfsexport.sh	表 B.9 “NFS 导出”
NFS/CIFS 挂载	netfs.sh	表 B.10 “NFS/CIFS 挂载”
Open LDAP	openldap.sh	表 B.11 “Open LDAP”
Oracle 10g	oracledb.sh	表 B.12 “Oracle® 10g”
PostgreSQL 8	postgres-8.sh	表 B.13 “PostgreSQL 8”
SAP 数据库	SAPDatabase	表 B.14 “SAP® 数据库”
SAP 事务	SAPInstance	表 B.15 “SAP® 事务”
Samba	samba.sh	表 B.16 “Samba 服务”
脚本	script.sh	表 B.17 “脚本”
服务	service.sh	表 B.18 “服务”
Sybase ASE	ASEHAagent.sh	表 B.19 “Sybase® ASE 故障切换事务”
Tomcat 6	tomcat-6.sh	表 B.20 “Tomcat 6”
虚拟机	vm.sh	表 B.21 “虚拟机” 备注：如果主机集群可支持虚拟机，则 Luci 会将其显示为虚拟服务。

表 B.2. Apache 服务器

字段	描述
Name	Apache 服务名称
Server Root	默认值为 <code>/etc/httpd</code> 。
Config File	指定 Apache 配置文件。默认值为 <code>/etc/httpd/conf</code> 。
httpd Options	httpd 的其它命令行选项。

字段	描述
Shutdown Wait (秒为单位)	指定等待正确关闭服务的秒数。

表 B.3. 文件系统

字段	描述
Name	指定文件系统资源名称。
File System Type	如果没有指定，mount 会尝试确定文件系统类型。
Mount Point	挂载这个文件系统的文件系统结构路径。
Device	指定与这个文件系统资源关联的设备。可以是块设备、文件系统标签或者文件系统的 UUID。
Options	挂载选项，即在挂载文件系统时使用的选项。这要根据具体文件系统决定。支持的挂载选项请查看 mount(8) man page。
File System ID	 备注 File System ID 只可由 NFS 服务使用 当创建新的文件系统资源时，您可以让此字段保持空白。保持空白后会在配置过程中提交参数后自动分配 file system ID。如果您要具体指定 file system ID，请在此字段中指定。
Force unmount	如果启用该功能，则会强制卸载文件系统。默认设置为 disabled。Force Unmount 会杀死使用该挂载点的所有进程以便在其尝试卸载时释放挂载点。
Reboot host node if unmount fails	如果启用该选项，则在卸载该文件系统失败时重启该节点。默认设置为 disabled。
Check file system before mounting	如果启用该选项，则会在挂载前在文件系统中运行 fsck。默认设置为 disabled。

表 B.4. GFS2

字段	描述
Name	文件系统资源名称
Mount Point	挂载文件系统资源的路径。
Device	与文件系统资源关联的设备文件。
Options	挂载选项

字段	描述
File System ID	 备注 File System ID 只可由 NFS 服务使用 当创建新的 GFS2 资源时，您可以让此字段保持空白。保持空白后会在配置过程中提交参数后自动分配 file system ID。如果您要具体指定 file system ID，请在此字段中指定。
Force Unmount	如果启用该选项，则会强制卸载文件系统。默认设置为 disabled。Force Unmount 会杀死使用该挂载点的所有进程以便在其尝试卸载时释放挂载点。使用 GFS2 资源时，服务停止时不卸载挂载点除非 Force Unmount 是 enabled。
Reboot Host Node if Unmount Fails (self fence)	如果启用该选项且卸载文件系统失败，则该节点将立即重启。通常这与 force-unmount 支持一同使用，但并不要求一定如此。

表 B.5. IP Address

字段	描述
IP Address	资源的 IP 地址。这是一个虚拟 IP 地址。支持 IPv4 和 IPv6，如同监控每个 IP 地址的 NIC 链接。
Monitor Link	如果没有这个 IP 地址绑定的 NIC 的链接，启用此选项将导致状态检查失败。

表 B.6. LVM

字段	描述
Name	这个 LVM 资源的唯一名称。
Volume Group Name	被管理卷组的说明性名称。
Logical Volume Name (可选)	被管理的逻辑卷名称。如果该卷组中被管理的逻辑卷在一个以上，则这个参数是可选的。

表 B.7. MySQL®

字段	描述
Name	指定 MySQL 服务器资源名称。
Config File	指定配置文件。默认值为 /etc/my.cnf。
Listen Address	指定 MySQL 服务器的 IP 地址。如果没有提供 IP 地址，则会为该设备使用第一个 IP 地址。
mysqld Options	httpd 的其它命令行选项。
Shutdown Wait (秒为单位)	指定等待正确关闭服务的秒数。

表 B.8. NFS 客户端

字段	描述
Name	这是客户端用来在资源树中进行参考的符号名。这与 Target 选项不一样。
Target	这是您要执行挂载的服务器。可使用主机名、通配符（基于 IP 地址或者主机名）或者定义主机或者导出主机的网络组群定义该服务器。
Option	为这个客户端定义一组选项 — 例如：额外客户端访问权力。有关详情请参考 exports (5) man page, 《常规选项》。

表 B.9. NFS 导出

字段	描述
Name	资源的说明性名称。NFS 导出资源确定 NFS 守护进程正在运行。它可重复使用，通常只需要 NFS 导出资源。  窍门 为 NFS 导出命名以便区别于其它 NFS 资源。

表 B.10. NFS/CIFS 挂载

字段	描述
Name	NFS 或者 CIFS 挂载的符号名。  备注 只有将集群服务配置为 NFS 客户端时才需要这个资源。
Mount Point	挂载文件系统资源的路径。
Host	NFS/CIFS 服务器 IP 地址或者主机名。
NFS Export Path or CIFS share	NFS 导出目录名或者 CIFS 共享名词。
File System type	文件系统类型： • NFS3 — 指定使用默认 NFS 版本。这是默认设置。 • NFS v4 — 指定使用 NFSv4 协议。 • CIFS — 指定使用 CIFS 协议。
Options	挂载选项。指定挂载选项列表。如果没有指定，则会使用 -o sync 挂载文件系统。
Force Unmount	如果启用 Force Unmount，则集群会在服务停止时杀死所有使用该文件系统的进程。杀死所有使用该文件系统的进程可释放文件系统。另外，如果卸载失败则会重启该服务。
No Unmount	如果启用，可指定在停止或者重新定位操作中不能卸载的文件系统。

表 B.11. Open LDAP

字段	描述
Name	指定文件系统名称用于日志记录或者其它目的。
Config File	指定配置文件的绝对路径。默认值为 /etc/openldap/slapd.conf。
URL List	默认值为 ldap:///。
slapd Options	slapd 的其它命令行选项。
Shutdown Wait (秒为单位)	指定等待正确关闭服务的秒数。

表 B.12. Oracle® 10g

字段	描述
Instance name (SID) of Oracle instance	事务名称
Oracle user name	这是在 Oracle AS 事务中运行的 Oracle 用户的用户名。
Oracle application home directory	这是 Oracle (应用程序, 不是用户) 主目录。您安装 Oracle 后就会配置这个目录。
Virtual hostname (可选)	与 Oracle 10g 安装主机名匹配的虚拟主机名。请注意: 在启动/停止某个 oracledb 资源的过程中, 您的主机名可能会临时更改为这个主机名。因此, 您应该只将 oracledb 资源配置为专有服务的一部分。

表 B.13. PostgreSQL 8

字段	描述
Name	指定文件系统名称用于日志记录或者其它目的。
Config File	定义配置文件的绝对路径。默认值为 /var/lib/pgsql/data/postgresql.conf。
Postmaster User	在 root 无法运行该数据库服务器时运行该服务器的用户。默认值为 postgres。
Postmaster Options	Postmaser 的其它命令行选项。
Shutdown Wait (秒为单位)	指定等待正确关闭服务的秒数。

表 B.14. SAP® 数据库

字段	描述
SAP Database Name	指定唯一 SAP 系统识别符。例如: P01。
SAP executable directory	为 sapstartsrv 和 sapcontrol 指定完全限定路径。
Database type	指定以下数据库类型之一: Oracle、DB6 或者 ADA。

字段	描述
Oracle TNS listener name	指定 Oracle TNS 侦听程序名称。
ABAP stack is not installed, only Java stack is installed	如果您没有在 SAP 数据库中安装 ABAP 栈，则启用这个参数。
J2EE instance bootstrap directory	J2EE 事务引导程序目录的完全限定路径。例如：/usr/sap/P01/J00/j2ee/cluster/bootstrap。
J2EE security store path	J2EE 安全存储目录的完全限定路径。例如：/usr/sap/P01/SYS/global/security/lib/tools。

表 B.15. SAP® 事务

字段	描述
SAP Instance Name	完全限定的 SAP 事务名称。例如：P01_DVEBMGS00_sapp01ci。
SAP executable directory	sapstartsrv 和 sapcontrol 的完全限定路径。
Directory containing the SAP START profile	SAP START 侧写的完全限定路径。
Name of the SAP START profile	SAP START 侧写的指定名称。



备注

根据 表 B.16 “Samba 服务”，当创建或者编辑集群服务时，直接将 Samba 服务资源而不是服务中的资源连接到该服务。

表 B.16. Samba 服务

字段	描述
Name	指定 Samba 服务器的名称。

表 B.17. 脚本

字段	描述
Name	指定自定义用户脚本的名称。该脚本资源允许使用符合 LSB 标准的 init 脚本启动集群的服务。
File (with path)	输入自定义脚本所在路径（例如：/etc/init.d/userscript）。

表 B.18. 服务

字段	描述
Service name	服务名称。这定义了资源集合，也称为资源组或者集群服务。
Automatically start this service	如果启用，则该服务（或者资源组）会在集群达到法定数额时自动启动。如果禁用这个参数，则这个服务就不会在集群达到法定数额时自动启动。该服务处于 disabled 状态。
Run exclusive	如果启用，这个服务（资源组）则只能重新定位独占另一个节点运行，即在没有其它服务运行的节点中运行。如果没有可用服务供其独占运行，则服务在失败后就无法重启。另外，由于使用 Run exclusive，其它服务也无法重新定位到运行这个服务的节点。您可以使用手动启动或者重新定位操作覆盖这个选项。
Failover Domain	定义在服务失败事件中可尝试的集群成员列表。
Recovery policy	Recovery policy 提供以下选项： • Disable — 如果组件失败则禁用资源组。 • Relocate — 尝试在另一个节点中重启服务，就是说不要在当前节点中尝试重启。 • Restart — 在尝试将服务 relocate（默认选项）到另一个节点前，请尝试在本地（在当前节点中）重启这个服务中失败的部分。 • Restart-Disable — 将在服务失败的地方重启该服务。但如果重启该服务失败，则会禁用该服务，而不是将其移动到该集群的另一台主机中。

表 B.19. Sybase® ASE 故障切换事务

字段	描述
Instance Name	指定 Sybase ASE 资源的事务名称。
ASE server name	为 HA 服务配置的 ASE 服务器名称。
Sybase home directory	Sybase 产品的主目录。
Login file	包含登录密码对的登录文件全路径。
Interfaces file	用来启动/访问 ASE 服务器的接口文件全路径。
SYBASE_ASE directory name	sybase_home 中安装 ASE 产品的目录名。
SYBASE_OCS directory name	sybase_home 中安装 OCS 产品的目录名。例如：ASE-15_0。
Sybase user	可运行 ASE 服务器的用户。
Deep probe timeout	在运行深度探测时，确定服务器没有响应前等待 ASE 服务器响应的最长秒数。

表 B.20. Tomcat 6

字段	描述
Name	指定文件系统名称用于日志记录或者其它目的。

字段	描述
Config File	指定配置文件的绝对路径。默认值为 <code>/etc/tomcat6/tomcat6.conf</code> 。
Tomcat User	运行 Tomcat 服务器的用户。默认值为 <code>tomcat</code> 。
Catalina Options	Catalina 的其它命令行选项。
Catalina Base	Catalina 基础目录（与其它服务不同）。默认值为 <code>/usr/share/tomcat6</code> 。
Shutdown Wait (秒为单位)	指定正常结束关闭服务的要等待的秒数。默认值为 <code>30</code> 。

表 B.21. 虚拟机

字段	描述
Name	指定虚拟机名称。当使用 <code>luci</code> 界面时，您可以将其指定为服务名。
Automatically start this virtual machine	如果启用，则这台虚拟机会在集群达到定额数后自动启动。如果禁用这个参数，则这台虚拟机就不会在集群达到定额数后自动启动。虚拟机处于 <code>disabled</code> 状态。
Run exclusive	如果启用，这台虚拟机则只能重新定位独占另一个节点运行，即在没有其它虚拟机运行的节点中运行。如果没有可用的虚拟机供其独占运行，则虚拟机在失败后就无法重启。另外，由于使用 <code>Run exclusive</code> ，其它虚拟机也无法重新定位到运行这台虚拟机的节点。您可以使用手动启动或者重新定位操作覆盖这个选项。
Failover domain	定义在虚拟机失败事件中可尝试的集群成员列表。
Recovery policy	Recovery policy 提供以下选项： • <code>Disable</code> — 失败后禁用该虚拟机。 • <code>Relocate</code> — 尝试在另一个节点中重启该虚拟机，即不在当前节点中重启。 • <code>Restart</code> — 在尝试将该虚拟机重新定位（默认选项）到另一个节点中前，尝试在本地（当前节点中）重启该虚拟机。 • <code>Restart-Disable</code> — 将在服务失败的地方重启该服务。但如果重启该服务失败，则会禁用该服务，而不是将其移动到该集群的另一台主机中。
Restart options	如果您选择「重启」或者「禁用重启」作为该服务的恢复策略，您可以指定在重新定位或者禁用该服务前最多重启失败的次数，并指定多少秒后不再重启。
Migration type	指定迁移类型 <code>live</code> 或者 <code>pause</code> ，默认设置为 <code>live</code> 。
Migration mapping	为迁移指定可替换接口。例如：您可以在当某个节点用于虚拟机迁移的网络地址与该节点用来进行集群通信的地址不同时指定可替换接口。 指定以下说明当您将虚拟机从 <code>member</code> 迁移到 <code>member2</code> 时，您实际上是迁移到 <code>target2</code> 。同样，当您从 <code>member2</code> 迁移到 <code>member</code> 时，您使用 <code>target</code> 迁移。 <code>member:target,member2:target2</code>
Status Program	除对出现的虚拟机进行常规检查外要运行的状态程序。如果指定，则每分钟执行一次状态程序。这可让您了解虚拟机中关键服务的状态。例如：如果某台虚拟机运行网页服务器，您的状态程序可查看该网页服务器是否启动并运行。如果该状态检查失败（返回一个非零值），就是覆盖了该虚拟机。

字段	描述
	启动虚拟机后，该虚拟机资源代理会周期性调用该状态程序，并期待一个成功返回代码（0）。5 分钟后超时。
Path to XML file used to create the VM	到 libvirt XML 文件的完整路径包含 libvirt 域定义。
VM configuration file path	一组用冒号分隔的路径，虚拟机资源代理（vm.sh）用它搜索虚拟机配置文件。例如： /mnt/guests/config:/etc/libvirt/qemu。  重要 该路径应该绝对不直接指向某个虚拟机配置文件。
Path to VM snapshot directory	到保存虚拟机映像的快照目录的路径。
Hypervisor URI	管理 URI（通常为自动）。
Migration URI	迁移 URI（通常为自动）。

附录 C. HA 资源行为

本附录描述了 HA 资源的一般行为，其目的是提供在配置 HA 服务时可能有所帮助的辅助信息。您可使用 `Luci` 或者 `etc/cluster/cluster.conf` 配置参数。有关 HA 资源参数的描述，请参考 [附录 B](#)，[HA 资源参数](#)。要了解资源代理详情请查看集群节点中的 `/usr/share/cluster` 文件。



备注

要完全理解本附录中的信息，您需要对资源代理以及集群配置文件 `/etc/cluster/cluster.conf` 有深入的理解。

HA 服务是一组在统一实体中配置的集群资源，可为客户端提供特定的服务。HA 服务在集群配置文件 `/etc/cluster/cluster.conf`（在每个集群节点中）中以资源树的形式出现。在集群配置文件中，每个资源树都使用一个 XML 代表，可指定每个资源及其属性和在资源树中与其它资源的关系（上级、下级和平级关系）。



备注

因为 HA 服务由分为层次树的资源组成，所有服务有时也指的是资源树或者资源组。这两个词组与 HA 服务有相同的含义。

在每个资源树的顶端是一个特殊的资源类型 — 服务资源。其它资源类型构成服务的其它部分，决定服务的特点。配置 HA 服务包括创建服务资源、创建下级集群资源以及将其组成统一实体形成该服务的分级限制。

本附录由以下部分组成：

- [第 C.1 节 “资源间的上级、下级和同级关系”](#)
- [第 C.2 节 “同级资源启动顺序以及资源下级顺序”](#)
- [第 C.3 节 “继承、<资源>块以及重新使用资源”](#)
- [第 C.4 节 “故障恢复和独立子树”](#)
- [第 C.5 节 “调整并测试服务和资源顺序”](#)



备注

在集群配置文件 `/etc/cluster/cluster.conf` 示例之后的这部分只作为演示使用。

C.1. 资源间的上级、下级和同级关系

集群服务是一个在 `rgmanager` 控制下运行的整合实体。服务中的所有资源都在同一节点中运行。从 `rgmanager` 角度来看，一个集群服务就是一个可启动、停止或者重新定位的实体。但在集群服务中，资源结构根据每个启动和停止的资源顺序而定。结构等级包括上级、下级和同级。

例 C.1 “服务 `foo` 的资源结构” 显示服务 `foo` 的资源树示例。在这个示例中，资源间的关系如下：

- `fs:myfs` (`<fs name="myfs" ...>`) 和 `ip:10.1.1.2` (`<ip address="10.1.1.2 .../>`) 是同级。
- `fs:myfs` (`<fs name="myfs" ...>`) 是 `script:script_child` (`<script name="script_child"/>`) 的上级。
- `script:script_child` (`<script name="script_child"/>`) 是 `fs:myfs` (`<fs name="myfs" ...>`) 的下级。

例 C.1. 服务 `foo` 的资源结构

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

在资源树的上/下级关系中采用以下规则：

- 上级资源在下级资源之前启动。
- 在停止上级资源前必须停止全部下级资源。
- 对于正常工作的资源，其下级资源必须全部正常工作。

C.2. 同级资源启动顺序以及资源下级顺序

服务资源根据是否为子资源指定子类型属性决定子资源的启动和停止顺序，如下：

- 指定子类型属性（归类的子资源）— 如果服务资源为子资源指定子类型属性，则该子资源就被归类了。该子类型属性明确决定该子资源的启动和停止顺序。
- 不指定子类型属性（不归类的子资源）— 如果服务资源不为子资源指定子类型属性，则该子资源是不归类的。该服务资源不会明确控制不归类子资源的启动和停止顺序。但不归类子资源根据其在 `/etc/cluster.cluster.conf` 中的顺序启动和停止。另外，不归类子资源在所有归类子资源启动后启动，并在所有归类子资源停止前停止。

备注

使用定义的子资源类型排序的唯一资源为服务资源。

有关归类子资源启动和停止顺序的详情请参考 [第 C.2.1 节 “归类子资源启动和停止顺序”](#)。有关不归类子资源启动和停止顺序的详情请参考 [第 C.2.2 节 “不归类子资源启动和停止顺序”](#)。

C.2.1. 归类子资源启动和停止顺序

在归类子资源中，子资源的类型属性定义每个资源类型的启动和停止顺序，数字从 1 到 100，一个值为启动顺序，一个值为停止顺序。数字越小，越早启动或者停止该资源类型。例如：在表 C.1 “子资源类型启动和停止顺序”中演示的每个资源类型值；例 C.2 “资源启动和停止值：服务资源代理 service.sh 除外”中演示的在服务资源代理 service.sh 中显示的启动和停止值。在服务资源中，首先启动所有 LVM 子资源，然后是所有文件系统子资源，之后是所有脚本子资源，以此类推。

表 C.1. 子资源类型启动和停止顺序

资源	子类型	启动顺序值	停止顺序值
LVM	lvm	1	9
File System	fs	2	8
GFS2 File System	clusterfs	3	7
NFS Mount	netfs	4	6
NFS Export	nfsexport	5	5
NFS Client	nfscient	6	4
IP Address	ip	7	2
Samba	smb	8	3
Script	script	9	1

例 C.2. 资源启动和停止值：服务资源代理 service.sh 除外

```
<special tag="rgmanager">
  <attributes root="1" maxinstances="1"/>
  <child type="lvm" start="1" stop="9"/>
  <child type="fs" start="2" stop="8"/>
  <child type="clusterfs" start="3" stop="7"/>
  <child type="netfs" start="4" stop="6"/>
  <child type="nfsexport" start="5" stop="5"/>
  <child type="nfscient" start="6" stop="4"/>
  <child type="ip" start="7" stop="2"/>
  <child type="smb" start="8" stop="3"/>
  <child type="script" start="9" stop="1"/>
</special>
```

资源类型的顺序与其在集群配置文件 /etc/cluster/cluster.conf 中的顺序一致。例如：注意例 C.3 “资源类型中的排序”中的归类子资源启动和停止顺序。

例 C.3. 资源类型中的排序

```
<service name="foo">
  <script name="1" .../>
  <lvm name="1" .../>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

归类子资源的启动顺序

在例 C.3 “资源类型中的排序”中，资源按如下顺序启动：

1. lvm:1 — 这是 LVM 资源。首先启动所有 LVM 资源。lvm:1 (<lvm name="1" .../>) 是 LVM 资源中第一个启动的 LVM 资源，因为它是 /etc/cluster/cluster.conf 文件 Service foo 部分的一个 LVM 资源。
2. lvm:2 — 这是 LVM 资源。首先启动所有 LVM 资源。lvm:2 (<lvm name="2" .../>) 是在 lvm:1 之后启动的资源，因为它列在 /etc/cluster/cluster.conf 文件 Service foo 部分 lvm:1 之后。
3. fs:1 — 这是文件系统资源。如果在 Service foo 中还有其它文件系统资源，则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。
4. ip:10.1.1.1 — 这是 IP 地址资源。如果在 Service foo 中还有其它 IP 地址资源，则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。
5. script:1 — 这是脚本资源。如果在 Service foo 中还有其它脚本资源，则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。

归类的子资源停止顺序

在 例 C.3 “资源类型中的排序” 中资源按照如下顺序停止：

1. script:1 — 这是脚本资源。如果在 Service foo 中还有其它脚本资源，则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
2. ip:10.1.1.1 — 这是 IP 地址资源。如果在 Service foo 中还有其它 IP 地址资源，则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
3. fs:1 — 这是文件系统资源。如果在 Service foo 中还有其它文件系统资源，则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
4. lvm:2 — 这是 LVM 资源。最后停止所有 LVM 资源。lvm:2 (<lvm name="2" .../>) 是在 lvm:1 之前停止的资源，资源类型组中的资源按照 /etc/cluster/cluster.conf 文件 Service foo 部分的相反顺序停止。
5. lvm:1 — 这是 LVM 资源。最后停止所有 LVM 资源。lvm:1 (<lvm name="1" .../>) 是在 lvm:2 之后停止的资源，资源类型组中的资源按照 /etc/cluster/cluster.conf 文件 Service foo 部分的相反顺序停止。

C.2.2. 不归类子资源启动和停止顺序

还需要考虑不归类的子资源。在不归类子资源中，服务资源不明确指定启动和停止顺序。反之，启动和停止顺序是由 /etc/cluster/cluster.conf 中的子资源顺序决定的。另外，不归类子资源在所有归类子资源启动后启动，所有归类子资源停止前停止。

例如：在 例 C.4 “服务中的不归类和归类子资源” 中演示的不归类子资源启动和停止顺序。

例 C.4. 服务中的不归类和归类子资源

```
<service name="foo">
  <script name="1" .../>
  <nontypedresource name="foo"/>
  <lvm name="1" .../>
  <nontypedresourcetwo name="bar"/>
  <ip address="10.1.1.1" .../>
  <fs name="1" .../>
  <lvm name="2" .../>
</service>
```

不归类子资源启动顺序

在 例 C.4 “服务中的不归类 and 归类子资源” 中子资源按如下顺序启动:

1. lvm:1 — 这是 LVM 资源。首先启动所有 LVM 资源。lvm:1 (<lvm name="1" .../>) 是 LVM 资源中第一个启动的 LVM 资源, 因为它是 /etc/cluster/cluster.conf 文件 Service foo 部分的一个 LVM 资源。
2. lvm:2 — 这是 LVM 资源。首先启动所有 LVM 资源。lvm:2 (<lvm name="2" .../>) 是在 lvm:1 之后启动的资源, 因为它列在 /etc/cluster/cluster.conf 文件 Service foo 部分 lvm:1 之后。
3. fs:1 — 这是文件系统资源。如果在 Service foo 中还有其它文件系统资源, 则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。
4. ip:10.1.1.1 — 这是 IP 地址资源。如果在 Service foo 中还有其它 IP 地址资源, 则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。
5. script:1 — 这是脚本资源。如果在 Service foo 中还有其它脚本资源, 则应按照 /etc/cluster/cluster.conf 文件 Service foo 部分的顺序启动。
6. nontypedresource:foo — 这是不归类资源。因为它是不归类资源, 因此在归类资源之后启动。另外, 它在服务资源中的顺序在其它不归类资源 nontypedresourcetwo:bar 前面, 因此它在 nontypedresourcetwo:bar 之前启动。(不归类资源按照它们在服务资源中出现的顺序启动。)
7. nontypedresourcetwo:bar — 这是不归类资源。因为它是不归类资源, 因此在归类资源之后启动。另外, 它在服务资源中的顺序在其它不归类资源 nontypedresourcetwo:foo 后面, 因此它在 nontypedresourcetwo:foo 之后启动。(不归类资源按照它们在服务资源中出现的顺序启动。)

不归类资源停止顺序

在 例 C.4 “服务中的不归类 and 归类子资源” 中子资源按照如下顺序停止:

1. nontypedresourcetwo:bar — 这是不归类资源。因为它是不归类资源, 因此在归类资源之前停止。另外, 它在服务资源中的顺序在其它不归类资源 nontypedresourcetwo:foo 后面, 因此它在 nontypedresourcetwo:foo 之前停止。(不归类资源按照它们在服务资源中出现的相反顺序停止。)
2. nontypedresource:foo — 这是不归类资源。因为它是不归类资源, 因此在归类资源之前停止。另外, 它在服务资源中的顺序在其它不归类资源 nontypedresourcetwo:bar 前面, 因此它在 nontypedresourcetwo:bar 之后停止。(不归类资源以其在服务资源中出现的相反顺序停止。)
3. script:1 — 这是脚本资源。如果在 Service foo 中还有其它脚本资源, 则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
4. ip:10.1.1.1 — 这是 IP 地址资源。如果在 Service foo 中还有其它 IP 地址资源, 则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
5. fs:1 — 这是文件系统资源。如果在 Service foo 中还有其它文件系统资源, 则应按照与 /etc/cluster/cluster.conf 文件 Service foo 部分相反的顺序停止。
6. lvm:2 — 这是 LVM 资源。最后停止所有 LVM 资源。lvm:2 (<lvm name="2" .../>) 是在 lvm:1 之前停止的资源, 资源类型组中的资源按照 /etc/cluster/cluster.conf 文件 Service foo 部分的相反顺序停止。

7. `lvm:1` — 这是 LVM 资源。最后停止所有 LVM 资源。`lvm:1` (`<lvm name="1" .../>`) 是在 `lvm:2` 之后停止的资源，资源类型组中的资源按照 `/etc/cluster/cluster.conf` 文件 Service foo 部分的相反顺序停止。

C.3. 继承、<资源>块以及重新使用资源

有些资源利用从上级资源中继承的值，通常是在 NFS 服务中。例 C.5 “资源再利用和继承的 NFS 服务设置” 演示了典型的 NFS 服务配置，资源再利用和继承设置。

例 C.5. 资源再利用和继承的 NFS 服务设置

```
<resources>
  <nfsclient name="bob" target="bob.test.com" options="rw,no_root_squash"/>
  <nfsclient name="jim" target="jim.test.com" options="rw,no_root_squash"/>
  <nfsexport name="exports"/>
</resources>
<service name="foo">
  <fs name="1" mountpoint="/mnt/foo" device="/dev/sdb1" fsid="12344">
    <nfsexport ref="exports"> <!-- nfsexport's path and fsid attributes
                                are inherited from the mountpoint &
                                fsid attribute of the parent fs
                                resource -->
    <nfsclient ref="bob"/> <!-- nfsclient's path is inherited from the
                                mountpoint and the fsid is added to the
                                options string during export -->
    <nfsclient ref="jim"/>
  </nfsexport>
</fs>
  <fs name="2" mountpoint="/mnt/bar" device="/dev/sdb2" fsid="12345">
    <nfsexport ref="exports">
    <nfsclient ref="bob"/> <!-- Because all of the critical data for this
                                resource is either defined in the
                                resources block or inherited, we can
                                reference it again! -->
    <nfsclient ref="jim"/>
  </nfsexport>
</fs>
  <ip address="10.2.13.20"/>
</service>
```

如果服务是平面的（即没有上级/下级关系），则需要按使用如下配置：

- 该服务需要四个 `nfsclient` 资源 — 每个文件系统一个（一共两个文件系统），每个目标机器一个（一共两台目标机器）。
- 服务需要为每个 `nfsclient` 指定 export path 和 file system ID，配置的这个部分容易出错。

但在例 C.5 “资源再利用和继承的 NFS 服务设置” 中只定义一次 NFS 客户端资源 `nfsclient:bob` 和 `nfsclient:jim`；同样也只定义一次 NFS 导出资源 `nfsexport:exports`。资源需要的所有属性都是从上级资源中继承的。因为继承的属性是动态的（并不与其它属性冲突），因此可能再利用那些资源 — 也就是为什么要在资源块中定义它们的原因。在很多情况下可能对配置一些资源没有可操作性。例如：在很多情况下配置文件系统资源可导致在两个节点中挂载一个文件系统，这样就会出问题。

C.4. 故障恢复和独立子树

在很多企业级环境中，如果服务的组件失败，修复的正常动作是重启整个服务。例如：在 例 C.6 “服务 foo 常见故障恢复” 中，如果这个服务中定义的任意脚本失败，则正常动作是重启（重新定位或者禁用，视服务恢复策略而定）该服务。但在有些情况下，服务的某些部分可能是不重要的，可能有必要在尝试一般恢复前重启该服务的某一部分。要完成这个操作，您可以使用 `__independent_subtree` 属性。例如：在 例 C.7 “使用 `__independent_subtree` 属性对服务 foo 执行故障恢复” 中，可使用 `__independent_subtree` 属性完成以下动作：

- 如果 `script:script_one` 失败，重启 `script:script_one`、`script:script_two` 和 `script:script_three`。
- 如果 `script:script_two` 失败，则只重启 `script:script_two`。
- 如果 `script:script_three` 失败，则重启 `script:script_one`、`script:script_two` 和 `script:script_three`。
- 如果 `script:script_four` 失败，则重启整个服务。

例 C.6. 服务 foo 常见故障恢复

```
<service name="foo">
  <script name="script_one" ...>
    <script name="script_two" .../>
  </script>
  <script name="script_three" .../>
</service>
```

例 C.7. 使用 `__independent_subtree` 属性对服务 foo 执行故障恢复

```
<service name="foo">
  <script name="script_one" __independent_subtree="1" ...>
    <script name="script_two" __independent_subtree="1" .../>
    <script name="script_three" .../>
  </script>
  <script name="script_four" .../>
</service>
```

在有些情况下，如果服务的某个组件失败，您想要只禁用那个组件而不禁用整个服务，以避免影响其它使用那个服务的其它组件的服务。从红帽企业版 Linux 6.1 发行本开始，您可以使用 `__independent_subtree="2"` 属性达到此目的，该属性将独立子树标为非关键（non-critical）。



注意

您可以在单一参考的资源中只使用 non-critical 标签。这个 non-critical 标签可在所有资源中使用，并可用于资源树的所有层，但在定义服务或者虚拟机时不应在顶层使用。

从红帽企业版 Linux 6.1 发行本开始，您可以在该资源数的每个节点中为独立的子树设定最多重启和重启过期。您可以使用 以下属性设定这些阈值：

- `__max_restarts` 设定在放弃努力前最多可承受的重启次数。

- `__restart_expire_time` 以秒为单位设定在多长时间后不再尝试重启。

C.5. 调整并测试服务和资源顺序

您可以使用 `rg_test` 程序调整并测试服务和资源顺序。`rg_test` 是命令行工具，它由 `rgmanager` 软件包提供，在 `shell` 或者终端中运行（在 `Conga` 中不可用）。表 C.2 “`rg_test` 程序小结”中总结了 `rg_test` 的动作和语法。

表 C.2. `rg_test` 程序小结

动作	语法
显示 <code>rg_test</code> 理解的资源规则。	<code>rg_test rules</code>
测试配置（和 <code>/usr/share/cluster</code> ）中的错误和冗余资源代理。	<code>rg_test test /etc/cluster/cluster.conf</code>
显示服务的启动和停止排序。	显示启动顺序： <code>rg_test noop /etc/cluster/cluster.conf start service servicename</code> 显示停止顺序：
明确启动或者停止服务。	<code>rg_test noop /etc/cluster/cluster.conf stop service servicename</code>  重要 只在一个节点中这样做并总是先禁用 <code>rgmanager</code> 中的服务。 启动服务： <code>rg_test test /etc/cluster/cluster.conf start service servicename</code> 停止服务：
计算并显示两个 <code>cluster.conf</code> 文件的资源树 <code>delta</code> 。	<code>rg_test test /etc/cluster/cluster.conf stop service servicename</code> <code>rg_test delta cluster.conf file 1 cluster.conf file 2</code> 例如： <code>rg_test delta /etc/cluster/cluster.conf.bak /etc/cluster/cluster.conf</code>

附录 D. 命令行工具小结

表 D.1 “命令行工具小结”总结了配置和管理高可用性附加组件的首选命令行工具。有关命令和变量的详情请参考每个命令行工具的 `man page`。

表 D.1. 命令行工具小结

命令行工具	使用	目的
<code>ccs_config_dump</code> — 集群配置转储工具	集群基本构架	<code>ccs_config_dump</code> 生成正在运行配置的 XML 输出。运行的配置有时与文件中保存的配置有所不同，因为有些子系统在配置中保存或者设定一些默认信息。那些值通常不出现在配置的磁盘版本中，但在运行时需要它们方可使集群正常工作。
<code>ccs_config_validate</code> — 集群配置验证工具	集群基本构架	<code>ccs_config_validate</code> 根据方案 <code>cluster.rng</code> （在每个节点中位于 <code>/usr/share/cluster/cluster.rng</code> ）验证 <code>cluster.conf</code> 。有关这个工具的详情请参考 <code>ccs_config_validate(8) man page</code> 。
<code>clustat</code> — 集群状态工具	高可用性服务管理组件	<code>clustat</code> 命令显示集群的状态。它显示成员信息、仲裁查看以及所有配置的用户服务状态。有关这个工具的详情请参考 <code>clustat(8) man page</code> 。
<code>clusvcadm</code> — 集群用户服务管理工具	高可用性服务管理组件	<code>clusvcadm</code> 命令允许您在集群中启用、禁用、重新定位以及重启高可用性风俗。有关这个工具的详情请参考 <code>clusvcadm(8) man page</code> 。
<code>cman_tool</code> — 集群管理工具	集群基本构架	<code>cman_tool</code> 是管理 CMAN 集群管理器的程序。它可提供加入或者离开集群、杀死节点或者更改集群中节点预期仲裁 <code>vote</code> 的功能。有关这个工具的详情请参考 <code>cman_tool(8) man page</code> 。
<code>fence_tool</code> — Fence 工具	集群基本构架	<code>fence_tool</code> 是用来加入和离开 fence 域的程序。有关这个工具的详情请参考 <code>fence_tool(8) man page</code> 。

附录 E. 修订记录

修订 2.0-1-0 Thu May 19 2011

Steven Levine slevine@redhat.com

红帽企业版 Linux 6.1 发行本

解决: #671250

编写 SNMP trap 支持文档。

解决: #659753

编写 ccs 命令文档。

解决: #665055

更新 Conga 文档, 添加更新的显示和功能支持。

解决: #680294

编写 ricci 代理的密码访问文档。

解决: #687871

添加故障排除一章。

解决: #673217

修复排版错误。

解决: #675805

在 HA 资源参数表中添加 cluster.conf 方案参考。

解决: #672697

更新 fence 设备参数, 添加目前支持的所有 fencing 设备。

解决: #677994

修改 fence_ilo fence 代理参数信息。

解决: #629471

添加有关在双节点集群中设定 consensus 值的技术说明。

解决: #579585

更新升级红帽高可用性附加组件软件一节的内容。

解决: #643216

阐明文档中的一些小问题。

解决: #643191

改进并修改 luci 文档。

解决: #704539

更新虚拟机资源参数表。

修订 1.0-1-0 Wed Nov 10 2010

Paul Kennedy pkennedy@redhat.com

初始发行

索引

符号

仲裁磁盘

使用注意事项, 17

全局管理

常规考虑, 5

关系

集群资源, 135

参数, fence 设备,

参数, HA 资源,

反馈, x, x

多播地址

使用网络开关和多播地址时的注意事项, 18

工具, 命令行,

常规

集群管理注意事项, 5

故障排除

诊断并修正集群中的问题, 70,

整合的 fence 设备

配置 ACPI, 7

电源开关,

(参见 电源控制器)

电源控制器连接, 配置,

硬件

兼容, 6

简介,

其它 Red Hat Enterprise Linux 文档,

类型

集群资源, 14

行为, HA 资源,

表格

HA 资源, 参数,

电源控制器连接, 配置,

配置

HA 服务, 11

集群

启动、停止、重启, 91

管理, , , ,

诊断并修正问题, 70,

集群服务, 36, 63, 85

(参见 添加到集群配置)

(参见 添加到集群配置中)

集群服务管理器

配置, 36, 63, 85

集群管理, , , ,

NetworkManager, 17

ricci 注意事项, 19

SELinux, 18

使用 clustat 显示 HA 服务, 100

使用 cman_tool version -r 更新集群配置
, 103

使用 qdisk 的注意事项, 17

使用 scp 更新集群配置, 105

使用仲裁磁盘的注意事项, 17

停止集群, 41, 69

兼容硬件, 6

删除集群, 41

删除集群节点, 41

加入集群, 40, 69

启动、停止、重启集群, 91

启动集群, 41, 69

启用 IP 端口, 6

在配置中删除节点; 在配置中添加节点, 93

更新配置, 103

添加集群节点, 40, 69

离开集群, 40, 69

管理集群节点, 39, 69

管理高可用性服务, 42, 100

管理高可用性服务, 冻结和解冻, 101, 102

网络切换和多播地址, 18

诊断并修正集群中的问题, 70,

配置 ACPI, 7

配置 iptables, 6

配置验证, 14

重启集群, 41

重启集群节点, 39

集群资源关系, 135

集群起源类型, 14

集群软件

配置, , ,

集群配置, , ,

删除或者添加节点, 93

更新, 103

验证

集群管理, 14

A

ACPI

配置, 7

C

Conga

访问, 3

consensus 值, 75

H

HA 服务配置

概述, 11

I

IP 端口

启用, 6

iptables

配置, 6

N

NetworkManager
在集群中禁用, 17

Q

qdisk
使用注意事项, 17

R

ricci
集群管理注意事项, 19

S

SELinux
配置, 18

T

totem 标签
consensus 值, 75