



**SUSE Linux Enterprise High Availability
Extension 15 SP2
管理指南**

管理指南

SUSE Linux Enterprise High Availability Extension 15 SP2

作者：Tanja Roth、Thomas Schraitle

本指南适用于需要使用 SUSE® Linux Enterprise High Availability Extension 设置、配置和维护群集的管理员。为了能快速且有效地进行配置和管理，产品提供有图形用户界面和命令行界面 (CLI)。本指南介绍了如何通过这两种方法执行关键任务。您可以根据自己的需要选择适当的工具。

出版日期：2024 年 12 月 12 日

<https://documentation.suse.com> 

版权所有 © 2006–2024 SUSE LLC 和贡献者。保留所有权利。

根据 GNU 自由文档许可证 (GNU Free Documentation License) 版本 1.2 或（根据您的选择）版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的许可权限；本版权声明和许可证附带不可变部分。许可证版本 1.2 的副本包含在题为“GNU 自由文档许可证”的部分。

有关 SUSE 商标，请参见 <http://www.suse.com/company/legal/> 。所有其他第三方商标是其各自所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

目录

关于本指南 xv

- 1 可用文档 xvi
- 2 提供反馈 xvii
- 3 文档约定 xvii
- 4 产品生命周期和支持 xix
 - SUSE Linux Enterprise High Availability Extension 的支持声明 xx · 技术预览 xxi

I 安装、设置和升级 1

1 产品概述 2

- 1.1 作为扩展提供 2
- 1.2 主要特征 2
 - 各种群集方案 2 · 灵活性 3 · 储存和数据复制 3 · 支持虚拟环境 4 · 支持本地、城域和 Geo 群集 4 · 资源代理 5 · 用户友好的管理工具 5
- 1.3 优势 6
- 1.4 群集配置：储存 8
- 1.5 体系结构 11
 - 体系结构层 11 · 处理流程 13

2 系统要求和建议 14

- 2.1 硬件要求 14
- 2.2 软件要求 15

2.3 储存要求 16

2.4 其他要求和建议 16

3 安装 High Availability Extension 18

3.1 手动安装 18

3.2 使用 AutoYaST 进行批量安装和部署 18

4 使用 YaST 群集模块 20

4.1 术语定义 20

4.2 YaST 群集模块 22

4.3 定义通讯通道 23

4.4 定义身份验证设置 28

4.5 将配置传送到所有节点 29

使用 YaST 配置 Csync2 30 · 使用 Csync2 同步更改 31

4.6 同步群集节点间的连接状态 33

4.7 配置服务 34

4.8 使群集联机 35

5 升级群集和更新软件包 37

5.1 术语 37

5.2 将群集升级到最新产品版本 38

SLE HA 和 SLE HA Geo 支持的升级路径 39 · 升级前必须满足的先决条件 42 · 群集脱机升级 42 · 群集滚动升级 45

5.3 更新群集节点上的软件包 46

5.4 更多信息 47

II 配置和管理 48

6 配置和管理基础 49

6.1 用例方案 49

6.2 仲裁判定 50

全局群集选项 51 · 全局选项 no-quorum-policy 51 · 全局选项 stonith-enabled 52 · 双节点群集的 Corosync 配置 52 · N 节点群集的 Corosync 配置 53

6.3 群集资源 54

资源管理 54 · 支持的资源代理类 55 · 资源类型 56 · 资源模板 57 · 高级资源类型 57 · 资源选项（元属性） 60 · 实例属性（参数） 63 · 资源操作 66 · 超时值 67

6.4 资源监视 68

6.5 资源约束 69

约束类型 70 · 分数和无限值 73 · 资源模板和约束 73 · 故障转移节点 74 · 故障回复节点 75 · 根据资源负载影响放置资源 76 · 使用标记分组资源 78

6.6 管理远程主机上的服务 79

使用监视插件监视远程主机上的服务 79 · 使用 pacemaker_remote 管理远程节点上的服务 81

6.7 监视系统运行状况 81

6.8 更多信息 83

7 使用 Hawk2 配置和管理群集资源 84

7.1 Hawk2 要求 84

7.2 登录 85

7.3 Hawk2 概述：主要元素 86

左侧导航栏 87 · 顶层行 88

- 7.4 配置全局群集选项 88
- 7.5 配置群集资源 90
 - 显示当前群集配置 (CIB) 91 · 使用向导添加资源 91 · 添加简单资源 92 · 添加资源模板 94 · 修改资源 94 · 添加 STONITH 资源 96 · 添加群集资源组 97 · 添加克隆资源 99 · 添加多状态资源 100 · 使用标记分组资源 101 · 配置资源监视 102
- 7.6 配置约束 104
 - 添加位置约束 105 · 添加共置约束 106 · 添加顺序约束 108 · 在约束中使用资源集 110 · 更多信息 111 · 指定资源故障转移节点 112 · 指定资源故障回复节点 (资源粘性) 113 · 根据负载影响配置资源放置 114
- 7.7 管理群集资源 116
 - 编辑资源和组 116 · 启动资源 117 · 清理资源 118 · 删除群集资源 118 · 迁移群集资源 119
- 7.8 监视群集 120
 - 监视单个群集 120 · 监视多个群集 121
- 7.9 使用批模式 124
- 7.10 查看群集历史记录 127
 - 查看节点或资源的最近事件 128 · 使用历史记录浏览器获得群集报告 129 · 在历史记录浏览器中查看转换细节 131
- 7.11 校验群集状态 133
- 8 配置和管理群集资源 (命令行) 134**
- 8.1 crmsh - 概述 134
 - 获得帮助 135 · 执行 crmsh 的子命令 136 · 显示有关 OCF 资源代理的信息 137 · 使用 crmsh 的外壳脚本 139 · 使用 crmsh 的群集脚本 139 · 使用配置模板 142 · 使用阴影配置进行测试 145 · 调试配置更改 145 · 群集图表 146
- 8.2 管理 Corosync 配置 146

- 8.3 配置群集资源 148
 - 从文件装载群集资源 148 · 创建群集资源 148 · 创建资源模板 149 · 创建 STONITH 资源 150 · 配置资源约束 152 · 指定资源故障转移节点 154 · 指定资源故障回复节点 (资源粘性) 155 · 根据负载影响配置资源放置 155 · 配置资源监视 158 · 配置群集资源组 158 · 配置克隆资源 159
- 8.4 管理群集资源 160
 - 显示群集资源 160 · 启动新的群集资源 162 · 停止群集资源 162 · 清理资源 163 · 删除群集资源 163 · 迁移群集资源 164 · 分组/标记资源 164 · 获取运行状态 165
- 8.5 设置独立于 cib.xml 的密码 165
- 8.6 检索历史记录信息 166
- 8.7 更多信息 167
- 9 添加或修改资源代理 168**
 - 9.1 STONITH 代理 168
 - 9.2 写入 OCF 资源代理 168
 - 9.3 OCF 返回码和故障恢复 169
- 10 屏障和 STONITH 172**
 - 10.1 屏蔽分类 172
 - 10.2 节点级别屏蔽 172
 - STONITH 设备 173 · STONITH 实施 174
 - 10.3 STONITH 资源和配置 174
 - STONITH 资源配置示例 175
 - 10.4 监视屏蔽设备 178
 - 10.5 特殊的屏蔽设备 179

- 10.6 基本建议 180
- 10.7 更多信息 181
- 11 储存保护和 SBD 182**
 - 11.1 概念概述 182
 - 11.2 手动设置 SBD 概述 184
 - 11.3 要求 184
 - 11.4 SBD 设备的数量 185
 - 11.5 超时计算 185
 - 11.6 设置检查包 187
 - 使用硬件检查包 187 · 使用软件检查包 (softdog) 188
 - 11.7 设置 SBD 与设备 189
 - 11.8 设置无磁盘 SBD 194
 - 11.9 测试 SBD 和屏蔽 195
 - 11.10 储存保护的其他机制 197
 - 配置 sg_persist 资源 197 · 使用 sfex 确保排它激活储存 199
 - 11.11 更多信息 200
- 12 访问控制列表 201**
 - 12.1 要求和先决条件 201
 - 12.2 在群集中启用 ACL 202
 - 12.3 ACL 的基本原理 203
 - 通过 XPath 表达式设置 ACL 规则 203 · 通过缩写设置 ACL 规则 205
 - 12.4 使用 Hawk 配置 ACL2 206
 - 12.5 使用 crmsh 配置 ACL 208

13 网络设备绑定 210

13.1 使用 YaST 配置联接设备 210

13.2 联接从属的热插拔 213

13.3 更多信息 215

14 负载均衡 216

14.1 概念概述 216

14.2 使用 Linux 虚拟服务器配置负载均衡 217
定向器 218 · 用户空间控制器和守护程序 218 · 包的转发 218 · 调度算法 219 · 使用 YaST 设置 IP 负载均衡 219 · 其他设置 224

14.3 使用 HAProxy 配置负载均衡 225

14.4 更多信息 229

15 Geo 群集（多站点群集） 230

16 执行维护任务 231

16.1 关闭群集节点所造成的影响 231

16.2 用于维护任务的不同选项 232

16.3 准备和完成维护工作 233

16.4 将群集置于维护模式 234

16.5 将节点置于维护模式 234

16.6 将节点置于待机模式 235

16.7 将资源置于维护模式 235

16.8 将资源置于不受管理模式 236

16.9 在维护模式下重引导群集节点 237

III 储存和数据复制 238

17 分布式锁管理器 (DLM) 239

- 17.1 用于 DLM 通讯的协议 239
- 17.2 配置 DLM 群集资源 239

18 OCFS2 241

- 18.1 功能和优点 241
- 18.2 OCFS2 包和管理实用程序 242
- 18.3 配置 OCFS2 服务和 STONITH 资源 243
- 18.4 创建 OCFS2 卷 244
- 18.5 装入 OCFS2 卷 246
- 18.6 使用 Hawk2 配置 OCFS2 资源 247
- 18.7 在 OCFS2 文件系统上使用定额 248
- 18.8 更多信息 249

19 GFS2 250

- 19.1 GFS2 包和管理实用程序 250
- 19.2 配置 GFS2 服务和 STONITH 资源 251
- 19.3 创建 GFS2 卷 251
- 19.4 装入 GFS2 卷 253

20 DRBD 255

- 20.1 概念概述 255
- 20.2 安装 DRBD 服务 256

- 20.3 设置 DRBD 服务 257
 - 手动配置 DRBD 258 · 使用 YaST 配置 DRBD 260 · 初始化和格式化 DRBD 资源 263
- 20.4 从 DRBD 8 迁移到 DRBD 9 264
- 20.5 创建堆叠式 DRBD 设备 265
- 20.6 使用资源级屏蔽 266
- 20.7 测试 DRBD 服务 267
- 20.8 监视 DRBD 设备 269
- 20.9 调整 DRBD 270
- 20.10 DRBD 查错 270
 - 配置 270 · 主机名 271 · TCP 端口 7788 271 · DRBD 设备在重引导后损坏 271
- 20.11 更多信息 272
- 21 群集逻辑卷管理器（群集 LVM） 273**
 - 21.1 概念概述 273
 - 21.2 群集式 LVM 的配置 273
 - 创建群集资源 274 · 方案：在 SAN 上将群集 LVM 与 iSCSI 搭配使用 276 · 方案：将群集 LVM 与 DRBD 搭配使用 280
 - 21.3 显式配置合格的 LVM2 设备 282
 - 21.4 从镜像 LV 联机迁移到群集 MD 283
 - 迁移之前的示例设置 283 · 将镜像 LV 迁移到群集 MD 284 · 迁移之后的示例设置 286
 - 21.5 更多信息 287
- 22 群集多设备（群集 MD） 288**
 - 22.1 概念概述 288

- 22.2 创建群集 MD RAID 设备 288
- 22.3 配置资源代理 290
- 22.4 添加设备 290
- 22.5 重新添加暂时发生故障的设备 291
- 22.6 去除设备 291

23 Samba 群集 292

- 23.1 概念概述 292
- 23.2 基本配置 293
- 23.3 加入 Active Directory 域 297
- 23.4 调试和测试群集 Samba 299
- 23.5 更多信息 301

24 使用 Rear (Relax-and-Recover) 实现灾难恢复 302

- 24.1 概念概述 302
 - 创建灾难恢复计划 302 · 灾难恢复意味着什么? 303 · 灾难恢复如何与 Rear 配合工作? 303 · Rear 要求 303 · Rear 版本更新 303 · 针对 Btrfs 的限制 304 · 方案和备份工具 305 · 基本步骤 305
- 24.2 设置 Rear 和您的备份解决方案 306
- 24.3 创建恢复安装系统 308
- 24.4 测试恢复过程 308
- 24.5 从灾难中恢复 309
- 24.6 更多信息 309

IV 附录 310

A 查错 311

- A.1 安装及初始步骤 311
- A.2 日志记录 312
- A.3 资源 314
- A.4 STONITH 和屏蔽 315
- A.5 历史记录 315
- A.6 Hawk2 317
- A.7 杂项 317
- A.8 更多信息 320

B 命名约定 321

C 群集管理工具（命令行） 322

D 在没有 root 访问权的情况下运行群集报告 324

- D.1 创建本地用户帐户 324
- D.2 配置无口令 SSH 帐户 325
- D.3 配置 `sudo` 327
- D.4 生成群集报告 329

术语表 330

E GNU 许可证 336

关于本指南

本指南适用于需要使用 SUSE® Linux Enterprise High Availability Extension 设置、配置和维护群集的管理员。为了能快速且有效地进行配置和管理，产品提供有图形用户界面和命令行界面 (CLI)。本指南介绍了如何通过这两种方法执行关键任务。您可以根据自己的需要选择适当的工具。

本指南分为以下部分：

安装、设置和升级

开始安装和配置群集之前，先熟悉群集的基础知识和体系结构，了解其关键功能和优点的概况。执行下一步之前，要先了解必须满足哪些软硬件要求以及做好哪些准备工作。使用 YaST 安装 HA 群集并进行基本设置。了解如何将群集升级到最新的发行版，或如何更新各个包。

配置和管理

使用 Web 界面 (Hawk2) 或命令行界面 (crmsh) 添加、配置和管理群集资源。为避免未经授权访问群集配置，请定义角色并为角色指派一些用户，以细化访问控制。了解如何使用负载均衡和屏蔽。如果您考虑编写自己的资源代理或修改现有的资源代理，请了解一些有关如何创建不同类型的资源代理的背景信息。

储存和数据复制

SUSE Linux Enterprise High Availability Extension 随附群集感知文件系统 OCFS2 和 GFS2，以及群集逻辑卷管理器（群集 LVM）。要复制数据，可以使用 DRBD*。该工具可让您将高可用性服务的数据从群集的活动节点镜像到其备用节点。此外，群集 Samba 服务器还可为异构环境提供 High Availability 解决方案。

附录

包含常见问题及其解决方案的概述。提供本文档中使用的有关群集、资源和约束的命名约定。包含 HA 特定术语的词汇表。

1 可用文档



注意：在线文档和最新更新

我们的产品文档可从 <https://documentation.suse.com/> 获取，您也可以在此处找到最新更新，以及浏览或下载各种格式的文档。最新的文档更新通常以英语版本提供。

针对本产品提供的文档如下：

《安装和设置快速入门》

本文档会指导您使用 `ha-cluster-bootstrap` 包提供的引导脚本完成最基本的双节点群集的设置。其中包括将虚拟 IP 地址配置为群集资源，以及在共享储存上使用 SBD 作为节点屏蔽机制。

《管理指南》

本指南适用于需要使用 SUSE® Linux Enterprise High Availability Extension 设置、配置和维护群集的管理员。为了能快速且有效地进行配置和管理，产品提供有图形用户界面和命令行界面 (CLI)。本指南介绍了如何通过这两种方法执行关键任务。您可以根据自己的需要选择适当的工具。

《使用 DRBD 和 Pacemaker 的高度可用 NFS 储存》

此文档介绍了如何使用以下组件在双节点群集中设置高度可用的 NFS 储存：

DRBD* (Distributed Replicated Block Device, 分布式复制块设备)、LVM (Logical Volume Manager, 逻辑卷管理器) 和群集资源管理器 Pacemaker。

《Pacemaker 远程快速入门》

本文档将指导您完成具有远程节点或 Guest 节点的高可用性群集设置，这些节点由 Pacemaker 和 `pacemaker_remote` 管理。`pacemaker_remote` 中的 Remote 并不表示实际距离的远近，而是表示节点的特殊状态，即这些节点不运行完整的群集堆栈，因此不属于群集的常规成员。

《Geo 群集快速入门》

Geo 群集可保护全局分布式数据中心范围的工作负载。本文档会指导您使用 `ha-cluster-bootstrap` 包提供的 Geo 引导脚本完成 Geo 群集的基本设置。

《Geo 群集指南》

本文档介绍 Geo 群集及其组件的设置选项和参数，例如投票间票据管理器、特定 Csync2 设置和所需群集资源的配置（以及如何在发生更改时将其传输到其他站点）。了解如何从命令行或使用 Hawk2 Web 界面监视和管理 Geo 群集。

2 提供反馈

欢迎为本文档提供反馈及帮助改进文档！我们提供了多种反馈渠道：

服务请求和支持

有关产品可用的服务和支持选项，请参见 <http://www.suse.com/support/>。

要创建服务请求，需在 SUSE Customer Center 中获取一个订阅。请转到 <https://scc.suse.com/support/requests> 并登录，然后单击新建。

Bug 报告

在 <https://bugzilla.suse.com/> 中报告文档问题。要简化此过程，可以使用本文档 HTML 版本中的标题旁边的报告文档 Bug 链接。这样，就会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。需要一个 Bugzilla 帐户。

贡献

要帮助改进本文档，请使用本文档 HTML 版本中的标题旁边的编辑源代码链接。这些链接会将您转到 GitHub 上的源代码，在其中可以创建拉取请求。需要一个 GitHub 帐户。

有关本文档使用的文档环境的详细信息，请参见 [储存库的 README \(https://github.com/SUSE/doc-sleha/blob/master/README.adoc\)](https://github.com/SUSE/doc-sleha/blob/master/README.adoc)。

邮件

或者，您可以向以下电子邮件地址报告错误以及发送有关本文档的反馈：doc-team@suse.com。请确保反馈中含有文档标题、产品版本和文档发布日期。请引用相关的章节号和标题（或者包含 URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- `tux > command`

可由任何用户（包括 `root` 用户）运行的命令。

- `root # command`

必须使用 `root` 特权运行的命令。您往往还可以在这些命令前加上 `sudo` 命令来运行它们。

- `crm(live)#`

在交互式 `crm` 外壳中执行的命令。有关细节，请参见第 8 章“配置和管理群集资源（命令行）”。

- `/etc/passwd`：目录名称和文件名
- `PLACEHOLDER`： `PLACEHOLDER` 将会替换为实际的值
- `PATH`：环境变量 `PATH`
- `ls`、`--help`：命令、选项和参数
- `user`：用户和组
- `packagename`：包名称
- `Alt`、`Alt - F1`：按键或组合键；这些键以大写形式显示，如在键盘上一样
- 文件、文件 > 另存为：菜单项，按钮
- `amd64, em64t, ipf`：本段仅与体系结构 `amd64`、`em64t` 和 `ipf` 相关。箭头标记文本块的开始位置和结束位置。◁
- 跳舞的企鹅（企鹅一章，↑其他手册）：此内容参见自其他手册中的一章。
- 注意



警告

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。



重要

在继续操作之前，您必须了解的重要信息。



注意

额外信息，例如有关软件版本差异的信息。



提示

有用信息，例如指导方针或实用性建议。

有关群集节点和名称、资源与约束的命名约定概述，请参见附录 B “命名约定”。

4 产品生命周期和支持

SUSE 产品的支持周期长达 13 年。要查看产品的生命周期日期，请参见 <https://www.suse.com/lifecycle/>。

SUSE Linux Enterprise 适用以下生命周期和发行周期：

- SUSE Linux Enterprise Server 的生命周期为 13 年：10 年的标准支持，3 年的扩展支持。
- SUSE Linux Enterprise Desktop 的生命周期为 10 年：7 年的标准支持，3 年的扩展支持。
- 主要版本每 4 年发行一次。服务包每 12-14 个月发行一次。
- 新的 SUSE Linux Enterprise 服务包发行后，SUSE 对以前的服务包的支持会延续 6 个月。

某些产品提供长期服务包支持 (LTSS)。有关我们的支持政策和选项的信息，请参见 <https://www.suse.com/support/policy.html> 和 <https://www.suse.com/support/programs/long-term-service-pack-support.html>。

模块的生命周期、更新策略和更新时限不同于其基础产品。模块包含软件包，是受到完全支持的 SUSE Linux Enterprise High Availability Extension 组件。

4.1 SUSE Linux Enterprise High Availability Extension 的支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 <https://www.suse.com/support/> 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题领域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Linux Enterprise High Availability Extension 将为除以下包外的其他所有包提供 L3 支持：

- 技术预览
- 声音、图形、字体和作品。
- 需要额外客户合同的包。
- 模块 Workstation Extension 随附的某些包仅享受 L2 支持。
- 名称以 `-devel` 结尾的包（包含头文件和类似的开发人员资源）只能同其主包一起接受支持。

SUSE 仅支持使用原始包，即，未发生更改且未重新编译的包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种包、堆栈或功能。随附这些预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈！如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

但是，技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们的功能可能不完备、不稳定，或者在其他方面不适合用于生产。
- 技术预览不受支持。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- 我们随时可能会放弃技术预览。例如，如果 SUSE 发现某个预览不符合客户或市场的需求，或者不能证明它符合企业标准，则可能会放弃该预览。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 <https://www.suse.com/releases/> 上的发行说明。

I 安装、设置和升级

- 1 产品概述 2
- 2 系统要求和建议 14
- 3 安装 High Availability Extension 18
- 4 使用 YaST 群集模块 20
- 5 升级群集和更新软件包 37

1 产品概述

SUSE® Linux Enterprise High Availability Extension 是一种开放源代码群集技术的集成套件。它可让您实施高度可用的物理和虚拟 Linux 群集，避免单一故障点。它可确保关键网络资源的高可用性和可管理性，这些网络资源包括数据、应用程序和服务。因此，它有助于维持业务连续性、保护数据完整性及减少 Linux 关键任务工作负荷的计划外停机时间。

它随附提供必需的监视、消息交换和群集资源管理功能（支持对独立管理的群集资源进行故障转移、故障回复和迁移（负载平衡））。

本章介绍 High Availability Extension 的主要产品功能和优点。您将在本章中找到多个示例群集并了解组成群集的组件。最后一节概述了体系结构，描述了群集内的各体系结构层和进程。

有关 High Availability 群集环境中使用的一些通用术语的解释，请参见[术语表](#)。

1.1 作为扩展提供

High Availability Extension 是以 SUSE Linux Enterprise Server 15 SP2 的扩展的形式提供的。安装 Geo Clustering for SUSE Linux Enterprise High Availability Extension 后，即可不受距离限制使用高可用性群集。

1.2 主要特征

SUSE® Linux Enterprise High Availability Extension 可帮助您保障和管理网络资源的可用性。以下各节重点说明一些关键功能：

1.2.1 各种群集方案

High Availability Extension 支持下列方案：

- 主动/主动配置
- 主动/被动配置：N+1、N+M、N 到 1 和 N 到 M
- 混合物理和虚拟群集，支持将虚拟服务器和物理服务器群集在一起。这可提高服务可用性和资源利用率。
- 本地群集
- 城域群集（“延伸的”本地群集）
- Geo 群集（地理位置分散的群集）

群集最多可包含 32 个 Linux 服务器。使用 `pacemaker_remote` 可扩展群集使之突破此限制，包含更多的 Linux 服务器。如果群集内的一台服务器发生故障，则群集内的任何其他服务器均可重新启动此服务器上的资源（应用程序、服务、IP 地址和文件系统）。

1.2.2 灵活性

High Availability Extension 附带了 Corosync 讯息交换和成员资格层以及 Pacemaker 群集资源管理器。使用 Pacemaker，管理员可持续监视其资源的运行状况和状态，并管理依赖项。可根据高度可配置的规则和策略，自动停止和启动服务。High Availability Extension 允许您根据适合您组织的特定应用程序和硬件基础体系结构对群集进行定制。基于时间的配置使服务可以在特定时间自动迁移回已修复的节点。

1.2.3 储存和数据复制

借助 High Availability Extension，您可以根据需要动态地指派和重指派服务器储存。它支持光纤通道或 iSCSI 储存区域网络 (SAN)。它还支持共享磁盘系统，但这不是必需的。SUSE Linux Enterprise High Availability Extension 还附带有群集感知文件系统 (OCFS2) 和群集式逻辑卷管理器（群集式 LVM2）。如需复制数据，可使用 DRBD* 将高可用性服务的数据从群集的活动节点镜像到其备用节点。此外，SUSE Linux Enterprise High Availability Extension 还支持 CTDB（Cluster Trivial Database，群集普通数据库），这是一种 Samba 群集技术。

1.2.4 支持虚拟环境

SUSE Linux Enterprise High Availability Extension 支持物理和虚拟 Linux 服务器的混合群集。SUSE Linux Enterprise Server 15 SP2 随附了 Xen（一种开放源代码虚拟化超级管理程序）和 KVM（基于内核的虚拟机）。KVM 是一款适用于 Linux 的虚拟化软件，基于硬件虚拟化扩展。High Availability Extension 中的群集资源管理器能够识别、监视和管理虚拟服务器以及物理服务器上正在运行的服务。Guest 系统可作为服务由群集管理。

1.2.5 支持本地、城域和 Geo 群集

SUSE Linux Enterprise High Availability Extension 已扩展，可以支持下列不同的地理方案。Geo Clustering for SUSE Linux Enterprise High Availability Extension 支持分散在不同地理位置的群集（Geo 群集）。

本地群集

一个位置的单个群集（例如，位于一个数据中心内的所有节点）。该群集使用多路广播或单路广播实现节点之间的通讯，并在内部管理故障转移。网络延迟可以忽略。储存通常由所有节点同步访问。

城域群集

使用光纤通道连接所有站点、可跨越多个建筑物或数据中心的单个群集。该群集使用多路广播或单路广播实现节点之间的通讯，并在内部管理故障转移。网络延迟通常很低（约 20 英里的距离 <5 毫秒）。储存频繁复制（镜像或同步复制）。

Geo 群集（多站点群集）

多个地理位置分散的站点，每个站点一个本地群集。站点通过 IP 通讯。站点间的故障转移由更高级别实体协调。Geo 群集需要应对有限网络带宽和高延迟问题。储存异步复制。

各个群集节点之间的地理距离越大，可能影响群集所提供的高可用性的因素就越多。网络延迟、有限带宽以及对储存的访问权是远距离群集面临的主要难题。

1.2.6 资源代理

SUSE Linux Enterprise High Availability Extension 包含大量资源代理来管理资源，如 Apache、IPv4 和 IPv6 等。它还为通用的第三方应用程序（例如 IBM WebSphere Application Server）提供了资源代理。如需产品随附的 Open Cluster Framework (OCF) 资源代理的概述，请根据第 8.1.3 节 “显示有关 OCF 资源代理的信息” 中所述使用 `crm ra` 命令。

1.2.7 用户友好的管理工具

High Availability Extension 附带有一套强大的工具。可使用这些工具进行基本的群集安装和设置，并能实现高效配置和管理：

YaST

常规系统安装和管理的图形用户界面。使用此用户界面，可按《安装和设置快速入门》中所述在 SUSE Linux Enterprise Server 的基础上安装 High Availability Extension。YaST 在 High Availability 类别中还提供以下模块，可帮助您配置群集或各个组件：

- 群集：基本群集设置。有关详细信息，请参见第 4 章 “使用 YaST 群集模块”。
- DRBD：配置分布式复制块设备。
- IP 负载平衡：使用 Linux 虚拟服务器或 HAProxy 配置负载平衡。有关详细信息，请参见第 14 章 “负载平衡”。

Hawk2

您可以在 Linux 或非 Linux 计算机上使用用户友好的 Web 界面来监视和管理高可用性群集。可使用（图形）Web 浏览器从群集内外的任何计算机访问 Hawk2。因此，即便您使用的只是提供极简图形用户界面的系统，也能完美满足您的需求。有关详细信息，请参见第 7 章 “使用 Hawk2 配置和管理群集资源”。

crm 外壳

强大的统一命令行界面，用于配置资源和执行所有监视或管理任务。有关详细信息，请参见第 8 章 “配置和管理群集资源（命令行）”。

1.3 优势

High Availability Extension 可让您将最多 32 个 Linux 服务器配置到一个高可用性群集（HA 群集）中。资源可在群集中的任何节点之间动态切换或移动。可以将资源配置为在发生节点故障时自动进行迁移，也可以手动移动资源以对硬件查错或平衡工作负载。

High Availability Extension 通过商品组件提供高可用性。通过将应用程序和操作合并到群集中降低了成本。High Availability Extension 还可让您集中管理整个群集。您可以调整资源以满足不断变化的工作负载要求（即对群集进行手动“负载平衡”）。允许群集的多个（两个以上）节点共享一个“热备份”也节约了成本。

它还具有另一个同等重要的优点，就是能够潜在地减少计划外服务中断时间及用于软件和硬件维护和升级的计划内中断时间。

实施群集的理由包括：

- 提高可用性
- 改善性能
- 降低操作成本
- 可伸缩性
- 灾难恢复
- 数据保护
- 服务器合并
- 储存合并

通过在共享磁盘子系统上实施 RAID 可获得共享磁盘容错。

以下方案说明了 High Availability Extension 具备的一些优点。

示例群集方案

假设您配置了一个包含三个节点的群集，并在群集内的每个节点上安装了 Web 服务器。群集内的每个节点分别托管两个网站。每个网站的所有数据、图形和网页内容都储存在一个与群集中每个节点都连接的共享磁盘子系统上。下图说明了该系统的结构。

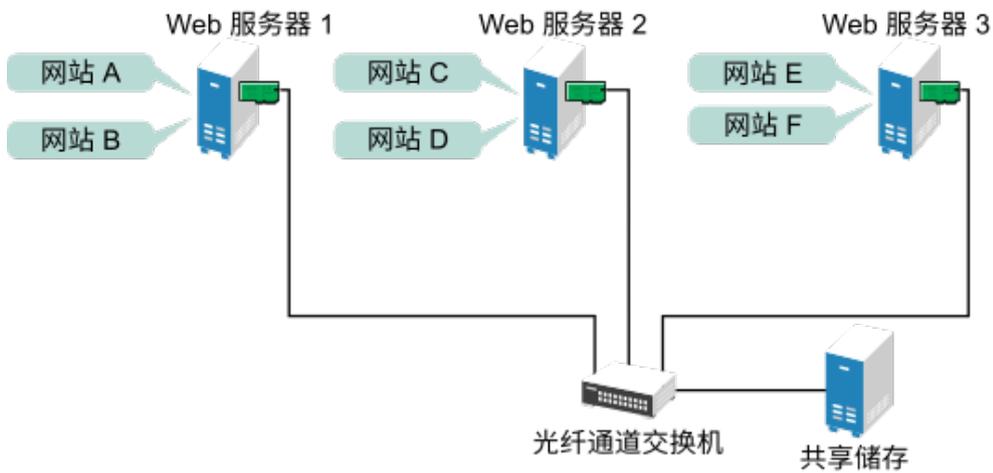


图 1.1：三台服务器的群集

在正常群集操作期间，每个节点都会与群集内的其他节点保持通讯，并会对所有已注册资源执行定期巡回检测以检测是否有故障发生。

假设 Web 服务器 1 出现硬件或软件故障，而依赖此 Web 服务器访问因特网、收发电子邮件和获取信息的用户失去了连接。下图说明了当 Web 服务器 1 出现故障时，资源的移动情况。

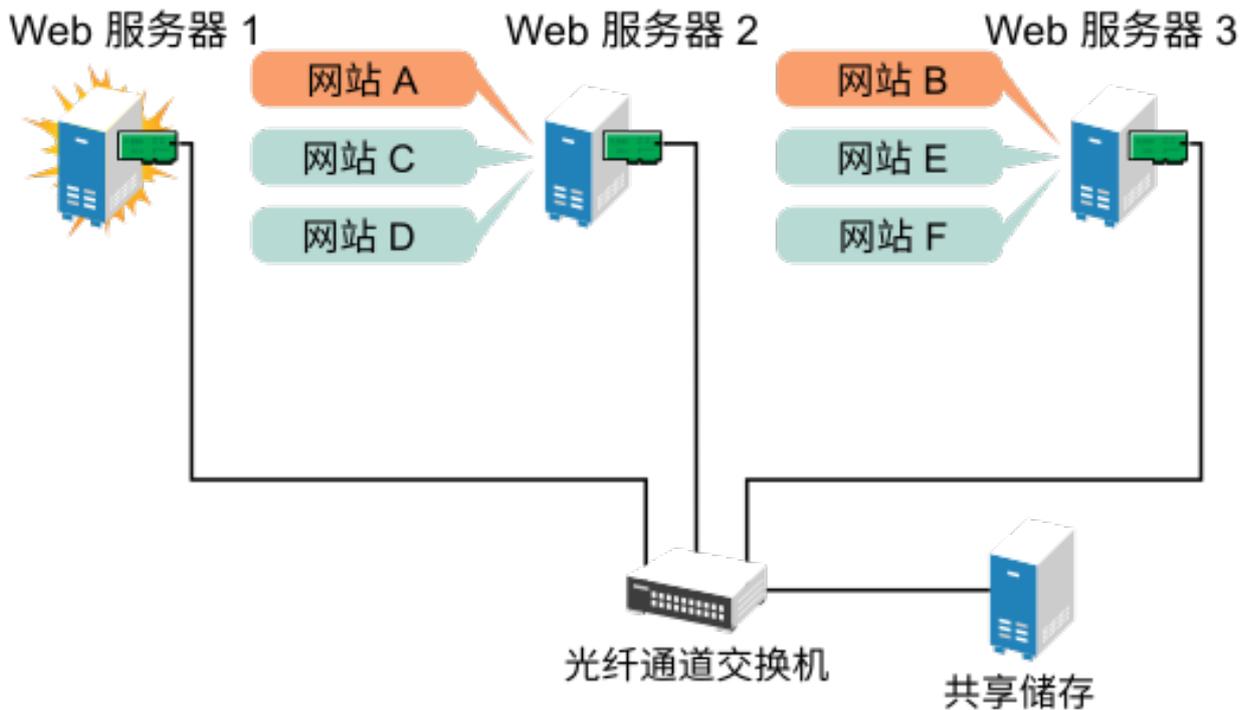


图 1.2：三台服务器的群集（其中一台服务器出现故障后）

网站 A 移至 Web 服务器 2，网站 B 移至 Web 服务器 3。IP 地址和证书也移至 Web 服务器 2 和 Web 服务器 3。

在配置群集时，您决定了在出现故障的情况下，每台 Web 服务器上的网站将移至哪里。在上例中，您已配置将网站 A 移至 Web 服务器 2，将网站 B 移至 Web 服务器 3。这样一来，原来由 Web 服务器 1 处理的工作负载仍可用，并将在所有仍然正常运作的群集成员之间均匀分布。

如果 Web 服务器 1 发生故障，则 High Availability Extension 软件会执行下列操作：

- 检测到故障，并与 STONITH 确认 Web 服务器 1 确实已出现故障。STONITH 是“Shoot The Other Node In The Head”（关闭其他节点）的首字母缩写。它是一种关闭行为异常节点的方式，可防止这些节点在群集中引发问题。
- 将以前安装在 Web 服务器 1 上的共享数据目录重新安装在 Web 服务器 2 和 Web 服务器 3 上。
- 在 Web 服务器 2 和 Web 服务器 3 上重新启动以前运行于 Web 服务器 1 上的应用程序。
- 将 IP 地址传送到 Web 服务器 2 和 Web 服务器 3。

在此示例中，故障转移过程迅速完成，用户在几秒钟之内就可以重新访问 Web 站点信息，而且通常无需重新登录。

现在，假设 Web 服务器 1 的故障已解决，它已恢复到正常工作状态。网站 A 和网站 B 可以自动故障回复（移回）至 Web 服务器 1，或者留在当前所在的服务器上。这取决于您是如何配置它们的资源的。将服务迁移回 Web 服务器 1 会造成一段时间停机。因此，High Availability Extension 也可让您选择延迟迁移，等到只会产生短暂服务中断或不会产生服务中断时再进行迁移。这两种选择都各有优缺点。

High Availability Extension 还提供资源迁移功能。可以根据系统管理的需要将应用程序、网站等资源移动到群集中的其他服务器。

例如，您可以手动将网站 A 或网站 B 从 Web 服务器 1 移至群集内的其他任何一台服务器。此操作的用例包括，对 Web 服务器 1 进行升级或定期维护，或者提高网站的性能或可访问性。

1.4 群集配置：储存

High Availability Extension 的群集配置可能包括共享磁盘子系统，也可能并不包括。共享磁盘子系统可通过高速光纤通道卡、电缆和交换机连接，也可配置为使用 iSCSI。如果有一个节点发生故障，群集中的另一个指定节点就会自动装入之前装入到故障节点上的共享磁盘目录。这样，网络用户就能继续访问共享磁盘子系统上的目录。

！ 重要：带 LVM2 的共享磁盘子系统

使用带 LVM2 的共享磁盘子系统时，必须将该子系统连接到群集中需要访问它的所有服务器。

典型的资源包括数据、应用程序和服务。下图显示典型光纤通道群集配置的可能构成。绿色线表示与以太网电源开关的连接。如此设备便可通过网络控制，并可在 ping 请求失败时重引导节点。

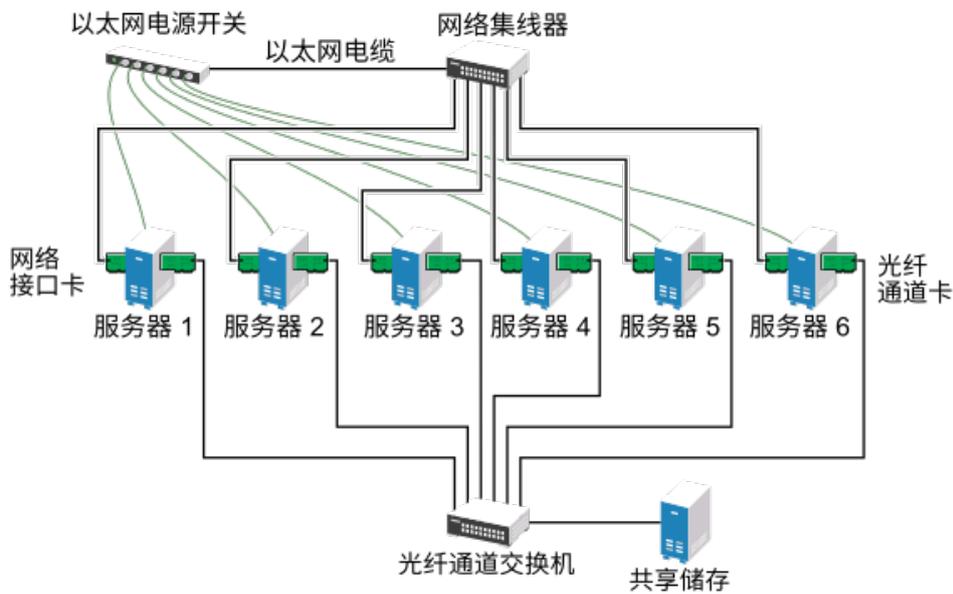


图 1.3：典型的光纤通道群集配置

虽然光纤通道提供的性能最佳，但也可以将群集配置为使用 iSCSI。iSCSI 是除光纤通道外的另一种选择，可用于创建低成本的储存区域网络 (SAN)。下图显示了一个典型的 iSCSI 群集配置。

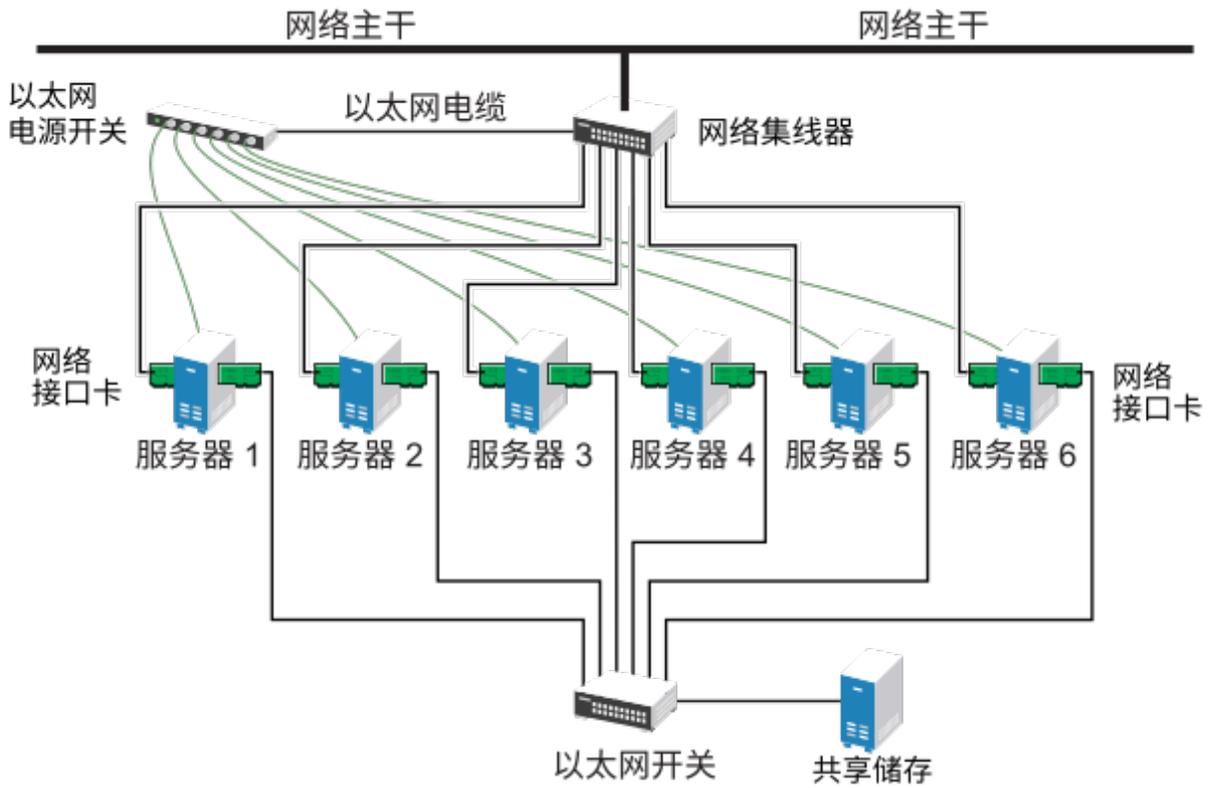


图 1.4：典型的 iSCSI 群集配置

虽然大多数群集都包括共享磁盘子系统，但也可以创建不含共享磁盘子系统的群集。下图显示了一个不含共享磁盘子系统的群集。

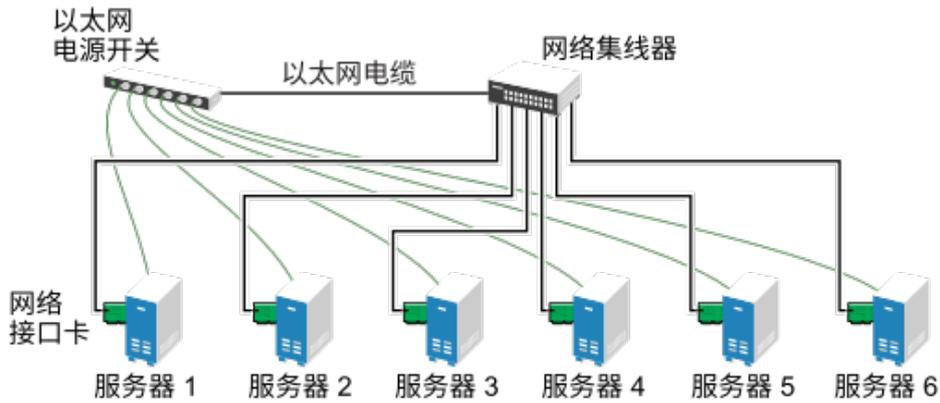


图 1.5：典型的不含共享储存的群集配置

1.5 体系结构

本节简要介绍 High Availability Extension 的体系结构。它提供了有关体系结构组件的信息，并描述了这些组件是如何协同工作的。

1.5.1 体系结构层

High Availability Extension 采用分层式体系结构。图 1.6 “体系结构” 说明了不同的层及其相关的组件。

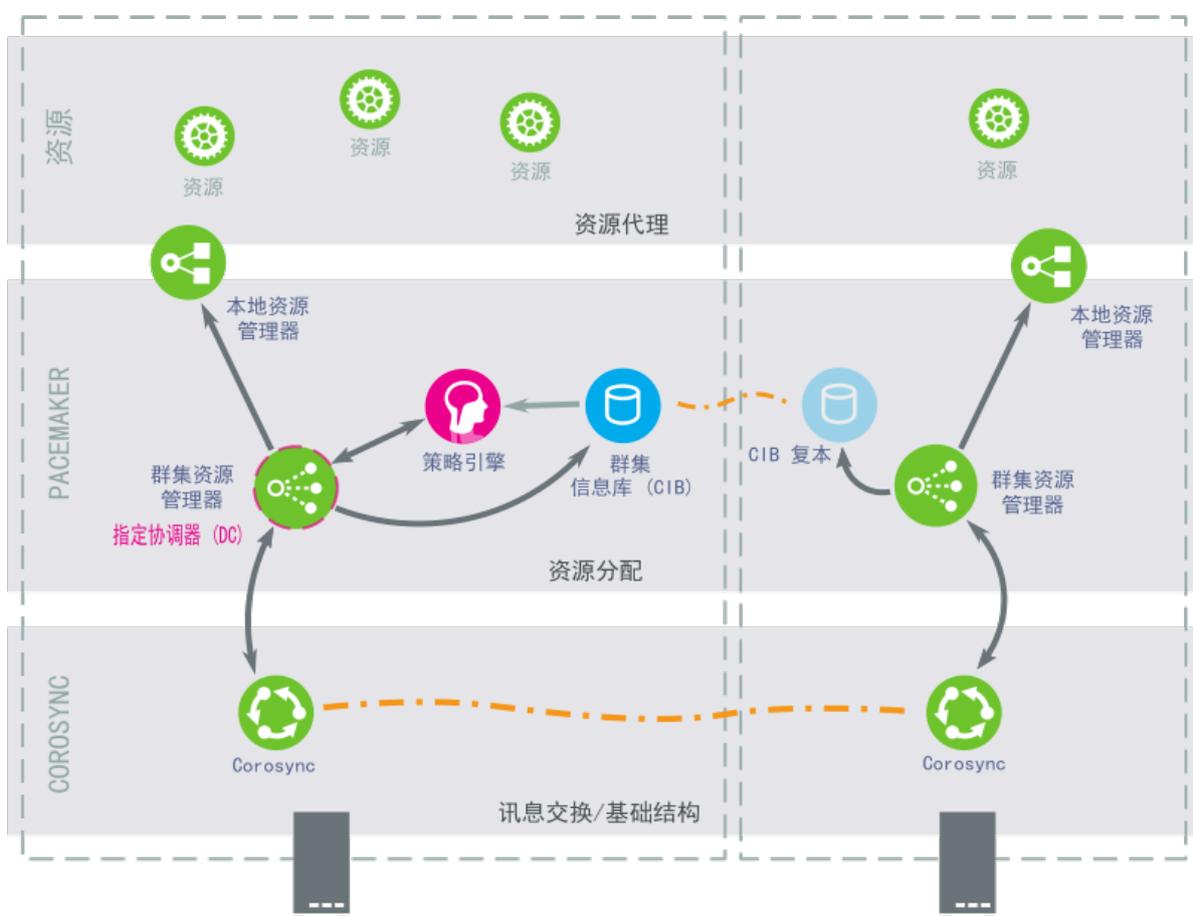


图 1.6：体系结构

1.5.1.1 成员资格和讯息交换层 (Corosync)

此组件提供可靠的讯息交换、成员资格，以及有关群集的仲裁信息。相应的过程由 Corosync 群集引擎（一个组通讯系统）处理。

1.5.1.2 群集资源管理器 (Pacemaker)

用作群集资源管理器的 Pacemaker 是对群集中所发生事件做出反应的“大脑”。它是作为 `pacemaker-controld` 实现的，即，协调所有操作的群集控制器。例如，节点加入或退出群集、资源故障，或者维护等安排的活动均为事件。

本地资源管理器

本地资源管理器位于每个节点上的 Pacemaker 层与资源层之间。它是作为 `pacemaker-execd` 守护程序实现的。通过此守护程序，Pacemaker 可以启动、停止和监视资源。

群集信息数据库 (CIB)

Pacemaker 在每个节点上维护群集信息数据库 (CIB)。CIB 是群集配置的 XML 表示形式（包括群集选项、节点、资源、约束及其相互之间的关系）。CIB 也反映当前群集状态。每个群集节点包含一个在整个群集中同步的 CIB 复本。`pacemaker-based` 守护程序会处理群集配置和状态的读取与写入。

指定协调器 (DC)

DC 是从群集中的所有节点选择出来的。如果当前没有 DC，或者当前的 DC 出于任何原因退出群集，则就会按此方式选择 DC。DC 是群集中唯一可以决定需要在整个群集执行更改（例如节点屏蔽或资源移动）的实体。所有其他节点都从当前 DC 获取他们的配置和资源分配信息。

策略引擎

策略引擎在每个节点上运行，但 DC 上的引擎是活动的引擎。该引擎作为 `pacemaker-schedulerd` 守护程序实现。需要群集转换时，`pacemaker-schedulerd` 会根据当前状态和配置，计算群集的下一预期状态。它会确定需要安排哪些操作来实现下一种状态。

1.5.1.3 资源和资源代理

在高可用性群集中，需要高度可用的服务称为“资源”。资源代理 (RA) 是用于启动、停止和监视群集资源的脚本。

1.5.2 处理流程

`pacemakerd` 守护程序将会启动并监视其他所有相关的守护程序。用于协调所有操作的守护程序 `pacemaker-controld` 在每个群集节点上都有一个实例。Pacemaker 通过选出其中一个实例作为主实例，实现所有群集决策制定的集中化。如果选出的 `pacemaker-controld` 守护程序发生失败，则会建立一个新的守护程序。

群集中执行的许多操作都将导致整个群集的更改。这些操作包括添加或删除群集资源、更改资源约束等等。了解执行这样的操作时群集中会发生的情况是很重要的。

例如，假设您要添加一个群集 IP 地址资源。为此，可以使用 `crm` 外壳或 Web 界面来修改 CIB。不必在 DC 上执行操作，您可以使用群集中任何节点上的任何一种工具，操作即会被转送到 DC。然后 DC 将把此 CIB 更改复制到所有群集节点。

随后，`pacemaker-schedulerd` 将会根据 CIB 中的信息，计算群集的理想状态以及如何实现该状态。它会将指令列表传递给 DC。DC 通过讯息交换/基础结构层发送命令，这些命令将由其他节点上的 `pacemaker-controld` 对等体接收。每个对等体使用自身的本地资源代理执行器（作为 `pacemaker-execd` 实现）来执行资源修改。`pacemaker-execd` 不是群集感知的，它直接与资源代理交互。

所有同级节点将操作的结果报告给 DC。一旦 DC 得出所有必需操作在群集中都已成功执行的结论，群集将回到空闲状态并等待后续事件。如果有任何操作未按计划执行，则会再次调用 `pacemaker-schedulerd`，CIB 中将记录新信息。

在某些情况下，可能需要关闭节点以保护共享数据或完成资源恢复。在 Pacemaker 群集中，节点级别屏蔽的实现为 STONITH。为此，Pacemaker 随附了一个屏蔽子系统 `pacemaker-fenced`。必须将 STONITH 设备配置为群集资源（使用特定的屏蔽代理），因为这样可以监视屏蔽设备。当客户端检测到故障时，会将一个请求发送到 `pacemaker-fenced`，后者再执行屏蔽代理来关闭节点。

2 系统要求和建议

下面的小节说明了系统要求以及 SUSE® Linux Enterprise High Availability Extension 的一些先决条件。此外，还提供了有关群集设置的建议。

2.1 硬件要求

以下列表指出了基于 SUSE® Linux Enterprise High Availability Extension 的群集的硬件要求。这些要求表示最低硬件配置。根据群集的用途，可能会需要其他硬件。

服务器

安装了第 2.2 节“软件要求”中指定的软件的 1 到 32 台 Linux 服务器。

服务器可以是裸机，也可以是虚拟机。两者不要求有相同的硬件（内存、磁盘空间等），但它们必须具有相同的体系结构。不支持跨平台群集。

使用 `pacemaker_remote` 可扩展群集使之突破 32 个节点的限制，包含更多的 Linux 服务器。

通讯通道

每个群集节点至少有两个 TCP/IP 通讯媒体。网络设备必须支持您要用于群集通讯的通讯方式：多路广播或单路广播。通讯媒体应支持 100 Mbit/s 或更高的数据传送速度。对于受支持的群集设置，要求有两个或更多冗余通讯路径。这可通过以下方式实现：

- 网络设备绑定（首选）。
- Corosync 中的另一个通讯通道。

有关细节，请分别参见第 13 章“网络设备绑定”和过程 4.3“定义冗余通讯通道”。

节点屏蔽/STONITH

为了避免发生“节点分裂”的情况，群集需要有节点屏蔽机制。在节点分裂的情况下，群集节点会由于硬件或软件故障或者网络连接断开而分成两个或更多互不相识的组。而屏蔽机制会隔离存在问题的节点（通常的做法是重设置该节点或关闭其电源）。这也称为 STONITH（“Shoot the other node in the head”，关闭其他节点）。节点屏蔽机制可以是物理设备（电源开关），也可以是 SBD（按磁盘 STONITH）等机制再结合检查包。使用 SBD 需要有共享存储。

除非使用了 SBD，否则高可用性群集中的每个节点都必须至少有一个 STONITH 设备。强烈建议每个节点上有多个 STONITH 设备。

重要：不支持无 STONITH 的配置

- 您必须为群集配置节点屏蔽机制。
- 全局群集选项 `stonith-enabled` 和 `startup-fencing` 必须设置为 `true`。如果您更改这些选项，将会失去支持。

2.2 软件要求

将加入群集的所有节点上都至少需安装以下模块和扩展：

- Base System Module 15 SP2
- Server Applications Module 15 SP2
- SUSE Linux Enterprise High Availability Extension 15 SP2

根据您在安装期间选择的 系统角色，默认会安装以下软件模式：

表 2.1：系统角色和已安装的模式

系统角色	软件模式 (YaST/Zypper)
HA 节点	<ul style="list-style-type: none">• 高可用性 (sles_ha)• 增强型基础系统 (enhanced_base)
HA GEO 节点	<ul style="list-style-type: none">• Geo Clustering for High Availability (ha_geo)• 增强型基础系统 (enhanced_base)

注意：极简安装

通过这些系统角色安装只能完成极简安装。如有必要，您可能需要手动添加更多包。

对于原先指派了另一个系统角色的计算机，您需要手动安装 `sles_ha` 或 `ha_geo` 模式及所需的任何其他包。

2.3 储存要求

有些服务需要使用共享储存。如果使用外部 NFS 共享，必须能够从所有群集节点通过冗余通讯路径可靠地访问该共享。

为确保数据的高可用性，我们建议您为群集设置共享磁盘系统（储存区域网络，简称 SAN）。如果使用共享磁盘子系统，请确保符合以下要求：

- 根据制造商的说明正确设置共享磁盘系统并且共享磁盘系统可正确运行。
- 共享磁盘系统中包含的磁盘应配置为使用镜像或 RAID，来为共享磁盘系统增加容错性。
- 如果准备对共享磁盘系统访问使用 iSCSI，则请确保正确配置了 iSCSI 启动器和目标。
- 使用 DRBD* 实施在两台计算机间分发数据的镜像 RAID 系统时，请确保只访问 DRBD 提供的设备，切勿访问备份设备。要利用冗余，可以使用群集剩余组件中所用的相同 NIC。

如果使用 SBD 作为 STONITH 机制，则共享储存还需要满足其他要求。有关细节，请参见第 11.3 节“要求”。

2.4 其他要求和建议

为了实现受支持且有用的高可用性设置，请考虑以下建议：

群集节点数

对于包含两个以上节点的群集，强烈建议使用奇数数目的群集节点，以便具有仲裁。有关仲裁的详细信息，请参见第 6.2 节“仲裁判定”。

时间同步

群集节点必须同步到群集外的 NTP 服务器。自 SUSE Linux Enterprise High Availability Extension 15 起，采用 chrony 作为 NTP 的默认实施。有关详细信息，请参见《SUSE Linux Enterprise Server 15 SP2 管理指南》(<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-ntp.html>) ↗。

如果节点未同步，群集可能无法正常运作。此外，日志文件和群集报告在不进行同步的情况下也很难进行分析。如果使用引导脚本，而 NTP 尚未配置，则系统会提出警告。

网络接口卡 (NIC) 名称

必须在所有节点上都相同。

主机名和 IP 地址

- 使用静态 IP 地址。
- 在 `/etc/hosts` 文件中列出了所有群集节点，包括其完全限定的主机名和简短主机名。群集成员必须能够按名称找到彼此。如果名称不可用，则将无法进行群集内部通讯。

有关 Pacemaker 如何获取节点名称的细节，请参见 http://clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Pacemaker_Explained/s-node-name.html。

SSH

所有群集节点都必须能通过 SSH 相互访问。`crm report`（用于查错）等工具和 Hawk2 的历史记录浏览器要求节点之间采用无口令 SSH 访问方式，否则它们只能从当前节点收集数据。



注意：合规要求

如果无口令 SSH 访问不符合法规要求，您可以使用附录 D “在没有 root 访问权的情况下运行群集报告”中所述的变通方法来运行 `crm report`。

对于历史记录浏览器，目前还没有其他方式可替代无口令登录。

3 安装 High Availability Extension

如果您是首次使用 SUSE® Linux Enterprise High Availability Extension 设置高可用性群集，最简单的方法就是从基本的双节点群集开始设置。您也可以使用双节点群集来运行一些测试。之后，您便可使用 AutoYaST 克隆现有的群集节点来添加更多节点。克隆的节点上会安装相同的包，并具有与原始节点相同的系统配置。

如果要升级运行较低版 SUSE Linux Enterprise High Availability Extension 的现有群集，请参见第 5 章“升级群集和更新软件包”。

3.1 手动安装

要手动安装 High Availability Extension 的包，请参见《安装和设置快速入门》文章。该指南会引导您完成基本双节点群集的设置。

3.2 使用 AutoYaST 进行批量安装和部署

安装并设置双节点群集后，您可以使用 AutoYaST 克隆现有节点并将克隆节点添加到群集，以便扩展群集。

AutoYaST 使用包含安装和配置数据的配置文件。配置文件会告知 AutoYaST 要安装的内容以及如何配置已安装系统，以最终获得一个现成可用的系统。然后可使用此配置文件以各种方式（例如，克隆现有群集节点）进行大批量部署。

有关在各种情况下如何使用 AutoYaST 的详细说明，请参见《AutoYaST Guide for SUSE Linux Enterprise Server 15 SP2》（SUSE Linux Enterprise Server 15 SP2 的 AutoYaST 指南）(<https://documentation.suse.com/sles/15-SP2/html/SLES-all/book-autoyast.html>) 

重要：相同硬件

过程 3.1 “使用 AutoYaST 克隆群集节点” 假设您要将 SUSE Linux Enterprise High Availability Extension 15 SP2 部署到硬件配置完全相同的一组计算机上。

如果您需要在配置不同的硬件上部署群集节点，请参见《SUSE Linux Enterprise 15 SP2 部署指南》“自动安装”一章中的“基于规则的自动安装”一节。

过程 3.1：使用 AUTOYAST 克隆群集节点

1. 确保已正确安装和配置要克隆的节点。有关细节，请参见《SUSE Linux Enterprise High Availability Extension 安装和设置快速入门》或第 4 章“使用 YaST 群集模块”。
2. 按照《SUSE Linux Enterprise 15 SP2 部署指南》中的概要说明进行简单的大批量安装。其中包括以下基本步骤：
 - a. 创建 AutoYaST 配置文件。使用 AutoYaST GUI 基于现有系统配置创建和修改配置文件。在 AutoYaST 中选择 High Availability 模块并单击克隆按钮。如果需要，调整其他模块中的配置，并将生成的控制文件另存为 XML 格式的文件。
如果您已配置 DRBD，也可以在 AutoYaST GUI 中选择并克隆此模块。
 - b. 确定 AutoYaST 配置文件的来源以及要传递到其他节点的安装例程的参数。
 - c. 确定 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 安装数据的来源。
 - d. 确定并设置自动安装的引导方案。
 - e. 通过手动添加参数或创建 `info` 文件，将命令行传递到安装例程。
 - f. 启动并监视自动安装进程。

成功安装克隆节点后，执行以下步骤将克隆节点加入群集中：

过程 3.2：使克隆节点处于联机状态

1. 按第 4.5 节“将配置传送到所有节点”中所述使用 Csync2 将密钥配置文件从已配置的节点传送到克隆节点。
2. 要使节点联机，请按第 4.8 节“使群集联机”中所述在克隆的节点上启动 Pacemaker 服务。

现在克隆节点将加入群集，因为 `/etc/corosync/corosync.conf` 文件已通过 Csync2 应用到克隆节点。CIB 将在群集节点间自动同步。

4 使用 YaST 群集模块

YaST 群集模块可让您手动从头开始设置群集，或修改现有群集的选项。

不过，如果您希望采用自动方式设置群集，请参见《安装和设置快速入门》文章。该指南介绍了如何安装所需的包并会引导您创建基本的双节点群集，该群集是使用 `ha-cluster-bootstrap` 脚本设置的。

您还可结合使用这两种设置方法，例如，使用 YaST 群集设置一个节点，然后使用其中一个引导脚本集成更多节点（或反之）。

4.1 术语定义

下面定义了 YaST 群集模块和本章中使用的多个关键术语。

绑定网络地址 (`bindnetaddr`)

Corosync 管理器应绑定的网络地址。为方便在群集间共享配置文件，Corosync 使用网络接口网络掩码来仅屏蔽用于路由网络的地址位。例如，如果本地接口是 `192.168.5.92` 并且网络掩码是 `255.255.255.0`，则 `bindnetaddr` 将设置为 `192.168.5.0`。例如，如果本地接口是 `192.168.5.92` 并且网络掩码是 `255.255.255.192`，则 `bindnetaddr` 将设置为 `192.168.5.64`。



注意：所有节点的网络地址

由于所有节点将使用相同的 Corosync 配置，请务必使用 `bindnetaddr` 之类的网络地址，而不是特定网络接口地址。

`contrack` 工具

可与内核内连接跟踪系统交互，以便对 iptables 启用有状态包检测。High Availability Extension 使用此工具来同步群集节点之间的连接状态。有关详细信息，请参见 <http://contrack-tools.netfilter.org/>。

Csync2

可用于在群集中的所有节点间（甚至在 Geo 群集间）复制配置文件的同步工具。Csync2 可处理排入同步组的任意数量的主机。每个同步组都有自己的成员主机列表及其包含/排除模式，包含/排除模式定义了同步组中应同步哪些文件。同步组、属于每个组的主机名以及每个组的包含/排除规则均在 Csync2 配置文件 `/etc/csync2/csync2.cfg` 中指定。对于身份验证，Csync2 使用 IP 地址和同步组中的预共享密钥。需要为每个同步组生成一个密钥文件，并将其复制到所有组成员。

有关 Csync2 的更多信息，请参见 <http://oss.linbit.com/csync2/paper.pdf>。

现有群集

术语“现有群集”指的是任何包括至少一个节点的群集。现有群集具有定义通讯通道的基本 Corosync 配置，但它们不一定已有资源配置。

多路广播

一种用于网络内一对多通讯的技术，可用于群集通讯。Corosync 支持多路广播和单路广播。



注意：交换机和多路广播

要使用多路广播进行群集通讯，请确保交换机支持多路广播。

多路广播地址 (`mcastaddr`)

Corosync 管理器使用 IP 地址进行多路广播。IP 地址可以为 IPv4 或 IPv6。如果使用 IPv6 网络，则必须指定节点 ID。可以使用专用网内的任何多路广播地址。

多路广播端口 (`mcastport`)

用于群集通讯的端口。Corosync 使用两个端口：一个用于接收多路广播的指定 `mcastport` 和一个用于发送多路广播的 `mcastport -1`。

Redundant Ring Protocol (RRP)

该协议支持使用多个冗余局域网来从部分或整体网络故障中恢复。这样，只要一个网络运行正常，群集通讯就仍可继续。Corosync 支持 Totem Redundant Ring Protocol。所有参与节点上都强制实施逻辑令牌传递环以确保可靠且有序地传递消息。只有拥有令牌的节点才允许广播消息。

在 Corosync 中定义了冗余通讯通道后，使用 RRP 告知群集如何使用这些接口。RRP 可具有三种模式 (`rrp_mode`):

- 如果设置为 `active`，则 Corosync 将主动使用这两个接口。但是，此模式已弃用。
- 如果设置为 `passive`，Corosync 将选择性地通过可用网络发送消息。
- 如果设置为 `none`，将会禁用 RRP。

单路广播

一种将消息发送到单个网络目标的技术。Corosync 支持多路广播和单路广播。在 Corosync 中，单路广播作为 UDP 单路广播 (UDPU) 实施。

4.2 YaST 群集模块

启动 YaST 并选择高可用性 > 群集。也可以从命令行启动模块：

```
sudo yast2 cluster
```

下面的列表显示了 YaST 群集模块中可用屏幕的概述。它还指出了屏幕是否包含成功设置群集必需的参数，或其参数是否可选。

通讯通道（必需）

允许您定义用于在群集节点之间进行通讯的一个或两个通讯通道。对于传输协议，请使用多路广播 (UDP) 或单路广播 (UDPU)。有关细节，请参见第 4.3 节“定义通讯通道”。



重要：冗余通讯路径

对于受支持的群集设置，要求有两个或更多冗余通讯路径。最好使用网络设备绑定，如第 13 章“网络设备绑定”中所述。

如果不可行，则您需要在 Corosync 中定义另一个通讯通道。

安全性（可选但建议使用）

允许您定义群集的身份验证设置。HMAC/SHA1 身份验证需要使用共享机密来保护和验证讯息。有关细节，请参见第 4.4 节“定义身份验证设置”。

配置 Csync2（可选但建议使用）

Csync2 将帮助您跟踪配置更改，并在群集节点之间保持文件同步。有关细节，请参见第 4.5 节“将配置传送到所有节点”。

配置 `contrackd` (可选)

可让您配置用户空间 `contrackd`。使用 `contrack` 工具为 `iptables` 进行有状态包检测。有关细节，请参见第 4.6 节“同步群集节点间的连接状态”。

服务 (必需)

允许您配置服务以使群集节点联机。定义是否在引导时启动 Pacemaker 服务，以及是否在防火墙中打开节点间通讯所需的端口。有关细节，请参见第 4.7 节“配置服务”。

如果是首次启动群集模块，它会显示为向导，引导您完成进行基本设置所需的所有步骤。否则，请单击左侧面板上的类别，以访问每个步骤的配置选项。



注意：YaST 群集模块中的设置

YaST 群集模块中的一些设置仅适用于当前节点。其他设置可以通过 Csync2 自动传送到所有节点。可在以下各部分中找到有关此配置的详细信息。

4.3 定义通讯通道

为实现群集节点间的成功通讯，请定义至少一个通讯通道。对于传输协议，请分别按过程 4.1 或过程 4.2 中所述使用多路广播 (UDP) 或单路广播 (UDPU)。如果要定义另一个冗余通道 (过程 4.3)，则这两个通讯通道必须使用相同的协议。



注意：公有云：使用单路广播

要在公有云平台中部署 SUSE Linux Enterprise High Availability Extension，请使用单路广播作为传输协议。云平台本身在一般情况下并不支持多路广播。

YaST 通讯通道屏幕中定义的所有设置都会写入到 `/etc/corosync/corosync.conf` 中。`/usr/share/doc/packages/corosync/` 中提供了一些多路广播和单路广播设置的示例文件。

如果您使用的是 IPv4 地址，则节点 ID 可以选填。如果使用的是 IPv6 地址，则必须填写节点 ID。YaST 群集模块提供了一个自动为每个群集节点生成唯一 ID 的选项，用户无需手动为每个节点指定 ID。

过程 4.1：定义第一个通讯通道（多路广播）

使用多路广播时，将对所有群集节点使用相同的 `bindnetaddr`、`mcastaddr` 和 `mcastport`。通过使用相同的多路广播地址，群集中的所有节点都将知晓彼此。对于不同的群集，请使用不同的多路广播地址。

1. 启动 YaST 群集模块，然后切换到通讯通道类别。
2. 将传输协议设为 `多路广播`。
3. 定义绑定网络地址。将此值设置为要用于群集多路广播的子网。
4. 定义多路广播地址。
5. 定义端口。
6. 要为每个群集节点自动生成唯一的 ID，请将自动生成节点 ID 保留为启用状态。
7. 定义群集名称。
8. 输入预期投票数。此值非常重要，Corosync 将使用它来为分区的群集计算法定票数。默认情况下，每个节点有 `1` 张投票。预期投票数必须与群集中的节点数匹配。
9. 确认更改。
10. 如果需要，请按过程 4.3 “定义冗余通讯通道” 中所述在 Corosync 中定义冗余通讯通道。

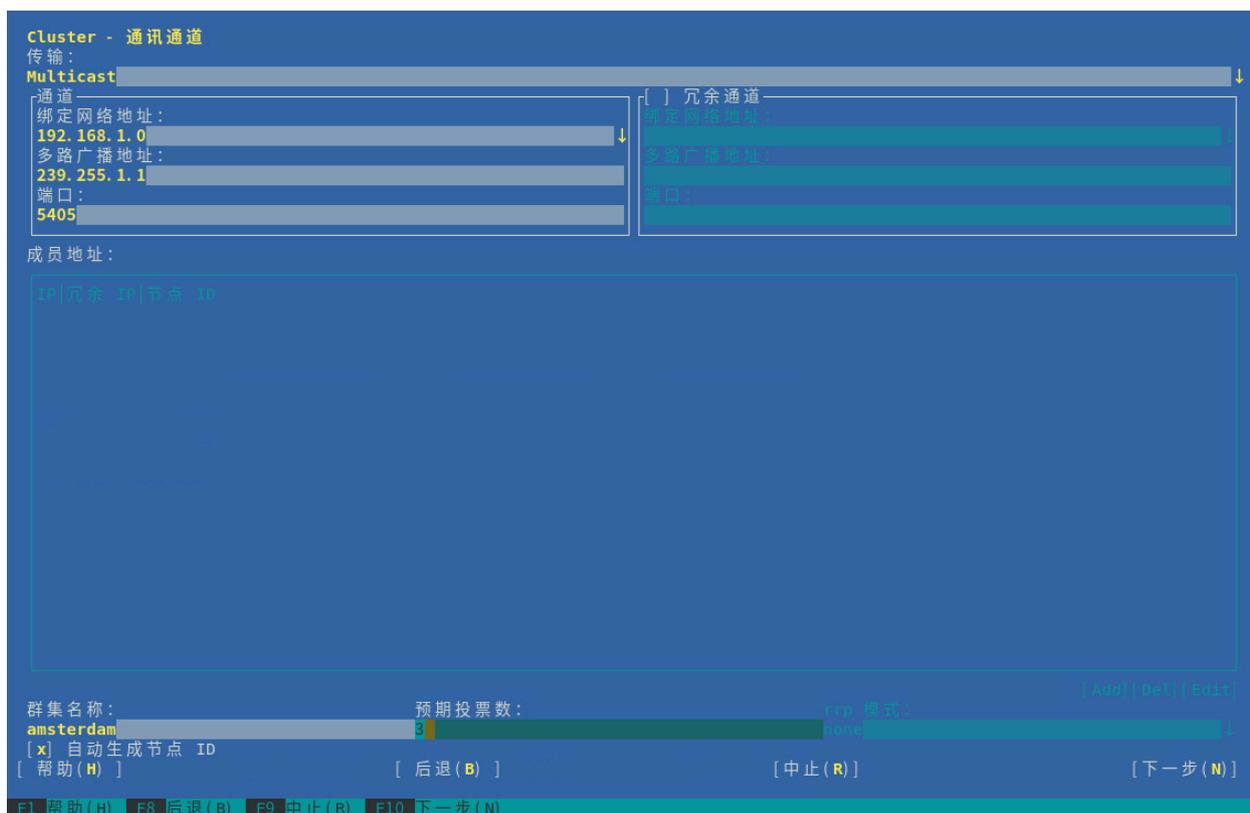


图 4.1：YAST 群集 - 多路广播配置

如果要使用单路广播（而非多路广播）进行群集通讯，请按如下所述继续操作。

过程 4.2：定义第一个通讯通道（单路广播）

1. 启动 YaST 群集模块，然后切换到通讯通道类别。
2. 将传输协议设为 单路广播。
3. 定义端口。
4. 对于单路广播通讯，Corosync 需要知道群集中所有节点的 IP 地址。对于将要加入群集的每个节点，单击添加并输入以下细节：
 - IP 地址
 - 冗余 IP 地址（仅当在 Corosync 中使用了第二个通讯通道时才需要指定）
 - 节点 ID（仅当禁用了自动生成节点 ID 选项时才需要指定）

要修改或删除群集成员的任何地址，请使用编辑或删除按钮。

5. 要为每个群集节点自动生成唯一的 ID，请将自动生成节点 ID 保留为启用状态。
6. 定义群集名称。
7. 输入预期投票数。此值非常重要，Corosync 将使用它来为分区的群集计算法定票数。默认情况下，每个节点有 1 张投票。预期投票数必须与群集中的节点数匹配。
8. 确认更改。
9. 如果需要，请按过程 4.3 “定义冗余通讯通道” 中所述在 Corosync 中定义冗余通讯通道。

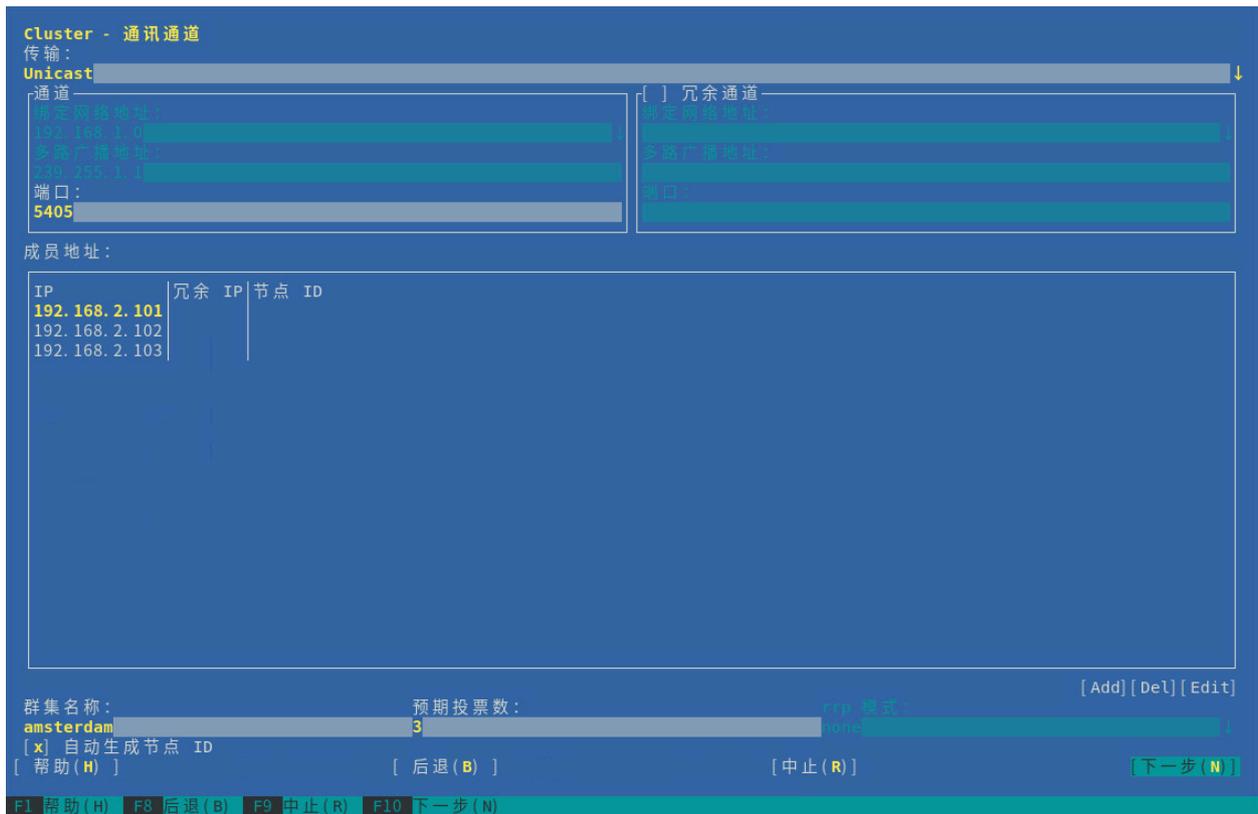


图 4.2：YAST 群集 - 单路广播配置

如果由于任何原因不能使用网络设备绑定，第二个最佳选择就是在 Corosync 中定义冗余通讯通道（次环）。这样就可使用两个物理上分隔的网络进行通讯。如果一个网络发生故障，群集节点仍可通过另一个网络进行通讯。

Corosync 中的另一个通讯通道将会形成另一个令牌传递环。在 `/etc/corosync/corosync.conf` 中，您配置的第一个通道就是主环，其环编号为 0。第二个环（冗余通道）的环编号为 1。

在 Corosync 中定义了冗余通讯通道后，使用 RRP 告知群集如何使用这些接口。有了 RRP，就可以使用两个物理位置分开的网络进行通讯。如果一个网络发生故障，群集节点仍可通过另一个网络进行通讯。

RRP 可以有三种模式：

- 如果设置为 `active`，则 Corosync 将主动使用这两个接口。但是，此模式已弃用。
- 如果设置为 `passive`，Corosync 将选择性地通过可用网络发送消息。
- 如果设置为 `none`，将会禁用 RRP。

过程 4.3：定义冗余通讯通道

! 重要：冗余环和 `/etc/hosts`

如果 Corosync 中配置了多个环，则每个节点都可具有多个 IP 地址。这需要在所有节点的 `/etc/hosts` 文件中反映出来。

1. 启动 YaST 群集模块，然后切换到通讯通道类别。
2. 请激活冗余通道。冗余通道必须使用与所定义的第一个通讯通道相同的协议。
3. 如果使用多路广播，请输入以下参数：要使用的绑定网络地址，冗余通道的多路广播地址和端口。
如果使用单路广播，请定义以下参数：要使用的绑定网络地址及端口。输入将加入群集的所有节点的 IP 地址。
4. 要告知 Corosync 如何以及何时使用其他通道，请选择要使用的 `rrp_mode`：
 - 如果只定义了一个通讯通道，`rrp_mode` 将自动禁用（值 `none`）。
 - 如果设置为 `active`，则 Corosync 将主动使用这两个接口。但是，此模式已弃用。
 - 如果设置为 `passive`，Corosync 将选择性地通过可用网络发送消息。

使用 RRP 时，High Availability Extension 会监视当前环的状态，并在发生故障后自动重新启用冗余环。

或者，使用 `corosync-cfgtool` 手动检查环状态。使用 `-h` 查看可用选项。

5. 确认更改。

4.4 定义身份验证设置

要定义群集的身份验证设置，您可以使用 HMAC/SHA1 身份验证。此方式需要使用共享机密来保护和验证讯息。指定的身份验证密钥（密码）将用于群集中的所有节点。

过程 4.4：启用安全性身份验证

1. 启动 YaST 群集模块，然后切换到安全性类别。
2. 激活启用安全身份验证。
3. 对于新创建的群集，请单击生成身份验证密钥文件。将创建身份验证密钥并将其写入 `/etc/corosync/authkey`。
如果希望当前计算机加入现有群集，则不用生成新的密钥文件。而是将 `/etc/corosync/authkey` 从一个节点复制到当前计算机（手动或使用 Csync2 皆可）。
4. 确认更改。YaST 会将此配置写入 `/etc/corosync/corosync.conf`。

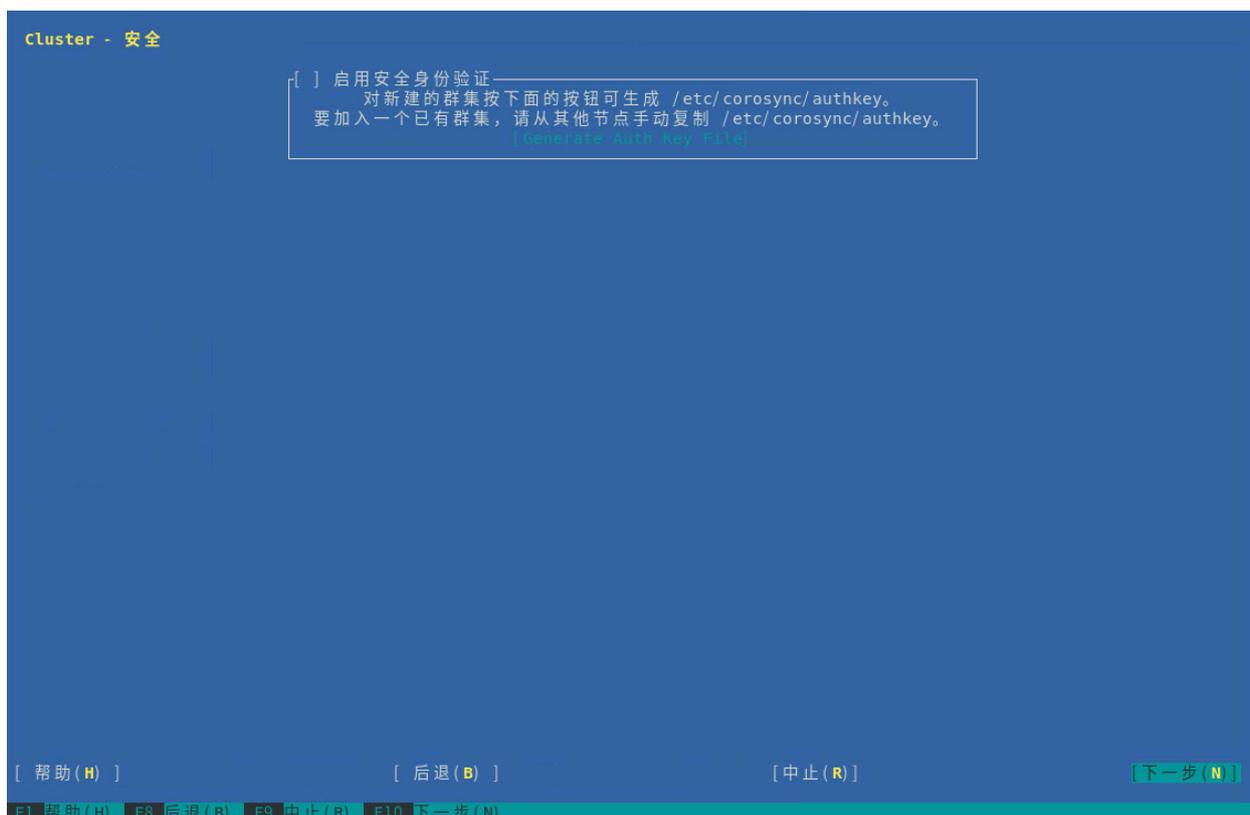


图 4.3：YAST 群集 - 安全性

4.5 将配置传送到所有节点

如果不想将生成的配置文件手动复制到所有节点，可使用 [csync2](#) 工具在群集中的所有节点间进行复制。

这需要以下基本步骤：

1. 使用 YaST 配置 Csync2。
2. 使用 Csync2 同步配置文件。

Csync2 将帮助您跟踪配置更改，并在群集节点之间保持文件同步。

- 可以定义对操作至关重要的文件列表。
- 可以显示这些文件的更改（对于其他群集节点）。

- 可以使用单个命令同步配置的文件。
- 使用 `~/ .bash_logout` 中的一个简单外壳脚本，您可以在从系统注销之前获得更改未同步的提醒。

<http://oss.linbit.com/csync2/> 和 <http://oss.linbit.com/csync2/paper.pdf> 上提供了有关 Csync2 的详细信息。

4.5.1 使用 YaST 配置 Csync2

1. 启动 YaST 群集模块，然后切换到 Csync2 类别。
2. 要指定同步组，请在同步主机组中单击添加，然后输入群集中所有节点的本地主机名。对于每个节点，必须使用 `hostname` 命令返回的确切字符串。



提示：主机名解析

如果您的网络中无法正常进行主机名解析，您也可以为每个群集节点指定主机名与 IP 地址的组合。为此，请使用字符串 `HOSTNAME@IP`，例如 `alice@192.168.2.100`。这样，Csync2 将在连接时使用 IP 地址。

3. 单击生成预共享密钥以创建同步组的密钥文件。密钥文件将写入 `/etc/csync2/key_hagroup`。创建后，必须将其手动复制到群集的所有成员。
4. 要使用通常需要在所有节点间同步的文件填充同步文件列表，请单击添加建议的文件。
5. 要在待同步的文件列表中编辑、添加或删除文件，请使用相应按钮。必须为每个文件输入绝对路径名。
6. 通过单击打开 Csync2 激活 Csync2。随后会执行以下命令，以在引导时自动启动 Csync2：

```
root # systemctl enable csync2.socket
```

7. 确认更改。YaST 会将 Csync2 配置写入 `/etc/csync2/csync2.cfg`。
8. 要立即启动同步过程，请继续第 4.5.2 节 “使用 Csync2 同步更改”。

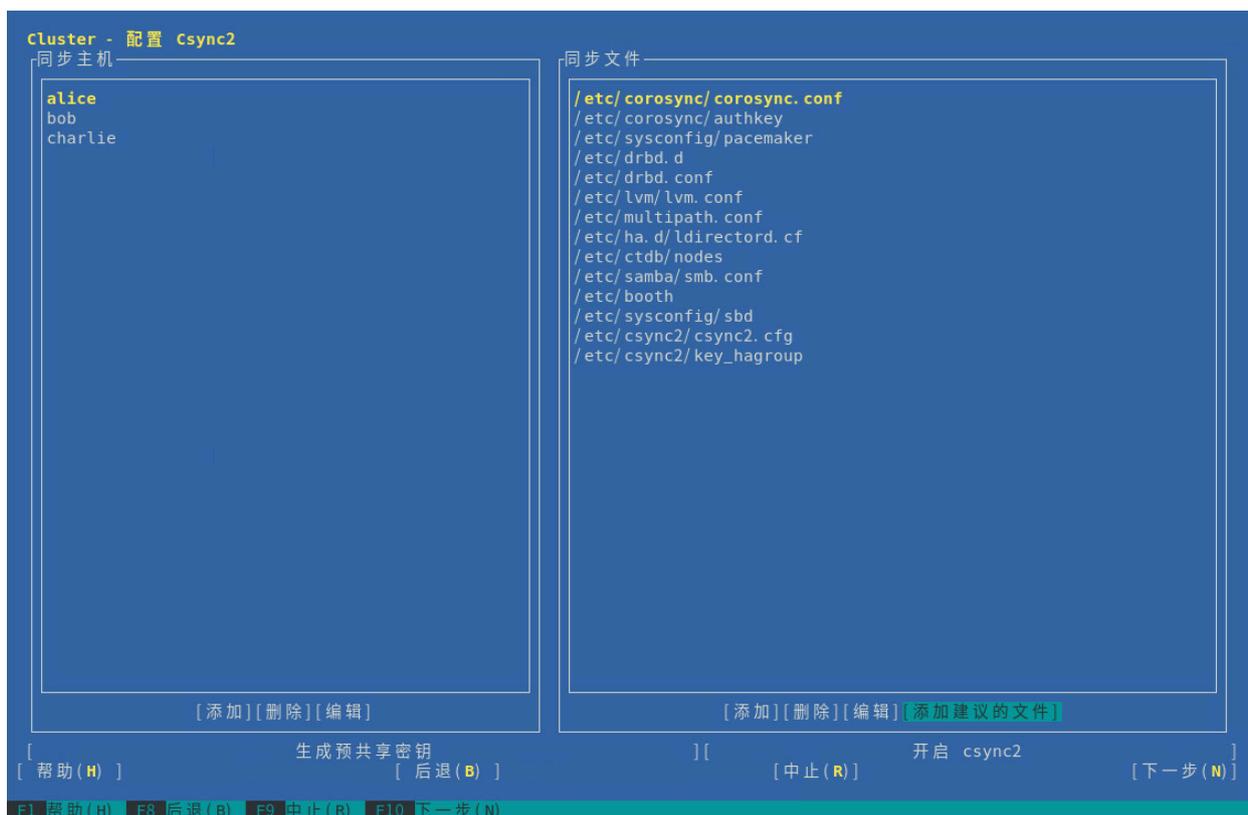


图 4.4：YAST 群集 - CSYNC2

4.5.2 使用 Csync2 同步更改

要成功使用 Csync2 同步文件，必须满足以下要求：

- 同一 Csync2 配置在所有群集节点上均可用。
- 同一 Csync2 身份验证密钥在所有群集节点上均可用。
- 所有群集节点上都必须正在运行 Csync2。

因此，在首次运行 Csync2 之前，需要做好以下准备：

过程 4.5：使用 CSYNC2 准备初始同步

1. 如所述配置后，将文件 `/etc/csync2/csync2.cfg` 第 4.5.1 节 “使用 YaST 配置 Csync2” 手动复制到所有节点。

2. 将执行第 4.5.1 节的步骤 3 时在一个节点上生成的 `/etc/csync2/key_hagroup` 文件复制到群集中的所有节点。它是 Csync2 在进行身份验证时需要使用的文件。但请勿在其他节点上重新生成该文件，因为所有节点上的文件都必须相同。
3. 在所有节点上执行以下命令，以便立即启动服务：

```
root # systemctl start csync2.socket
```

过程 4.6：使用 CSYNC2 同步配置文件

1. 要对所有文件执行一次初始同步，请在要从中复制配置的计算机上执行以下命令：

```
root # csync2 -xv
```

这会将文件推送到其他节点，从而一次同步所有文件。如果所有文件都成功同步，则 Csync2 将完成，不会报错。

如果在其他节点（不仅在当前节点）上对要同步的一个或多个文件进行了修改，则 Csync2 会报告冲突。您将得到类似以下内容的输出：

```
While syncing file /etc/corosync/corosync.conf:  
ERROR from peer hex-14: File is also marked dirty here!  
Finished with 1 errors.
```

2. 如果确信当前节点上的文件版本是“最佳”版本，可以通过强制使用此文件并重新同步来解决冲突：

```
root # csync2 -f /etc/corosync/corosync.conf  
root # csync2 -x
```

有关 Csync2 选项的更多信息，请运行

```
csync2 -help
```



注意：在任何更改后推送同步

Csync2 仅推送更改。它不会在计算机之间连续同步文件。

每次更新需要同步的文件时，都必须将更改推送到其他计算机：在您进行了更改的计算机上运行 `csync2 -xv`。如果对未更改文件的任何其他计算机运行此命令，则不会执行任何操作。

4.6 同步群集节点间的连接状态

要对 iptables 启用有状态包检测，请配置并使用 `contrack` 工具。这需要以下基本步骤：

1. 使用 YaST 配置 `contrackd`。
2. 为 `contrackd` 配置一个资源（类：`ocf`，提供程序：`heartbeat`）。如果使用 Hawk2 添加资源，请使用 Hawk2 推荐的默认值。

配置 `contrack` 工具后，可对 Linux 虚拟服务器使用这些工具，请参见[负载平衡](#)。

过程 4.7：使用 YAST 配置 `contrackd`

使用 YaST 群集模块配置用户空间 `contrackd`。这需要未用于其他通讯通道的专用网络接口。守护程序可随后通过资源代理启动。

1. 启动 YaST 群集模块，然后切换到配置 `contrackd` 类别。
2. 选择专用接口来同步连接状态。会自动检测所选接口的 IPv4 地址并显示在 YaST 中。该地址必须已经配置并且必须支持多路广播。
3. 定义要用于同步连接状态的多路广播地址。
4. 在组号中，定义要将连接状态同步到其中的组的数字 ID。
5. 单击生成 `/etc/contrackd/contrackd.conf` 来创建 `contrackd` 的配置文件。
6. 如果修改了现有群集的任何选项，请确认更改并关闭群集模块。
7. 有关进一步群集配置，请单击下一步并继续[第 4.7 节“配置服务”](#)。



图 4.5：YAST 群集 - contrackd

4.7 配置服务

在 YaST 群集模块中，定义是否在引导节点时启动其上的特定服务。也可使用模块手动启动和停止服务。为使群集节点联机并启动群集资源管理器，Pacemaker 必须作为服务运行。

过程 4.8：启用 PACEMAKER

1. 在 YaST 群集模块中，切换到服务类别。
2. 要在每次引导此群集节点时启动 Pacemaker，请在引导组中选择相应选项。如果在引导组中选择关，则必须在每次引导此节点时手动启动 Pacemaker。要手动启动 Pacemaker，请使用以下命令：

```
root # crm cluster start
```

3. 要立即启动或停止 Pacemaker，请单击相应按钮。

4. 要在防火墙中打开所需的端口以在当前计算机上进行群集通讯，请激活打开防火墙中的端口。
5. 确认更改。请注意，该配置仅应用于当前计算机，而不是所有群集节点。

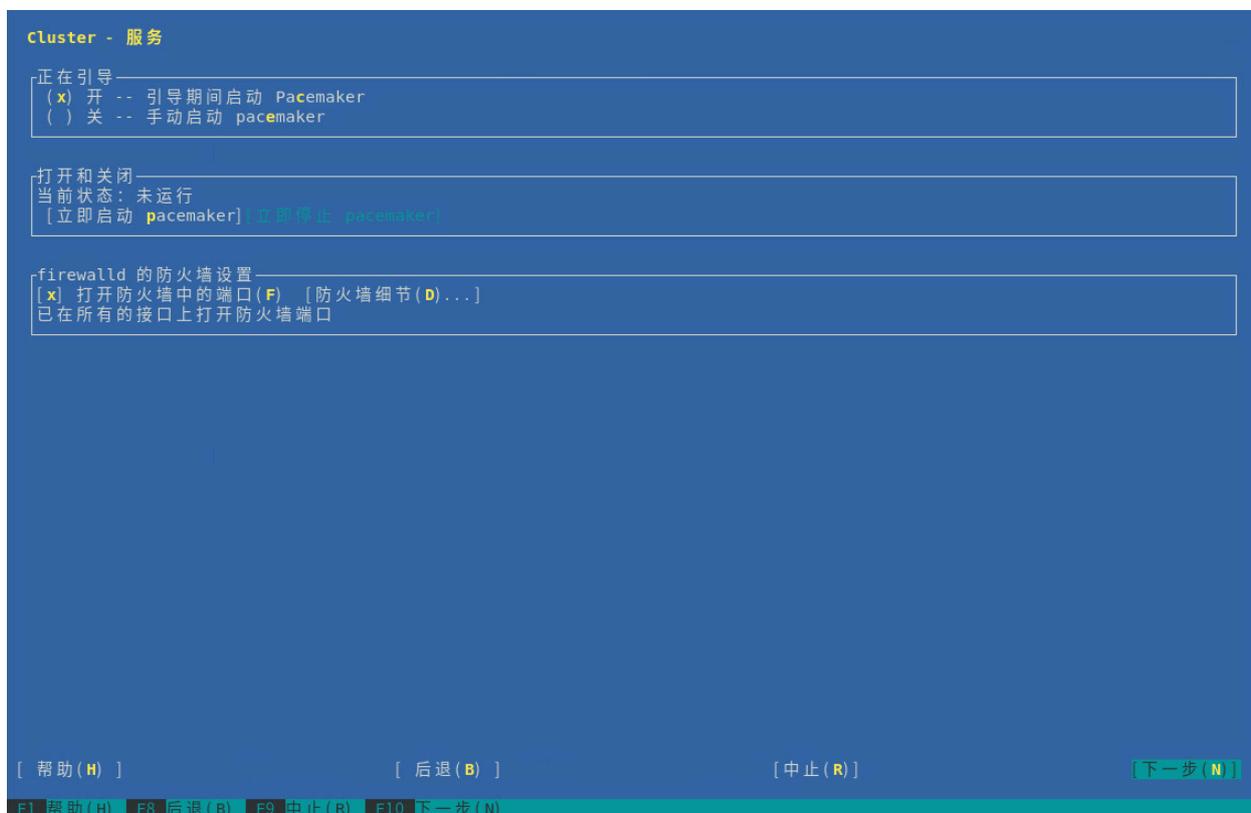


图 4.6：YAST 群集 - 服务

4.8 使群集联机

完成初始群集配置后，在每个群集节点上启动 Pacemaker 服务，以使堆栈联机：

过程 4.9：启动 PACEMAKER 并检查状态

1. 登录到现有节点。
2. 检查服务是否已在运行。

```
root # crm cluster status
```

如果尚未启动，立即启动 Pacemaker：

```
root # crm cluster start
```

3. 对每个群集节点重复上述步骤。
4. 在某个节点上，使用 `crm status` 命令检查群集状态。如果所有节点都联机，则输出应类似于如下内容：

```
root # crm status
Last updated: Thu Jul  3 11:07:10 2014
Last change: Thu Jul  3 10:58:43 2014
Current DC: alice (175704363) - partition with quorum
2 Nodes configured
0 Resources configured

Online: [ alice bob ]
```

此输出表示群集资源管理器已启动，可以管理资源了。

完成基本配置并使节点处于联机状态后，可以开始配置群集资源。使用其中一种群集管理工具，例如 `crm` 外壳 (`crmsh`) 或 `Hawk2`。有关更多信息，请参见第 7 章 “使用 `Hawk2` 配置和管理群集资源” 或第 8 章 “配置和管理群集资源（命令行）”。

5 升级群集和更新软件包

本章介绍两种不同方案：将群集升级为 SUSE Linux Enterprise High Availability Extension 的另一个版本（主要版本或服务包），以及更新群集节点上的单个包。请参见第 5.2 节“将群集升级到最新产品版本”与第 5.3 节“更新群集节点上的软件包”。

如果您要升级群集，请在开始升级之前查看第 5.2.1 节“SLE HA 和 SLE HA Geo 支持的升级路径”和第 5.2.2 节“升级前必须满足的先决条件”。

5.1 术语

下面介绍本章中使用的最重要术语的定义：

主要版本,

正式发布 (GA) 版本

主要版本是一个新的产品版本，增加了新功能和工具并停用了先前弃用的组件。其含有不向后兼容的更改。

群集脱机升级

如果新产品版本包含不可向后兼容的重大更改，则需要通过群集脱机升级来升级群集。需要先使所有节点脱机并将群集作为一个整体进行升级，然后才能使所有节点重新联机。

群集滚动升级

执行群集滚动升级时，每次会升级一个群集节点，此时，群集的其他节点仍在运行中。需将第一个节点脱机，进行升级，然后再使其重新联机以加入群集。然后，需要逐个对其余节点重复上述过程，直到所有群集节点都升级为主要版本。

服务包 (SP)

将几个增补程序合并到便于安装或部署的一个组织体中。服务包是有编号的并通常包含安全性修复、更新、升级或程序增强。

更新

安装某个包的较新次要版本，其中通常包含安全修复和其他重要修复。

升级

安装包或分发包的更新主要版本，引入新功能。另请参见[群集脱机升级](#)与[群集滚动升级](#)。

5.2 将群集升级到最新产品版本

至于支持哪种升级路径以及如何执行升级，取决于当前产品版本，以及您要迁移到的目标版本。

- 群集滚动升级只能在同一个主要版本中进行，例如从产品版本 GA 滚动升级到下一个服务包，以及从一个服务包滚动升级到下一个服务包。
- 要从一个主要版本升级到下一个主要版本（例如，从 SLE HA 12 升级到 SLE HA 15），或者从一个主要版本中的服务包升级到下一个主要版本（例如，从 SLE HA 12 SP3 升级到 SLE HA 15），需要执行群集脱机升级。

第 5.2.1 节提供了 SLE HA (Geo) 的受支持升级路径概述。有关详细信息列中显示您应参考的特定升级文档（还包括基本系统和 Geo Clustering for SUSE Linux Enterprise High Availability Extension）。此文档可从以下位置获得：

- <https://documentation.suse.com/sles> ↗
- <https://documentation.suse.com/sle-ha> ↗

重要：升级后不支持混合群集和逆转

- 不支持在 SUSE Linux Enterprise High Availability Extension 12/SUSE Linux Enterprise High Availability Extension 15 上运行混合群集。
- 完成到产品版本 15 的升级过程后，不支持再还原到产品版本 12。

5.2.1 SLE HA 和 SLE HA Geo 支持的升级路径

升级前版本和目标版本	升级路径	相关细节
SLE HA 11 SP3 到 SLE HA (Geo) 12	群集脱机升级	<ul style="list-style-type: none">• 基本系统：SLES 12 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分• SLE HA：从产品版本 11 升级到 12：群集脱机升级• SLE HA Geo：SLE HA 12 的《Geo 群集快速入门》中的“从 SLE HA (Geo) 11 SP3 升级到 SLE HA Geo 12”一节
从 SLE HA (Geo) 11 SP4 升级到 SLE HA (Geo) 12 SP1	群集脱机升级	<ul style="list-style-type: none">• 基本系统：SLES 12 SP1 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分• SLE HA：从产品版本 11 升级到 12：群集脱机升级• SLE HA Geo：SLE HA 12 SP1 的《Geo 群集快速入门》中的“升级到最新的产品版本”一节
从 SLE HA (Geo) 12 升级到 SLE HA (Geo) 12 SP1	群集滚动升级	<ul style="list-style-type: none">• 基本系统：SLES 12 SP1 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分• SLE HA：执行群集滚动升级• SLE HA Geo：SLE HA 12 SP1 的《Geo 群集快速入门》中的“升级到最新的产品版本”一节

升级前版本和目标版本	升级路径	相关细节
从 SLE HA (Geo) 12 SP1 升级到 SLE HA (Geo) 12 SP2	群集滚动升级	<ul style="list-style-type: none"> • 基本系统：SLES 12 SP2 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分 • SLE HA：执行群集滚动升级 • SLE HA Geo：SLE HA 12 SP2 的《Geo 群集快速入门》中的“升级到最新的产品版本”一节 • DRBD 8 到 DRBD 9：从 DRBD 8 迁移到 DRBD 9
从 SLE HA (Geo) 12 SP2 升级到 SLE HA (Geo) 12 SP3	群集滚动升级	<ul style="list-style-type: none"> • 基本系统：SLES 12 SP3 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分 • SLE HA：执行群集滚动升级 • SLE HA Geo：SLE HA 12 SP3 的《Geo 群集指南》中的“升级到最新的产品版本”一节
从 SLE HA (Geo) 12 SP3 升级到 SLE HA (Geo) 12 SP4	群集滚动升级	<ul style="list-style-type: none"> • 基本系统：SLES 12 SP4 的《部署指南》中的“更新和升级 SUSE Linux Enterprise”部分 • SLE HA：执行群集滚动升级 • SLE HA Geo：SLE HA 12 SP4 的《Geo 群集指南》中的“升级到最新的产品版本”一节
从 SLE HA (Geo) 12 SP3 到 SLE HA (Geo) 15	群集脱机升级	<ul style="list-style-type: none"> • 基本系统：SLES 15 的《升级指南》 • SLE HA：从产品版本 12 升级到 15：群集脱机升级

升级前版本和目标版本	升级路径	相关细节
		<ul style="list-style-type: none"> • SLE HA Geo: 《Geo 群集指南》, 第 10 章 “升级到最新产品版本” • 群集式 LVM: 从镜像 LV 联机迁移到群集 MD
从 SLE HA (Geo) 15 SP4 升级到 SLE HA (Geo) 12 SP1	群集脱机升级	<ul style="list-style-type: none"> • 基础系统: SLES 15 SP1 的《升级指南》 • SLE HA: 从产品版本 12 升级到 15: 群集脱机升级 • SLE HA Geo: 《Geo 群集指南》, 第 10 章 “升级到最新产品版本” • 群集式 LVM: 从镜像 LV 联机迁移到群集 MD
从 SLE HA (Geo) 15 升级到 SLE HA (Geo) 15 SP1	群集滚动升级	<ul style="list-style-type: none"> • 基础系统: SLES 15 SP1 的《升级指南》 • SLE HA: 执行群集滚动升级 • SLE HA Geo: 《Geo 群集指南》, 第 10 章 “升级到最新产品版本”
从 SLE HA (Geo) 15 SP1 升级到 SLE HA (Geo) 15 SP2	群集滚动升级	<ul style="list-style-type: none"> • 基础系统: SLES 15 SP2 的《升级指南》 • SLE HA: 执行群集滚动升级 • SLE HA Geo: 《Geo 群集指南》, 第 10 章 “升级到最新产品版本”

5.2.2 升级前必须满足的先决条件

备份

确保系统备份为最新的且可恢复。

测试

请先在群集设置的临时实例上测试升级过程，然后再在生产环境中执行该过程。这样，您可以估算出维护期所需的时间段。这还有助于检测和解决可能会出现的任何意外问题。

5.2.3 群集脱机升级

本节内容适用于以下方案：

- 从 SLE HA 11 SP3 升级到 SLE HA 12- 有关细节，请参见过程 5.1 “从产品版本 11 升级到 12：群集脱机升级”。
- 从 SLE HA 11 SP4 升级到 SLE HA 12 SP1 - 有关细节，请参见过程 5.1 “从产品版本 11 升级到 12：群集脱机升级”。
- 从 SLE HA 12 SP3 升级到 SLE HA 15 - 有关细节，请参见过程 5.2 “从产品版本 12 升级到 15：群集脱机升级”。
- 从 SLE HA 12 SP4 升级到 SLE HA 15 SP1 - 有关细节，请参见过程 5.2 “从产品版本 12 升级到 15：群集脱机升级”。

如果您的群集仍旧基于早期的产品版本而不是上面所列的版本，请先将它升级到 SLES 和 SLE HA 的某个版本，而该版本可用作升级到所需目标版本的源。

过程 5.1：从产品版本 11 升级到 12：群集脱机升级

High Availability Extension 12 群集堆栈的各个组件包含了重大更改（例如 `/etc/corosync/corosync.conf`、OCFS2 的磁盘格式）。因此，不支持从任何 SUSE Linux Enterprise High Availability Extension 11 版本进行 群集滚动升级。必须将所有群集节点脱机，并根据下面所述将群集作为一个整体升级。

1. 登录到每个群集节点，并使用以下命令停止群集堆栈：

```
root # rcopenais stop
```

2. 将每个群集节点都升级到 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 的所需目标版本 — 请参见第 5.2.1 节 “SLE HA 和 SLE HA Geo 支持的升级路径”。
3. 完成升级过程后，请重引导装有 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 升级版的每个节点。

4. 如果在群集设置中使用了 OCFS2，请执行以下命令以更新设备上的结构：

```
root # o2cluster --update PATH_TO_DEVICE
```

它会为磁盘添加额外参数。SUSE Linux Enterprise High Availability Extension 12 和 12 SPx 随附的已更新 OCFS2 版本需要这些参数。

5. 要更新 Corosync 2 的 `/etc/corosync/corosync.conf`，请执行以下操作：

- a. 登录到某个节点，然后启动 YaST 群集模块。
- b. 切换到通讯通道类别并输入以下新参数的值：群集名称和预期投票数。有关细节，请分别参见过程 4.1 “定义第一个通讯通道（多路广播）” 或过程 4.2 “定义第一个通讯通道（单路广播）”。

如果 YaST 检测到对 Corosync 2 无效或缺失的任何其他选项，它会提示您进行更改。

- c. 确认在 YaST 中所做更改。YaST 会将配置写入 `/etc/corosync/corosync.conf`。
- d. 如果为群集配置了 Csync2，请使用以下命令将更新的 Corosync 配置推送到其他群集节点：

```
root # csync2 -xv
```

有关 Csync2 的细节，请参见第 4.5 节 “将配置传送到所有节点”。

或者，也可以通过将 `/etc/corosync/corosync.conf` 手动复制到所有群集节点，来同步更新的 Corosync 配置。

6. 登录到每个节点，并使用以下命令启动群集堆栈：

```
root # crm cluster start
```

7. 使用 `crm status` 或 Hawk2 检查群集状态。

8. 将以下服务配置为在引导时启动：

```
root # systemctl enable pacemaker
root # systemctl enable hawk
root # systemctl enable sbd
```



注意：升级 CIB 语法版本

有时，新功能只能在最新的 CIB 语法版本中使用。升级到新的产品版本时，默认不会升级 CIB 语法版本。

1. 使用以下命令检查版本：

```
cibadmin -Q | grep validate-with
```

2. 使用以下命令升级到最新的 CIB 语法版本：

```
root # cibadmin --upgrade --force
```

过程 5.2：从产品版本 12 升级到 15：群集脱机升级



重要：从头开始安装

如果您决定从头开始安装群集节点（而不是升级它们），请参见第 2.2 节“软件要求”以获取 SUSE Linux Enterprise High Availability Extension 15 SP2 所需的模块列表。有关模块、扩展及相关产品的详细信息，请参见 SUSE Linux Enterprise Server 15 的发行说明。可从 <https://www.suse.com/releasenotes/> 访问这些文档。

1. 在开始脱机升级到 SUSE Linux Enterprise High Availability Extension 15 之前，请如**注意：升级 CIB 语法版本**中所述手动升级当前群集中的 CIB 语法。
2. 登录到每个群集节点，并使用以下命令停止群集堆栈：

```
root # crm cluster stop
```

3. 将每个群集节点都升级到 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 的所需目标版本 — 请参见第 5.2.1 节 “SLE HA 和 SLE HA Geo 支持的升级路径”。
4. 完成升级过程后，请登录每个节点，并引导装有 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 升级版的每个节点。
5. 如果您使用群集 LVM，则需要从 `clvmd` 迁移到 `lvmlockd`。请参见 `lvmlockd` 的手册页中的 `changing clvm VG to lockd VG`（将 `clvm VG` 更改为 `lockd VG`）部分，另请参见第 21.4 节 “从镜像 LV 联机迁移到群集 MD”。
6. 使用以下命令启动群集堆栈：

```
root # crm cluster start
```
7. 使用 `crm status` 或 Hawk2 检查群集状态。

5.2.4 群集滚动升级

本节内容适用于以下方案：

- 从 SLE HA 12 升级到 SLE HA 12 SP1
- 从 SLE HA 12 SP1 升级到 SLE HA 12 SP2
- 从 SLE HA 12 SP2 升级到 SLE HA 12 SP3
- 从 SLE HA 12 SP3 升级到 SLE HA 12 SP4
- 从 SLE HA 15 升级到 SLE HA 15 SP1
- 从 SLE HA 15 SP1 升级到 SLE HA 15 SP2



警告：活动的群集堆栈

在开始升级某个节点之前，请停止该节点上的群集堆栈。

如果在软件更新期间，群集资源管理器所在节点处于活动状态，可能会导致不可预料的结果，例如活动的节点被屏蔽。

！ 重要：群集滚动升级的时间限制

只有在将所有群集节点都升级到最新产品版本之后，最新产品版本提供的新功能才可用。在群集滚动升级期间，只有较短的一段时间支持混合版本群集升级。请在一周内完成群集滚动升级。

过程 5.3：执行群集滚动升级

1. 以 `root` 用户身份登录要升级的节点，并停止群集堆栈：

```
root # crm cluster stop
```

2. 升级到 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise High Availability Extension 的所需目标版本。要了解各个升级过程的细节，请参见第 5.2.1 节“SLE HA 和 SLE HA Geo 支持的升级路径”。
3. 在升级后的节点上启动群集堆栈，使该节点重新加入群集：

```
root # crm cluster start
```

4. 使下一个节点处于脱机状态，并对此节点重复上述过程。
5. 使用 `crm status` 或 Hawk2 检查群集状态。

如果检测到您的群集节点有不同的 CRM 版本，Hawk2 状态屏幕还会显示一条警告。

5.3 更新群集节点上的软件包

🚫 警告：活动的群集堆栈

启动某节点的包更新之前，请停止该节点上的群集堆栈，或将该节点置于维护模式，具体取决于群集堆栈是否受影响。有关详细信息，请参见步骤 1。

如果在软件更新期间，群集资源管理器所在节点处于活动状态，可能会导致不可预料的结果，例如活动的节点被屏蔽。

1. 在节点上安装任何包更新之前，请先确认以下问题：

- 这种更新是否会影响属于 SUSE Linux Enterprise High Availability Extension 或 Geo Clustering for SUSE Linux Enterprise High Availability Extension 的任何包？如果是，请先在节点上停止群集堆栈，然后再开始软件更新：

```
root # crm cluster stop
```

- 包更新操作是否需要重引导计算机？如果是，请先在节点上停止群集堆栈，然后再开始软件更新：

```
root # crm cluster stop
```

- 如果不属于上述任何一种情况，则不需要停止群集堆栈。在这种情况下，请先将节点置于维护模式，然后再开始软件更新：

```
root # crm node maintenance NODE_NAME
```

有关维护模式的更多细节，请参见第 16.2 节 “用于维护任务的不同选项”。

2. 使用 YaST 或 Zypper 来安装包更新。

3. 在成功安装更新后：

- 启动相应节点上的群集堆栈（如果在执行步骤 1 时已将它停止）：

```
root # crm cluster start
```

- 或者去除维护标志，使节点恢复正常模式：

```
root # crm node ready NODE_NAME
```

4. 使用 `crm status` 或 Hawk2 检查群集状态。

5.4 更多信息

有关您要升级到的产品的任何更改和新功能的详细信息，请参见其发行说明，所在网址为 <https://www.suse.com/releases/notes/>。

II 配置和管理

- 6 配置和管理基础 49
- 7 使用 Hawk2 配置和管理群集资源 84
- 8 配置和管理群集资源（命令行） 134
- 9 添加或修改资源代理 168
- 10 屏障和 STONITH 172
- 11 储存保护和 SBD 182
- 12 访问控制列表 201
- 13 网络设备绑定 210
- 14 负载均衡 216
- 15 Geo 群集（多站点群集） 230
- 16 执行维护任务 231

6 配置和管理基础

HA 群集的主要目的是管理用户服务。Apache Web 服务器或数据库便是一种典型的用户服务。从用户角度来看，服务就是在客户的要求下执行某些操作。但对群集来说，服务只是可以启动或停止的资源，其本质与群集无关。

在本章中，我们将介绍一些配置资源和管理群集时需要了解的基本概念。以下章节介绍如何使用 High Availability Extension 提供的每种管理工具执行主配置和管理任务。

6.1 用例方案

群集一般分为以下两种类别：

- 双节点群集
- 包含两个以上节点的群集。这通常表示节点数是奇数。

添加不同的拓扑可以衍生不同的用例。下面是最常见的用例：

位于一个位置的双节点群集

配置： FC SAN 或类似的共享储存，第 2 层网络。

使用方案： 嵌入式群集，注重服务的高可用性，而不是实现数据冗余来进行数据复制。例如，此类设置可用于无线电台或装配生产线控制器。

位于两个位置的双节点群集（使用最广泛）

配置： 对称的延伸群集，FC SAN，以及跨两个位置的第 2 层网络。

使用方案： 典型的延伸群集，注重服务的高可用性和本地数据冗余。用于数据库和企业资源规划。过去数年来最流行的设置之一。

位于三个位置的奇数数目的节点

配置： $2 \times N + 1$ 个节点，FC SAN 跨两个主要位置第三个辅助站点不部署 FC SAN，而是充当多数仲裁者。第 2 层网络至少跨两个主要位置。

使用方案：典型的延伸群集，注重服务的高可用性和数据冗余。例如数据库和企业资源规划。

6.2 仲裁判定

当一个或多个节点与群集的剩余节点之间的通讯失败时，即会发生群集分区。这些节点只能与同一分区中的其他节点通讯，并不知道被隔开的节点的存在。如果群集分区具有多数节点（或投票），则将其定义为具有仲裁（是“具有法定票数的”）。通过仲裁计算来获得此结果。要实现屏蔽，就必须具有仲裁。

SUSE Linux Enterprise High Availability Extension 11 与 SUSE Linux Enterprise High Availability Extension 15 的仲裁计算方式有所不同。在 SUSE Linux Enterprise High Availability Extension 11 中，仲裁由 Pacemaker 计算。从 SUSE Linux Enterprise High Availability Extension 12 开始，Corosync 可以直接处理双节点群集的仲裁，而无需更改 Pacemaker 配置。

仲裁计算方式受以下因素的影响：

群集节点数

为使服务保持运行状态，包含两个以上节点的群集依赖法定票数（多数票决）来解决群集分区。根据以下公式，您可以计算群集正常运行所需的最小工作节点数目：

$$N \geq C/2 + 1$$

`N = minimum number of operational nodes`

`C = number of cluster nodes`

例如，五节点群集至少需要三个工作节点（或两个可以故障转移的节点）。

我们强烈建议使用双节点群集或奇数数目的群集节点。双节点群集适合跨两个站点的延伸设置。所含节点数为奇数的群集可以构建于单个站点上，也可以分散在三个站点之间。

Corosync 配置

Corosync 是一个讯息交换和成员资格层，具体请参见第 6.2.4 节“双节点群集的 Corosync 配置”和第 6.2.5 节“N 节点群集的 Corosync 配置”。

6.2.1 全局群集选项

全局群集选项控制群集在遇到特定情况时的行为方式。它们被分成若干组，可通过 Hawk2 和 `crm` 外壳之类的群集管理工具来查看和修改。

通常可保留预定义值。但为了使群集的关键功能正常工作，需要在进行基本群集设置后调整以下参数：

- 全局选项 `no-quorum-policy`
- 全局选项 `stonith-enabled`

6.2.2 全局选项 `no-quorum-policy`

此全局选项定义在群集分区不具有法定票数（分区不具有多数节点投票）时应执行的操作。

允许的值有：

ignore

将 `no-quorum-policy` 设置为 `ignore` 可使群集的行为如同它具有仲裁一样。它仍会进行资源管理。

在 SLES 11 中，建议对双节点群集使用此设置。此选项从 SLES 12 开始已过时。Corosync 依据配置和条件为群集节点或单个节点提供“仲裁” — 或者不提供仲裁。对于双节点群集，唯一有意义的行为就是始终像失去仲裁一样来做出反应。第一个步骤始终应该是尝试屏蔽丢失的节点。

freeze

如果失去法定票数，群集分区将会冻结。继续进行资源管理：正在运行的资源不会停止（但可能重新启动以响应监视事件），但不会启动受影响分区中的任何其他资源。

如果群集中的某些资源依赖于与其他节点的通讯（例如，OCFS2 装入），建议对此类群集使用此设置。在这种情况下，默认设置 `no-quorum-policy=stop` 没有任何作用，因为它将导致以下情况：在对等节点不可到达时将无法停止这些资源。反之，尝试停止这些资源最终将超时并导致 `stop failure`，进而触发升级恢复和屏蔽。

stop (默认值)

如果失去法定票数，受影响群集分区中的所有资源都将以一种有序的方式停止。

suicide

如果失去法定票数，受影响群集分区中的所有节点都将被屏蔽。此选项只能与 SBD 结合使用，具体请参见第 11 章“储存保护和 SBD”。

6.2.3 全局选项 stonith-enabled

此全局选项定义是否要应用屏蔽，以允许 STONITH 设备关闭发生故障的节点以及无法停止其资源的节点。默认情况下，此全局选项设置为 `true`，因为对于常规的群集操作，有必要使用 STONITH 设备。根据默认值，如果未定义 STONITH 资源，则群集将拒绝启动任何资源。

如果出于任何原因而需要禁用屏蔽，请将 `stonith-enabled` 设置为 `false`，但请注意，这会影响产品的支持状态。此外，在 `stonith-enabled="false"` 的情况下，分布式锁管理器 (DLM) 等资源以及依赖于 DLM 的所有服务（例如 cLVM、GFS2 和 OCFS2）都将无法启动。

重要：不支持无 STONITH 的配置

不支持无 STONITH 资源的群集。

6.2.4 双节点群集的 Corosync 配置

使用引导脚本时，Corosync 配置包含一个 `quorum` 段落，其中包含以下选项：

例 6.1：双节点群集的 COROSYNC 配置摘录

```
quorum {
  # Enable and configure quorum subsystem (default: off)
  # see also corosync.conf.5 and votequorum.5
  provider: corosync_votequorum
  expected_votes: 2
  two_node: 1
}
```

与 SUSE Linux Enterprise 11 相比，SUSE Linux Enterprise 12 中的 `votequorum` 子系统基于 Corosync 版本 2.x。这表示不得使用 `no-quorum-policy=ignore` 选项。

当设置了 `two_node: 1` 时，默认会自动启用 `wait_for_all` 选项。如果未启用 `wait_for_all`，则群集应在两个节点上并行启动。否则，第一个节点将对缺失的第二个节点执行启动屏蔽。

6.2.5 N 节点群集的 Corosync 配置

如果不使用双节点群集，我们强烈建议使用奇数数目的节点来构成 N 节点群集。在仲裁配置方面，可以使用以下选项：

- 使用 `ha-cluster-join` 命令添加更多的节点，或
- 手动调整 Corosync 配置。

如果要手动调整 `/etc/corosync/corosync.conf`，请使用以下设置：

例 6.2：N 节点群集的 COROSYNC 配置摘录

```
quorum {
  provider: corosync_votequorum ❶
  expected_votes: N ❷
  wait_for_all: 1 ❸
}
```

- ❶ 使用 Corosync 的仲裁服务
- ❷ 预期投票数。可以在 `quorum` 段落中提供此参数，或者在提供了 `odelist` 段落时由系统自动计算此参数。
- ❸ 启用“等待所有节点” (WFA) 功能。如果启用了 WFA，只有在所有节点可见之后，群集才会首次具有法定票数。要避免启动时出现某些资源争用情况，将 `wait_for_all` 设置为 `1` 可能有帮助。例如，在五节点群集中，每个节点有一个投票，因此，`expected_votes` 设置为 `5`。三个或更多个节点彼此可见后，群集分区将立即具有法定票数，可以开始运行。

6.3 群集资源

作为群集管理员，您需要在群集中为服务器上运行的每个资源或应用程序创建群集资源。群集资源可以包括网站、电子邮件服务器、数据库、文件系统、虚拟机和任何其他基于服务器的应用程序或在任意时间对用户都可用的服务。

6.3.1 资源管理

必须先设置群集中的资源，然后才能使用它。例如，要使用 Apache 服务器作为群集资源，请先设置 Apache 服务器并完成 Apache 配置，然后才能在群集中启动相应的资源。

如果资源有特定环境要求，请确保这些要求已得到满足并且在所有群集节点上均相同。这种配置不由 High Availability Extension 管理。您必须自行管理。



注意：不要处理由群集管理的服务

使用 High Availability Extension 管理资源时，不得以其他方式（在群集外，例如手动或在引导时或重引导时）启动或停止同一资源。High Availability Extension 软件负责所有服务的启动或停止操作。

如果当服务已在群集控制下运行后您需要执行测试或维护任务，请确保先将资源、节点或整个群集置于维护模式，然后再进行手动处理。有关细节，请参见第 16.2 节“用于维护任务的不同选项”。

配置群集中的资源后，请使用群集管理工具手动启动、停止、清理、删除或迁移资源。有关如何使用首选群集管理工具执行此操作的细节：

- Hawk2：第 7 章“使用 Hawk2 配置和管理群集资源”
- crmsh：第 8 章“配置和管理群集资源（命令行）”



重要：资源 ID 和节点名称

群集资源和群集节点的名称应该不同。否则 Hawk2 将会失败。

6.3.2 支持的资源代理类

对于添加的每个群集资源，需要定义资源代理需遵守的标准。资源代理提取它们提供的服务并显示群集的确切状态，以使群集对其管理的资源不作确答。群集依赖于资源代理在收到启动、停止或监视命令时作出相应反应。

通常，资源代理的形式为外壳脚本。High Availability Extension 支持以下类别的资源代理：

Open Cluster Framework (OCF) 资源代理

OCF RA 代理最适合与高可用性搭配使用，特别是当您需要可升级克隆资源或特殊监视功能时。这些代理通常位于 `/usr/lib/ocf/resource.d/provider/` 中。其功能与 LSB 脚本的功能相似。但是，始终使用环境变量进行配置，这样可轻松接受和处理参数。OCF 规范有严格的定义，其中包括操作必须返回退出码，请参见第 9.3 节“OCF 返回码和故障恢复”。群集严格遵循这些规范。

要求所有 OCF 资源代理至少包含操作 `start`、`stop`、`status`、`monitor` 和 `meta-data`。`meta-data` 操作可检索有关如何配置代理的信息。例如，要了解提供程序 `heartbeat` 的 `IPaddr` 代理的更多信息，请使用以下命令：

```
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/IPaddr meta-data
```

输出是 XML 格式的信息，包括多个部分（代理的常规描述、可用参数和可用操作）。或者，也可以使用 `crmsh` 来查看有关 OCF 资源代理的信息。有关细节，请参见第 8.1.3 节“显示有关 OCF 资源代理的信息”。

Linux Standards Base (LSB) 脚本

LSB 资源代理一般由操作系统/分发包提供，并可在 `/etc/init.d` 中找到。要用于群集，它们必须遵守 LSB init 脚本规范。例如，它们必须实施了多个操作，至少包括 `start`、`stop`、`restart`、`reload`、`force-reload` 和 `status`。有关详细信息，请参见 http://refspecs.linuxbase.org/LSB_4.1.0/LSB-Core-generic/LSB-Core-generic/iniscrptact.html。

这些服务的配置没有标准化。如果要将 LSB 脚本用于 High Availability，请确保您了解如何配置相关脚本。通常，您可以在 `/usr/share/doc/packages/PACKAGENAME` 中的相关包文档中找到配置此类脚本的信息。

Systemd

从 SUSE Linux Enterprise 12 起，systemd 取代了常用的 System V init 守护程序。Pacemaker 可以管理 systemd 服务（如果有）。systemd 不使用 init 脚本，而是使用单元文件。通常，服务（或单元文件）由操作系统提供。如果您要转换现有的 init 脚本，可访问 <http://0pointer.de/blog/projects/systemd-for-admins-3.html> 找到更多信息。

服务

当前同时存在许多“通用”类型的系统服务：LSB（属于 System V init）、systemd 和（在某些分发版中提供的）upstart。因此，Pacemaker 支持使用特殊别名，这样可智能识别要将哪个服务应用到给定的群集节点。当群集中混合使用 systemd、upstart 和 LSB 服务时，此功能尤其有用。Pacemaker 将尝试按以下顺序查找指定服务：LSB (SYS-V) init 脚本、Systemd 单元文件或 Upstart 作业。

Nagios

使用监视插件（以前称为 Nagios 插件）可以监视远程主机上的服务。Pacemaker 可以使用监视插件（如果有）来执行远程监视。有关详细信息，请参见第 6.6.1 节“使用监视插件监视远程主机上的服务”。

STONITH（屏蔽）资源代理

此类仅用于与屏蔽相关的资源。有关详细信息，参见第 10 章“屏障和 STONITH”。

随 High Availability Extension 提供的代理已写入 OCF 规范。

6.3.3 资源类型

可创建以下类型的资源：

原始资源

基元资源是最基本的资源类型。

了解如何使用首选群集管理工具创建原始资源：

- Hawk2: [过程 7.5 “添加原始资源”](#)
- crmsh: [第 8.3.2 节 “创建群集资源”](#)

组

组包含一组需要放在一起、按顺序启动和按相反顺序停止的资源。有关更多信息，请参见第 6.3.5.1 节“组”。

克隆资源

克隆是可以在多个主机上处于活动状态的资源。如果各个资源代理支持，则任何资源均可克隆。有关更多信息，请参见第 6.3.5.2 节“克隆资源”。

可升级克隆（以前称为主/从资源或多状态资源）是一种可以升级的特殊类型的克隆资源。

6.3.4 资源模板

如果希望创建具有类似配置的多个资源，则定义资源模板是最简单的方式。定义后，就可以在基元资源或特定类型的约束中引用该模板，请参见第 6.5.3 节“资源模板和约束”。

如果在原始资源中引用了模板，则该原始资源将继承在模板中定义的所有操作、实例属性（参数）、元属性和利用率属性。此外，还可以为原始资源定义特定的操作或属性。如果在模板和原始资源中都定义了以上内容，则原始资源中定义的值将优先于模板中定义的值。

了解如何使用首选群集配置工具定义资源模板：

- Hawk2: 过程 7.6 “添加资源模板”
- crmsh: 第 8.3.3 节 “创建资源模板”

6.3.5 高级资源类型

原始资源是最简单的一种资源，易于配置，不过您的群集配置可能还需要更为高级的资源类型，如组、克隆资源或可升级克隆资源。

6.3.5.1 组

某些群集资源依赖于其他组件或资源，它们要求每个组件或资源都按特定顺序启动并在同一服务器上与它所依赖的资源一起运行。要简化此配置，可以使用群集资源组。

例 6.3：WEB 服务器的资源组

资源组示例可以是需要 IP 地址和文件系统的 Web 服务器。在本例中，每个组件都是组成群集资源组的一个单独资源。资源组将在一台或多台服务器上运行。在发生软件或硬件故障时，资源组会将故障转移到群集中的另一台服务器，这一点与单个群集资源类似。

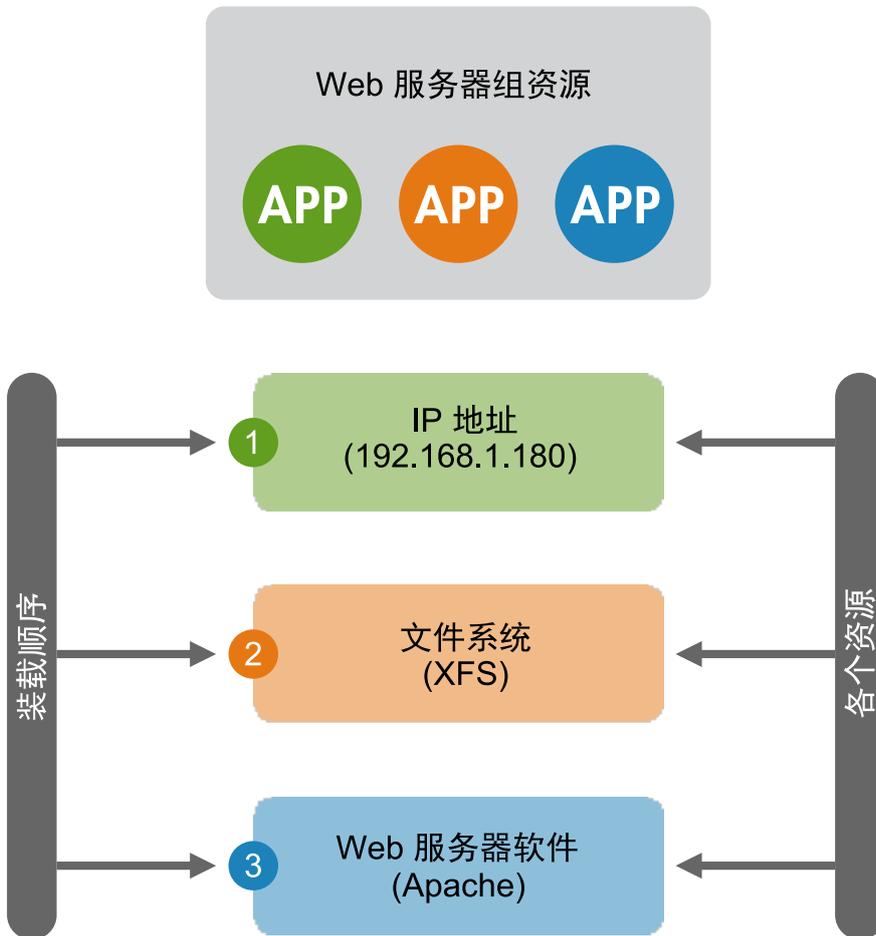


图 6.1：组资源

组具有以下属性：

启动和停止

资源按其显示顺序启动并按相反顺序停止。

相关性

如果组中某个资源在某处无法运行，则该组中位于其之后的任何资源都不允许运行。

内容

组可能仅包含一些原始群集资源。组必须包含至少一个资源，否则配置无效。要引用组资源的子级，请使用子级 ID 而不是组 ID。

限制

尽管在约束中可以引用组的子代，但通常倾向于使用组的名称。

粘性

粘性在组中可以累加。每个活动的组成员可以将其粘性值累加到组的总分中。因此，如果 `resource-stickiness` 的默认值是 `100`，且组中有 7 个成员（其中 5 个成员处于活动状态），那么整个组首选其当前位置（分数为 `500`）。

资源监视

要为组启用资源监视，必须为组中每个要监视的资源分别配置监视。

了解如何使用首选群集管理工具创建组：

- Hawk2: [过程 7.9 “添加资源组”](#)
- crmsh: [第 8.3.10 节 “配置群集资源组”](#)

6.3.5.2 克隆资源

您可能希望某些资源在群集的多个节点上同时运行。为此，必须将资源配置为克隆资源。可以配置为克隆的资源示例包括群集文件系统（如 OCFS2）。可以克隆提供的任何资源。资源的资源代理支持此操作。克隆资源的配置甚至也有不同，具体取决于资源驻留的节点。

资源克隆有三种类型：

匿名克隆

这是最简单的克隆类型。这种克隆类型在所有位置上的运行方式都相同。因此，每台计算机上只能有一个匿名克隆实例是活动的。

全局唯一克隆

这些资源各不相同。一个节点上运行的克隆实例与另一个节点上运行的实例不同，同一个节点上运行的任何两个实例也不同。

可升级克隆（多状态资源）

这些资源的活动实例分为两种状态：主动和被动。有时也称为主要和辅助，或主和从。可升级克隆可以是匿名的，也可以是全局唯一。另请参见第 6.3.5.3 节“可升级克隆（多状态资源）”。

克隆资源必须正好包含一组或一个常规资源。

配置资源监视或约束时，克隆资源与简单资源具有不同的要求。有关细节，请参见 <http://www.clusterlabs.org/pacemaker/doc/> 上的《Pacemaker Explained》（Pacemaker 配置说明）。请参见“Clones - Resources That Get Active on Multiple Hosts”（克隆资源 - 在多个主机上处于活动状态的资源）一节。

了解如何使用首选群集管理工具创建克隆资源：

- Hawk2: 过程 7.10 “添加克隆资源”
- crmsh: 第 8.3.11 节 “配置克隆资源”。

6.3.5.3 可升级克隆（多状态资源）

可升级克隆（以前称为多状态资源）是一种特殊的克隆。它们允许实例处于两种操作模式中的一种（即 `master` 或 `slave`，不过您也可以按照自己的想法来命名这些模式）。可升级克隆只能包含一个组或一个常规资源。

配置资源监视或约束时，可升级克隆的要求与简单资源不同。有关详细信息，请参见《Pacemaker Explained》（Pacemaker 说明）。Pacemaker 1.1 版本可从 <http://www.clusterlabs.org/pacemaker/doc/> 获得。请参见“Multi-state - Resources That Have Multiple Modes”（多状态 - 具有多个节点的资源）一节。

6.3.6 资源选项（元属性）

您可以为添加的每个资源定义选项。群集使用这些选项来决定资源的行为方式，它们会告知 CRM 如何对待特定的资源。可以使用 `crm_resource --meta` 命令或者按过程 7.5 “添加原始资源”中所述使用 Hawk2 来设置资源选项。

表 6.1：原始资源选项

选项	描述	默认值
<u>priority</u>	如果不允许所有的资源都处于活动状态，群集会停止优先级较低的资源以便让优先级较高的资源处于活动状态。	<u>0</u>
<u>target-role</u>	群集应在哪种状态下尝试保留此资源？允许的值有： <u>stopped</u> 、 <u>started</u> 、 <u>master</u> 。	<u>started</u>
<u>is-managed</u>	是否允许群集启动和停止资源？允许的值： <u>true</u> 、 <u>false</u> 。如果该值设置为 <u>false</u> ，则资源的状态仍受监视，并会报告任何故障。这与将资源设置为 <u>maintenance="true"</u> 的情况完全不同。	<u>true</u>
<u>maintenance</u>	是否可以手动处理资源？允许的值： <u>true</u> 、 <u>false</u> 。如果设置为 <u>true</u> ，则所有资源将变为不受管理：群集将停止监视这些资源，因此不知道这些资源的状态。您可以根据需要停止或重新启动群集资源，而无需等待群集尝试重新启动它们。	<u>false</u>
<u>resource-stickiness</u>	资源留在所处位置的自愿程度如何？	<u>calculated</u>

选项	描述	默认值
<u>migration-threshold</u>	节点上的此资源应发生多少故障后才能确定该节点没有资格主管此资源？	<u>INFINITY</u> (已禁用)
<u>multiple-active</u>	如果群集发现资源在多个节点上处于活动状态，应执行什么操作？允许的值： <u>block</u> (将资源标记为未受管)、 <u>stop_only</u> 和 <u>stop_start</u> 。	<u>stop_start</u>
<u>failure-timeout</u>	在恢复为如同未发生故障一样正常工作（并允许资源返回它发生故障的节点）之前，需要等待几秒钟？	<u>0</u> (已禁用)
<u>allow-migrate</u>	允许对支持 <u>migrate_to</u> / <u>migrate_from</u> 操作的资源进行资源迁移。	<u>false</u>
<u>remote-node</u>	此资源定义的远程节点的名称。这会将资源作为远程节点启用，同时定义唯一的名称用于标识该远程节点。如果未设置其他参数，则此值还会被假定为要在 <u>remote-port</u> 端口上连接到的主机名。	<u>none</u> (已禁用)

选项	描述	默认值
	 警告：使用唯一 ID 此值不得与任何现有资源 ID 或节点 ID 重复。	
<code>remote-port</code>	guest 与 pacemaker_remote 建立连接时使用的自定义端口。	<code>3121</code>
<code>remote-addr</code>	当远程节点的名称不是 guest 的主机名时要连接到的 IP 地址或主机名。	<code>remote-node</code> （用作主机名的值）
<code>remote-connect-timeout</code>	待发 guest 连接在超时之前的等待时间。	<code>60s</code>

6.3.7 实例属性（参数）

可为所有资源类的脚本指定参数，这些参数可确定脚本的行为方式和所控制的服务实例。如果资源代理支持参数，则可使用 `crm_resource` 命令或按过程 7.5 “添加原始资源” 中所述使用 Hawk2 来添加这些参数。在 `crm` 命令行实用程序和 Hawk2 中，实例属性分别称为 `params` 和 `Parameter`。通过以 `root` 身份执行以下命令，可找到 OCF 脚本支持的实例属性列表：

```
root # crm ra info [class:[provider:]]resource_agent
```

或（无可选部分）：

```
root # crm ra info resource_agent
```

输出列出了所有支持的属性及其用途和默认值。

例如，命令

```
root # crm ra info IPAddr
```

返回以下输出:

```
Manages virtual IPv4 addresses (portable version) (ocf:heartbeat:IPAddr)
```

```
This script manages IP alias IP addresses
```

```
It can add an IP alias, or remove one.
```

```
Parameters (* denotes required, [] the default):
```

```
ip* (string): IPv4 address
```

```
The IPv4 address to be configured in dotted quad notation, for example  
"192.168.1.1".
```

```
nic (string, [eth0]): Network interface
```

```
The base network interface on which the IP address will be brought  
online.
```

```
If left empty, the script will try and determine this from the  
routing table.
```

```
Do NOT specify an alias interface in the form eth0:1 or anything here;  
rather, specify the base interface only.
```

```
cidr_netmask (string): Netmask
```

```
The netmask for the interface in CIDR format. (ie, 24), or in  
dotted quad notation 255.255.255.0).
```

```
If unspecified, the script will also try to determine this from the  
routing table.
```

```
broadcast (string): Broadcast address
```

```
Broadcast address associated with the IP. If left empty, the script will  
determine this from the netmask.
```

```
iflabel (string): Interface label
```

```
You can specify an additional label for your IP address here.
```

```
lvs_support (boolean, [false]): Enable support for LVS DR
```

Enable support for LVS Direct Routing configurations. In case a IP address is stopped, only move it to the loopback device to allow the local node to continue to service requests, but no longer advertise it on the network.

`local_stop_script` (string):
Script called when the IP is released

`local_start_script` (string):
Script called when the IP is added

`ARP_INTERVAL_MS` (integer, [500]): milliseconds between gratuitous ARPs
milliseconds between ARPs

`ARP_REPEAT` (integer, [10]): repeat count
How many gratuitous ARPs to send out when bringing up a new address

`ARP_BACKGROUND` (boolean, [yes]): run in background
run in background (no longer any reason to do this)

`ARP_NETMASK` (string, [ffffffffffff]): netmask for ARP
netmask for ARP - in nonstandard hexadecimal format.

Operations' defaults (advisory minimum):

<code>start</code>	<code>timeout=90</code>
<code>stop</code>	<code>timeout=100</code>
<code>monitor_0</code>	<code>interval=5s timeout=20s</code>



注意：组、克隆或可升级克隆的实例属性

请注意，组、克隆和可升级克隆资源没有实例属性。但是，组、克隆或可升级克隆的子级会继承任何实例属性集。

6.3.8 资源操作

默认情况下，群集将不会确保您的资源一直正常。要指示群集执行此操作，需要将监视操作添加到资源定义中。可为所有类或资源代理添加监视操作。有关更多信息，请参见第 6.4 节“资源监视”。

表 6.2：资源操作属性

操作	描述
<u>ID</u>	您的操作名称。必须是唯一的。（不会显示 ID）。
<u>name</u>	要执行的操作。常见值： <u>monitor</u> 、 <u>start</u> 和 <u>stop</u> 。
<u>interval</u>	执行操作的频率。单位：秒
<u>timeout</u>	需要等待多久才能声明操作失败。
<u>requires</u>	需要满足哪些条件才会发生此操作。允许的值： <u>nothing</u> 、 <u>quorum</u> 和 <u>fencing</u> 。默认值取决于是否启用屏蔽和资源的类是否为 <u>stonith</u> 。对于 STONITH 资源，默认值为 <u>nothing</u> 。
<u>on-fail</u>	此操作失败时执行的操作。允许的值： <ul style="list-style-type: none">• <u>ignore</u>：假装资源没有失败。• <u>block</u>：不对资源执行任何进一步操作。• <u>stop</u>：停止资源并且不在其他位置启动该资源。• <u>restart</u>：停止资源并（可能在不同的节点上）重新启动。

操作	描述
	<ul style="list-style-type: none"> • <u>fence</u>：关闭资源故障的节点 (STONITH)。 • <u>standby</u>：将所有资源从资源失败的节点上移走。
<u>enabled</u>	如果值为 <u>false</u> ，将操作视为不存在。允许的值： <u>true</u> 、 <u>false</u> 。
<u>role</u>	仅当资源具有此角色时才运行操作。
<u>record-pending</u>	可全局设置或为单独资源设置。使 CIB 反映资源上“正在进行中的”操作的状态。
<u>description</u>	操作描述。

6.3.9 超时值

资源的超时值会受以下参数的影响：

- op_defaults（操作的全局超时），
- 在资源模板中定义的特定超时值，
- 为资源定义的特定超时值。



注意：值的优先级

如果为资源定义了特定值，则该值优先于全局默认值。资源的特定值也优先于在资源模板中定义的值。

获取超时值权限非常重要。将它们设置得太小会导致大量（不必要的）屏蔽操作，原因如下：

1. 如果资源运行超时，它将失败，群集将尝试停止它。
2. 如果停止资源也失败（例如，由于停止超时值设置得太小），群集将屏蔽该节点。发生此状况时所在的节点将被视为已失去控制。

您可以使用 `crmsh` 和 `Hawk2` 调整操作的全局默认值并设置任何特定的超时值。确定和设置超时值的最佳实践如下所示：

过程 6.1：确定超时值

1. 检查资源启动和停止（在负载状况下）所需的时间。
2. 如果需要，请添加 `op_defaults` 参数并相应地设置（默认）超时值：
 - a. 例如，将 `default-action-timeout` 设置为 `60` 秒：

```
crm(live)configure# op_defaults timeout=60
```

- b. 对于需要更长时间期限的资源，则定义单独的超时值。
3. 为资源配置操作时，添加单独的 `start` 和 `stop` 操作。使用 `Hawk2` 配置操作时，它会针对这些操作提供有用的超时建议。

6.4 资源监视

如果要确保资源正在运行，必须为其配置资源监视。

如果资源监视程序检测到故障，将发生以下情况：

- 根据 `/etc/corosync/corosync.conf` 中 `logging` 部分指定的配置生成日志文件消息。
- 故障会在群集管理工具（`Hawk2`、`crm status`）中和 CIB 状态部分反映出来。
- 群集将启动重要的恢复操作，可包括停止资源以修复故障状态以及在本地或在其他节点上重新启动资源。资源也可能不会重新启动，具体取决于配置和群集状态。

如果不配置资源监视，则不会告知成功启动的资源故障，且群集始终显示资源状况正常。

监视已停止的资源

通常，只要资源在运行，就仅受群集监视。但是，为了检测并发违例，还需为停止的资源配置监视。例如：

```
primitive dummy1 ocf:heartbeat:Dummy \  
    op monitor interval="300s" role="Stopped" timeout="10s" \  
    op monitor interval="30s" timeout="10s"
```

当资源 `dummy1` 处于 `role="Stopped"` 状态时，此配置就会每 `300` 秒触发一次对该资源的监视操作。在运行时，针对它的监视间隔为 `30` 秒。

检测

CRM 会对每个节点上的各个资源执行初始监视，也称为探测。清理资源之后也会执行探测。如果为资源定义了多项监视操作，则 CRM 将选择间隔时间最小的一项操作，并会使用其超时值作为探测的默认超时值。如果未配置任何监视操作，则将应用整个群集的默认值。默认值为 `20` 秒（如果未通过配置 `op_defaults` 参数来指定其他默认值）。如果您不想依赖自动计算或 `op_defaults` 值，请为此资源的探测定义具体的监视操作。为此，可以添加一个监视操作并将 `interval` 设置为 `0`，例如：

```
crm(live)configure# primitive rsc1 ocf:pacemaker:Dummy \  
    op monitor interval="0" timeout="60"
```

`rsc1` 的探测将在 `60` 秒后超时，而不管 `op_defaults` 中定义的全局超时或者配置的任何其他操作超时如何。如果未设置 `interval="0"` 以指定相应资源的探测，CRM 将自动检查是否为该资源定义了任何其他监视操作，并按如上所述计算探测的超时值。

了解如何使用首选群集管理工具添加对资源的监视操作：

- Hawk2: [过程 7.13 “添加和修改操作”](#)
- crmsh: [第 8.3.9 节 “配置资源监视”](#)

6.5 资源约束

配置好所有资源只是完成了该任务的一部分。即便群集熟悉所有必需资源，它可能还无法进行正确处理。资源约束允许您指定在哪些群集节点上运行资源、以何种顺序装载资源，以及特定资源依赖于哪些其他资源。

6.5.1 约束类型

提供三种不同的约束：

资源位置

位置约束定义资源可以、不可以或首选在哪些节点上运行。

资源共置

共置约束告知群集哪些资源可以或不可以在一个节点上一起运行。

资源顺序

顺序约束定义操作的顺序。

! 重要：约束与特定资源类型的限制

- 不要为资源组的成员创建共置约束，而是应该创建指向整个资源组的共置约束。其他所有类型的约束可安全地用于资源组的成员。
- 不要对包含克隆资源或者应用了可升级克隆资源的资源使用任何约束。约束必须应用于克隆资源或可升级克隆资源，而不能应用于子资源。

6.5.1.1 资源集

6.5.1.1.1 使用资源集定义约束

可以使用 资源集 作为定义位置、共置或顺序约束的备用格式，在资源集中，基元资源已被全部分组到一个集合中。以前，为了实现此目的，用户可以定义一个资源组（不一定总能准确表达设计意图），也可以将每种关系定义为单个约束。随着资源和组合数目的增加，后面这种做法会导致约束过度膨胀。通过资源集进行配置并不一定会减少复杂程度，但更易于理解和维护，如以下示例中所示。

例 6.4：用于位置约束的资源集

例如，可以在 `crmsh` 中使用资源集 (`loc-alice`) 的以下配置，将两个虚拟 IP (`vip1` 和 `vip2`) 置于同一个节点 `alice`：

```

crm(live)configure# primitive vip1 ocf:heartbeat:IPaddr2 params
ip=192.168.1.5
crm(live)configure# primitive vip2 ocf:heartbeat:IPaddr2 params
ip=192.168.1.6
crm(live)configure# location loc-alice { vip1 vip2 } Mandatory: alice

```

如果想要使用资源集来替换共置约束的配置，请考虑以下两个示例：

例 6.5：共置资源链

```

<constraints>
  <rsc_colocation id="coloc-1" rsc="B" with-rsc="A" score="INFINITY"/>
  <rsc_colocation id="coloc-2" rsc="C" with-rsc="B" score="INFINITY"/>
  <rsc_colocation id="coloc-3" rsc="D" with-rsc="C" score="INFINITY"/>
</constraints>

```

由资源集表示的相同配置：

```

<constraints>
  <rsc_colocation id="coloc-1" score="INFINITY" >
    <resource_set id="colocated-set-example" sequential="true">
      <resource_ref id="A"/>
      <resource_ref id="B"/>
      <resource_ref id="C"/>
      <resource_ref id="D"/>
    </resource_set>
  </rsc_colocation>
</constraints>

```

如果想使用资源集来替换顺序约束的配置，请考虑以下两个示例：

例 6.6：有序资源链

```

<constraints>
  <rsc_order id="order-1" first="A" then="B" />
  <rsc_order id="order-2" first="B" then="C" />
  <rsc_order id="order-3" first="C" then="D" />
</constraints>

```

可以使用包含有序资源的资源集来实现相同的目的：

例 6.7：以资源集表示的有序资源链

```
<constraints>
  <rsc_order id="order-1">
    <resource_set id="ordered-set-example" sequential="true">
      <resource_ref id="A"/>
      <resource_ref id="B"/>
      <resource_ref id="C"/>
      <resource_ref id="D"/>
    </resource_set>
  </rsc_order>
</constraints>
```

资源集可以是有序的 (`sequential=true`)，也可以是无序的 (`sequential=false`)。此外，可以使用 `require-all` 属性在 `AND` 与 `OR` 逻辑之间切换。

6.5.1.1.2 不带依赖项的共置约束的资源集

有时，将一组资源放置在同一个节点上（定义共置约束）会很有用，但前提是这些资源之间不存在硬依赖性。例如，您想要在同一节点上放置两个资源，但不希望群集在其中一个资源发生故障时重新启动另一个资源。可以在 `crm` 外壳中使用 `weak bond` 命令实现此目的。

了解如何使用首选群集管理工具设置这些“弱绑定”：

- `crmsh`：第 8.3.5.3 节 “共置没有依赖性的资源集”

6.5.1.2 更多信息

了解如何使用首选群集管理工具添加各种约束：

- `Hawk2`：第 7.6 节 “配置约束”
- `crmsh`：第 8.3.5 节 “配置资源约束”

有关配置约束的更多信息以及顺序和共置基本概念的详细背景信息，请参见以下文档。可以从 <http://www.clusterlabs.org/pacemaker/doc/> 访问这些文档：

- 《Pacemaker Explained》（Pacemaker 配置说明），“Resource Constraints”（资源约束）一章
- 《Colocation Explained》（共置说明）
- 《Ordering Explained》（顺序说明）

6.5.2 分数和无限值

定义约束时，还需要指定分数。各种分数是群集工作方式的重要组成部分。其实，从迁移资源到决定在已降级群集中停止哪些资源的整个过程是通过以某种方式操纵分数来实现的。分数按每个资源来计算，资源分数为负的任何节点都无法运行该资源。计算资源的分数后，群集会选分数最高的节点。

INFINITY（无穷大）目前定义为 1,000,000。提高或降低分数需遵循以下三个基本规则：

- 任何值 + 无穷大 = 无穷大
- 任何值 - 无穷大 = -无穷大
- 无穷大 - 无穷大 = -无穷大

定义资源约束时，需为每个约束指定一个分数。分数表示您指派给此资源约束的值。分数较高的约束先应用，分数较低的约束后应用。通过使用不同的分数为既定资源创建更多位置约束，可以指定资源要故障转移至的目标节点的顺序。

6.5.3 资源模板和约束

如果定义了资源模板（请参见第 6.3.4 节“资源模板”），则可在以下类型的约束中引用该模板：

- 顺序约束
- 共置约束
- rsc_ticket 约束（用于 Geo 群集）。

但是，共置约束不得包含多个对模板的引用。资源集不得包含对模板的引用。

在约束中引用的资源模板代表派生自该模板的所有原始资源。这意味着，约束将应用于引用资源模板的所有原始资源。在约束中引用资源模板是资源集的备用方式，它可以显著简化群集配置。有关资源集的细节，请参见过程 7.17 “为约束使用资源集”。

6.5.4 故障转移节点

资源在出现故障时会自动重新启动。如果在当前节点上无法实现此操作，或者此操作在当前节点上失败了 N 次，它将尝试故障转移到其他节点。每次资源失败时，其失败计数都会增加。您可以定义资源的故障次数（migration-threshold），在该值之后资源会迁移到新节点。如果群集中存在两个以上的节点，则特定资源故障转移的节点由 High Availability 软件选择。

但可以通过为资源配置一个或多个位置约束和一个 migration-threshold 来指定此资源将故障转移到的节点。

了解如何使用首选群集管理工具指定故障转移节点：

- Hawk2: 第 7.6.6 节 “指定资源故障转移节点”
- crmsh: 第 8.3.6 节 “指定资源故障转移节点”

例 6.8：迁移阈值 - 流程

例如，假设您已经为 rsc1 资源配制了一个首选在 alice 节点上运行的位置约束。如果那里失败了，系统会检查 migration-threshold 并与故障计数进行比较。如果故障计数 \geq migration-threshold，会将资源迁移到下一个自选节点。

一旦达到阈值，节点将不再能运行失败资源，直到重置资源的 failcount 为止。这可以由群集管理员手动执行或通过设置资源的 failure-timeout 选项执行。

例如，设置 migration-threshold=2 和 failure-timeout=60s 会导致资源在发生两次故障后迁移到新节点。允许该资源在一分钟后移回（具体取决于粘性和约束分数）。

迁移阈值概念有两个异常，发生在资源启动失败或停止失败时：

- 启动失败将失败计数设置为 INFINITY，因此总是会导致立即迁移。
- 停止故障会导致屏蔽（stonith-enabled 设置为 true 时，这是默认设置）。

如果不定义 STONITH 资源（或 `stonith-enabled` 设置为 `false`），则该资源不会迁移。

有关配合首选群集管理工具使用迁移阈值和重置故障计数的详细信息，请参见：

- Hawk2：第 7.6.6 节 “指定资源故障转移节点”
- crmsh：第 8.3.6 节 “指定资源故障转移节点”

6.5.5 故障回复节点

当原始节点恢复联机并位于群集中时，资源可能会故障回复到该节点。为防止资源故障回复到之前运行它的节点，或者要指定让该资源故障回复到其他节点，请更改其资源粘性值。可以在创建资源时或之后指定资源粘性。

指定资源粘性值时请考虑以下含义：

值为 `0`：

这是默认选项。资源放置在系统中的最适合位置。这意味着当负载能力“较好”或较差的节点变得可用时才转移资源。此选项的作用几乎等同于自动故障回复，只是资源可能会转移到非之前活动的节点上。

值大于 `0`：

资源更愿意留在当前位置，但是如果有更合适的节点可用时会移动。值越高表示资源越愿意留在当前位置。

值小于 `0`：

资源更愿意移离当前位置。绝对值越高表示资源越愿意离开当前位置。

值为 `INFINITY`：

如果不是因节点不适合运行资源（节点关机、节点待机、达到 `migration-threshold` 或配置更改）而强制资源转移，资源总是留在当前位置。此选项的作用几乎等同于完全禁用自动故障回复。

值为 `-INFINITY`：

资源总是移离当前位置。

6.5.6 根据资源负载影响放置资源

并非所有资源都相等。某些资源（如 Xen guest）需要托管它们的节点满足其容量要求。如果所放置资源的总需求超过了提供的容量，则资源性能将降低（或甚至失败）。

要考虑此情况，可使用 High Availability Extension 指定以下参数：

1. 特定节点提供的容量。
2. 特定资源需要的容量。
3. 资源放置整体策略。

了解如何使用首选群集管理工具配置这些设置：

- Hawk2: 第 7.6.8 节 “根据负载影响配置资源放置”
- crmsh: 第 8.3.8 节 “根据负载影响配置资源放置”

如果节点有充足的可用容量来满足资源要求，则此节点将被视为此资源的有效节点。容量的性质对 High Availability Extension 而言完全无关紧要；它只是确保在将资源移到节点上之前满足资源的所有容量要求。

要手动配置资源要求和节点提供的容量，请使用利用率属性。可根据个人喜好命名利用率属性，并根据配置需要定义多个名称/值对。但是，属性值必须是整数。

如果将具有利用率属性的多个资源组合或设置共置约束，则 High Availability Extension 会考虑此情况。如有可能，资源将被放置到可以满足所有容量要求的节点上。



注意：组的利用率属性

无法直接为资源组设置利用率属性。但是，为了简化组的配置，可以使用组中所有资源所需的总容量添加利用率属性。

High Availability Extension 还提供了方法来自动检测和配置节点容量和资源要求：

NodeUtilization 资源代理检查节点的容量（与 CPU 和 RAM 有关）。要配置自动检测，请创建类、提供程序和类型如下的克隆资源：ocf:pacemaker:NodeUtilization。每个节点上应都有一个克隆实例在运行。实例启动后，利用率部分将添加到节点的 CIB 配置中。

为了自动检测资源的最低要求（与 RAM 和 CPU 有关），Xen 资源代理已得到改善。启动 Xen 资源后，它将反映 RAM 和 CPU 的使用情况。利用率属性将自动添加到资源配置中。



注意：适用于 Xen 和 libvirt 的不同资源代理

`ocf:heartbeat:Xen` 资源代理不应与 `libvirt` 搭配使用，因为 `libvirt` 需要能够修改计算机说明文件。

对于 `libvirt`，请使用 `ocf:heartbeat:VirtualDomain` 资源代理。

除了检测最低要求外，High Availability Extension 还允许通过 `VirtualDomain` 资源代理监视当前的利用率。它检测虚拟机的 CPU 和 RAM 使用情况。要使用此功能，请配置类、提供程序和类型如下的资源：`ocf:heartbeat:VirtualDomain`。可以使用以下实例属性：`autoset_utilization_cpu` 和 `autoset_utilization_hv_memory`。两者都默认为 `true`。这将在每个监视周期中更新 CIB 中的利用率值。

与手动或自动配置容量和要求无关，放置策略必须使用 `placement-strategy` 属性（在全局群集选项中）指定。可用值如下：

`default` (默认值)

不考虑利用率值。根据位置得分分配资源。如果分数相等，资源将均匀分布在节点中。

`utilization`

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。但仍会根据分配给节点的资源数执行负载平衡。

`minimal`

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。尝试将资源集中到尽可能少的节点上（以节省其余节点上的能耗）。

`balanced`

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。尝试均匀分布资源，从而优化资源性能。



注意：配置资源优先级

可用的放置策略是最佳方法 - 它们不使用复杂的启发式解析程序即可始终实现最佳分配结果。确保正确设置资源优先级，以便首选调度最重要的资源。

例 6.9：负载均衡放置配置示例

以下示例演示了配有四台虚拟机、节点数相等的三节点群集。

```
node alice utilization memory="4000"
node bob utilization memory="4000"
node charlie utilization memory="4000"
primitive xenA ocf:heartbeat:Xen utilization hv_memory="3500" \
  params xmfile="/etc/xen/shared-vm/vm1"
  meta priority="10"
primitive xenB ocf:heartbeat:Xen utilization hv_memory="2000" \
  params xmfile="/etc/xen/shared-vm/vm2"
  meta priority="1"
primitive xenC ocf:heartbeat:Xen utilization hv_memory="2000" \
  params xmfile="/etc/xen/shared-vm/vm3"
  meta priority="1"
primitive xenD ocf:heartbeat:Xen utilization hv_memory="1000" \
  params xmfile="/etc/xen/shared-vm/vm4"
  meta priority="5"
property placement-strategy="minimal"
```

如果三个节点都处于正常状态，那么资源 xenA 将首先放置到一个节点上，然后是 xenD。xenB 和 xenC 将分配在一起或者其中一个与 xenD 分配在一起。

如果一个节点出现故障，可用的总内存将不足以托管所有资源。将确保分配 xenA，xenD 同样如此。但是，只能再放置剩余资源 xenB 和 xenC 中的一个。由于它们的优先级相同，结果未定。要解决这种不确定性，需要为其中一个资源设置更高的优先级。

6.5.7 使用标记分组资源

标记是 Pacemaker 中最近新增的功能。使用标记可以一次性引用多个资源，而无需在这些资源之间创建任何共置或顺序关系。此功能十分适用于对概念上相关的资源进行分组。例如，如果有多个资源与某个数据库相关，您可以创建一个名为 databases 的标记，并将与该数据库相关的所有资源都添加到此标记。这样，只需使用一条命令就能停止或启动所有这些资源。

标记也可以用于约束。例如，loc-db-prefer 位置约束将应用到标记了 databases 的一组资源：

```
location loc-db-prefer databases 100: alice
```

了解如何使用首选群集管理工具创建标记：

- Hawk2: [过程 7.12 “添加标记”](#)
- crmsh: [第 8.4.7 节 “分组/标记资源”](#)

6.6 管理远程主机上的服务

在最近几年中，是否能够监视和管理远程主机上的服务已变得越来越重要。SUSE Linux Enterprise High Availability Extension 11 SP3 可让用户通过监视插件来密切监视远程主机上的服务。最近添加的 `pacemaker_remote` 服务现在允许 SUSE Linux Enterprise High Availability Extension 15 SP2 全面管理和监视远程主机上的资源，就如同这些资源是真实的群集节点一样，并且无需用户在远程计算机上安装群集堆栈。

6.6.1 使用监视插件监视远程主机上的服务

虚拟机的监视可以通过 VM 代理来完成（只有在超级管理程序中出现 `guest` 时才可选择 VM 代理），或者通过从 `VirtualDomain` 或 `Xen` 代理调用外部脚本来完成。直到现在为止，仍只有通过虚拟机中对高可用性堆栈进行完全设置才能实现更细化的监视。

通过提供对监视插件（以前称为 Nagios 插件）的支持，High Availability Extension 现在还可让您监视远程主机上的服务。您可以收集 `guest` 上的外部状态，而无需修改 `guest` 映像。例如，VM `guest` 可能会运行需要能够访问的 Web 服务或简单的网络资源。现在，有了 Nagios 资源代理，您就可以监视 `guest` 上的 Web 服务或网络资源。如果这些服务再也无法访问，High Availability Extension 将触发相应 `guest` 的重启或迁移操作。

如果您的 `guest` 依赖于某项服务（例如，`guest` 要使用 NFS 服务器），则这项服务可以是由群集管理的普通资源，也可以是使用 Nagios 资源进行监视的外部服务。

要配置 Nagios 资源，必须在主机上安装以下包：

- `monitoring-plugins`
- `monitoring-plugins-metadata`

必要时，YaST 或 Zypper 将解决对后续包的任何依赖项问题。

将监视插件配置为属于资源容器（通常是 VM）的资源便是其中一个典型用例。如果容器中有任何资源发生故障，则将重启该容器。有关配置示例，请参见例 6.10 “为监视插件配置资源”。或者，若要使用 Nagios 资源代理通过网络监视主机或服务，还可将这些代理配置为普通资源。

例 6.10：为监视插件配置资源

```
primitive vm1 ocf:heartbeat:VirtualDomain \  
    params hypervisor="qemu:///system" config="/etc/libvirt/qemu/vm1.xml" \  
    op start interval="0" timeout="90" \  
    op stop interval="0" timeout="90" \  
    op monitor interval="10" timeout="30" \  
primitive vm1-sshd nagios:check_tcp \  
    params hostname="vm1" port="22" \  
    op start interval="0" timeout="120" \  
    op monitor interval="10" \  
group g-vm1-and-services vm1 vm1-sshd \  
    meta container="vm1"
```

- ① 支持的参数与监视插件的长选项相同。监视插件通过参数 `hostname` 与服务连接。因此属性的值必须是可解析的主机名或 IP 地址。
- ② 因为启动 guest 操作系统并让其服务运行需要一段时间，所以必须设置足够长的监视资源启动超时。
- ③ `ocf:heartbeat:Xen`、`ocf:heartbeat:VirtualDomain` 或 `ocf:heartbeat:lxc` 类型的群集资源容器。可以是 VM 或 Linux 容器。

以上示例仅包含适用于 `check_tcp` 插件的一个资源，但也可以针对不同的插件类型配置多个资源（例如 `check_http` 或 `check_udp`）。

如果服务的主机名相同，还可以为组指定 `hostname` 参数，而无需为各个基元资源一一添加该参数。例如：

```
group g-vm1-and-services vm1 vm1-sshd vm1-httpd \  
    meta container="vm1" \  
    params hostname="vm1"
```

如果监视插件监视的任何服务在 VM 中发生故障，则群集会检测到该情况并重启容器资源 (VM)。可以通过指定服务监视操作的 `on-fail` 属性来配置在这种情况下要执行的操作。默认值为 `restart-container`。

在考虑 VM 的 migration-threshold 时，会将服务的故障计数考虑在内。

6.6.2 使用 `pacemaker_remote` 管理远程节点上的服务

使用 `pacemaker_remote` 服务可将高可用性群集扩展到虚拟节点或远程裸机计算机。这些虚拟节点或远程裸机无需运行群集堆栈就能成为群集的成员。

High Availability Extension 现在可以启动虚拟环境（KVM 和 LXC）以及驻留在这些虚拟环境中的资源，而无需虚拟环境运行 Pacemaker 或 Corosync。

对于同时要管理用作群集资源的虚拟机以及 VM 中驻留的资源的用例，您现在可以使用以下设置：

- “常规”（裸机）群集节点运行 High Availability Extension。
- 虚拟机运行 `pacemaker_remote` 服务（几乎不需要在 VM 端进行任何配置）。
- “常规”群集节点上的群集堆栈会启动 VM 并连接到 VM 上运行的 `pacemaker_remote` 服务，以将 VM 作为远程节点集成到群集中。

由于远程节点上未安装群集堆栈，因此这意味着：

- 远程节点不参与仲裁。
- 远程节点无法成为 DC。
- 远程节点不受可伸缩性限制（Corosync 将成员数限制为 32 个节点）的约束。

《Pacemaker Remote 快速入门》文章中介绍了有关 `remote_pacemaker` 服务的更多信息，包括多个用例和详细的设置说明。

6.7 监视系统运行状况

为避免节点耗尽磁盘空间而使得系统无法管理已分配给该节点的任何资源，High Availability Extension 提供了一个资源代理 `ocf:pacemaker:SysInfo`。使用此代理可监视节点在磁盘分区的状况。SysInfo RA 会创建一个名为 `#health_disk` 的节点属性，如果任何受监视磁盘的可用空间低于指定限制，就会将其设置为 `red`。

要定义 CRM 在节点状况到达临界状态时应如何反应，请使用全局群集选项 `node-health-strategy`。

过程 6.2：配置系统运行状况监视

要在某个节点耗尽磁盘空间时从该节点自动移除资源，请执行以下操作：

1. 配置 `ocf:pacemaker:SysInfo` 资源：

```
primitive sysinfo ocf:pacemaker:SysInfo \  
  params disks="/tmp /var" ① min_disk_free="100M" ② disk_unit="M" ③ \  
  op monitor interval="15s"
```

- ① 要监视的磁盘分区。例如，`/tmp`、`/usr`、`/var` 和 `/dev`。要指定多个分区作为属性值，请以空格进行分隔。



注意：始终监视 / 文件系统

您无需在 `disks` 中指定根分区 (`/`)。此分区默认将始终受到监视。

- ② 这些分区所需的最小可用磁盘空间。您也可以指定度量单位（在上例中，`M` 表示兆字节）。如果未指定，`min_disk_free` 默认将使用 `disk_unit` 参数中定义的单位。
- ③ 报告磁盘空间所使用的单位。

2. 要完成资源配置，请创建 `ocf:pacemaker:SysInfo` 的克隆并在每个群集节点上启动此克隆。

3. 将 `node-health-strategy` 设置为 `migrate-on-red`：

```
property node-health-strategy="migrate-on-red"
```

如果 `#health_disk` 属性设置为 `red`，`pacemaker-schedulerd` 会将 `-INF` 添加到该节点的资源分数中。此时将从此节点移除所有资源。STONITH 资源将是最后一个被停止的资源，但即使 STONITH 资源不再运行，仍可屏蔽该节点。屏蔽对 CIB 有直接访问权且将继续起作用。

当节点状况变成 `red` 状态后，解决会导致问题的状况。然后清除 `red` 状态，使节点能够再次运行资源。登录到群集节点并使用下列其中一种方法：

- 执行以下命令：

```
root # crm node status-attr NODE delete #health_disk
```

- 在该节点上重新启动 Pacemaker。
- 重引导该节点。

该节点将恢复服务并可再次运行资源。

6.8 更多信息

<http://crmsh.github.io/> 

用于高可用性群集管理的高级命令行界面 crm 外壳 (crmsh) 的主页。

<http://crmsh.github.io/documentation> 

提供有关 crm 外壳的多份文档，包括使用 crmsh 完成基本群集设置的入门教程，以及 crm 外壳的综合性手册。后者可在 <http://crmsh.github.io/man-2.0/>  上访问。<http://crmsh.github.io/start-guide/>  上提供了相关教程。

<http://clusterlabs.org/> 

Pacemaker 主页，随 High Availability Extension 提供的群集资源管理器。

<http://www.clusterlabs.org/pacemaker/doc/> 

提供数个综合性手册，以及一些解释一般概念的简短文档。例如：

- 《Pacemaker Explained》（Pacemaker 说明）：包含全面、详尽的参考信息。
- 《Colocation Explained》（共置说明）
- 《Ordering Explained》（顺序说明）

7 使用 Hawk2 配置和管理群集资源

要配置和管理群集资源，请使用 Hawk2 或 crm 外壳 (crmsh) 命令行实用程序。如果您从安装了 Hawk 的较低版 SUSE® Linux Enterprise High Availability Extension 升级，该包将替换为最新版本 Hawk2。

Hawk2 的用户友好 Web 界面可让您从 Linux 或非 Linux 计算机监视和管理高可用性群集。可使用（图形）Web 浏览器从群集内外的任何计算机访问 Hawk2。

7.1 Hawk2 要求

仅当系统满足以下要求后，用户才能登录 Hawk2：

hawk2 包

此 hawk2 包必须安装在要使用 Hawk2 连接的所有群集节点上。

Web 浏览器

在要使用 Hawk2 访问群集节点的计算机上，需要安装启用了 JavaScript 和 Cookie 的（图形）Web 浏览器才能建立连接。

Hawk2 服务

要使用 Hawk2，必须在要通过 Web 界面连接到的节点上启动相应的 Web 服务。请参见过程 7.1 “启动 Hawk2 服务”。

如果您是使用 ha-cluster-bootstrap 包中的脚本设置群集的，则此时 Hawk2 服务已启用。

每个群集节点上的用户名、组和口令

Hawk2 用户必须是 haclient 组的成员。安装程序将创建名为 hacluster 的 Linux 用户，该用户将添加到 haclient 组中。

使用 **ha-cluster-init** 脚本进行设置时，将为 hacluster 用户设置默认口令。在启动 Hawk2 之前，请将它更改为安全口令。如果您不使用 **ha-cluster-init** 脚本，请先为 hacluster 设置口令，或者创建属于 haclient 组成员的新用户。请在要使用 Hawk2 连接的所有节点上执行此操作。

过程 7.1：启动 HAWK2 服务

1. 在要连接到的节点上，打开外壳并以 `root` 用户身份登录。
2. 通过输入以下命令，检查服务的状态

```
root # systemctl status hawk
```

3. 如果服务未在运行，请使用以下命令启动服务

```
root # systemctl start hawk
```

如果希望 Hawk2 在引导时自动启动，请执行以下命令：

```
root # systemctl enable hawk
```

7.2 登录

Hawk2 Web 界面使用 HTTPS 协议和端口 `7630`。

您无需使用 Hawk2 登录个别群集节点，而是可以将一个浮动的虚拟 IP 地址（`IPAddr` 或 `IPAddr2`）配置为群集资源。该地址无需任何特殊配置。如此，无论 Hawk 服务在哪个物理节点上运行，客户端都可以连接到该服务。

在使用 `ha-cluster-bootstrap` 脚本设置群集时，系统会询问您是否配置虚拟 IP 以用于群集管理。

过程 7.2：登录 HAWK2 WEB 界面

1. 在任一计算机上，启动 Web 浏览器并输入以下 URL：

```
https://HAWKSERVER:7630/
```

使用运行 Hawk Web 服务的任何群集节点的 IP 地址或主机名替换 `HAWKSERVER`。

如果已配置虚拟 IP 地址以使用 Hawk2 进行群集管理，请使用该虚拟 IP 地址替换 `HAWKSERVER`。



注意：证书警告

当您首次尝试访问 URL 时如果显示证书警告，则表示使用了自我签名证书。默认情况下，自我签名证书不被视为可信证书。

要校验证书，请联系群集操作员获取证书细节。

要继续，可在浏览器中添加例外，以绕过警告。

有关如何将自我签名证书替换为官方证书颁发机构签名的证书的信息，请参见[替换自签名证书](#)。

2. 在 Hawk2 登录屏幕上，输入 hacluster 用户（或属于 haclient 组的任何其他用户）的 用户名 和 口令。
3. 单击登录。

7.3 Hawk2 概述：主要元素

登录 Hawk2 后，左侧会显示一个导航栏，右侧会显示一个顶层行，其中包含若干链接。



注意：Hawk2 中的可用功能

默认情况下，以 root 或 hacluster 身份登录的用户对所有群集配置任务具有完全读写访问权。不过，使用[访问控制列表 \(ACL\)](#) 可以定义细化的访问权限。

如果在 CRM 中启用了 ACL，Hawk2 中的可用功能取决于用户角色和指派给这些角色的访问权限。Hawk2 中的历史记录浏览器只能由用户 hacluster 来执行。

7.3.1 左侧导航栏

监控

- 状态：一目了然地显示当前的群集状态（与 `crmsh` 上的 `crm status` 作用类似）。有关详细信息，请参见第 7.8.1 节“[监视单个群集](#)”。如果群集包含 `guest` 节点（运行 `pacemaker_remote` 守护程序的节点），则也会显示这些节点。屏幕刷新频率接近实时刷新：任何节点或资源状态的更改都几乎立即可见。
- 仪表盘：可用于监视多个群集（如果您设置了 Geo 群集，还会位于多个不同站点）。有关详细信息，请参见第 7.8.2 节“[监视多个群集](#)”。如果群集包含 `guest` 节点（运行 `pacemaker_remote` 守护程序的节点），则也会显示这些节点。屏幕刷新频率接近实时刷新：任何节点或资源状态的更改都几乎立即可见。

查错

- 历史记录：打开历史记录浏览器，您可以从中生成群集报告。有关细节，请参见第 7.10 节“[查看群集历史记录](#)”。
- 命令日志：列出 Hawk2 最近执行的 `crmsh` 命令。

配置

- 添加资源：打开资源配置屏幕。有关细节，请参见第 7.5 节“[配置群集资源](#)”。
- 添加约束：打开约束配置屏幕。有关细节，请参见第 7.6 节“[配置约束](#)”。
- 向导：可让您从数个向导中选择一个，以便引导您完成成为某个工作负载（例如，某个 DRBD 块设备）创建资源的流程。有关细节，请参见第 7.5.2 节“[使用向导添加资源](#)”。
- 编辑配置：可用于编辑资源、约束、节点名称和属性、标记、警报 (http://crmsh.github.io/man/#cmdhelp_configure_alert) 和屏蔽拓扑 (http://crmsh.github.io/man/#cmdhelp_configure_fencing_topology)。
- 群集配置：可用于修改全局群集选项以及资源和操作默认值。有关细节，请参见第 7.4 节“[配置全局群集选项](#)”。

- 访问控制 > 角色：打开一个屏幕，您可在其中为访问控制列表（即用于描述对 CIB 的访问权限的规则集）创建角色。有关细节，请参见过程 12.2 “使用 Hawk2 添加 Monitor 角色”。
- 访问控制 > 目标：打开一个屏幕，您可在其中为访问控制列表创建目标（系统用户），并为这些目标指派角色。有关细节，请参见过程 12.3 “使用 Hawk2 向目标指派角色”。

7.3.2 顶层行

Hawk2 的顶层行显示以下条目：

- 批：单击可切换到批模式。可用于模拟和分阶段进行更改并通过单次事务应用这些更改。有关细节，请参见第 7.9 节 “使用批模式”。
- 用户名：可用于设置 Hawk2 的首选项（例如，Web 界面的语言，或者是否在 STONITH 处于禁用状态时显示警告）。
- 帮助：访问 SUSE Linux Enterprise High Availability Extension 文档、阅读发行说明或报告 Bug。
- 注销：单击可注销。

7.4 配置全局群集选项

全局群集选项控制群集在遇到特定情况时的行为方式。它们被分成若干组，可通过 Hawk2 和 crmsh 之类的群集管理工具来查看和修改。通常可保留预定义值。但为了确保群集的关键功能正常工作，需要在进行基本群集设置后调整以下参数：

- 全局选项 `no-quorum-policy`
- 全局选项 `stonith-enabled`

过程 7.3：修改全局群集选项

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 群集配置。

群集配置屏幕即会打开。屏幕中显示全局群集选项及其当前值。

要在屏幕右侧显示某个参数的简要描述，请将鼠标悬停在该参数上方。

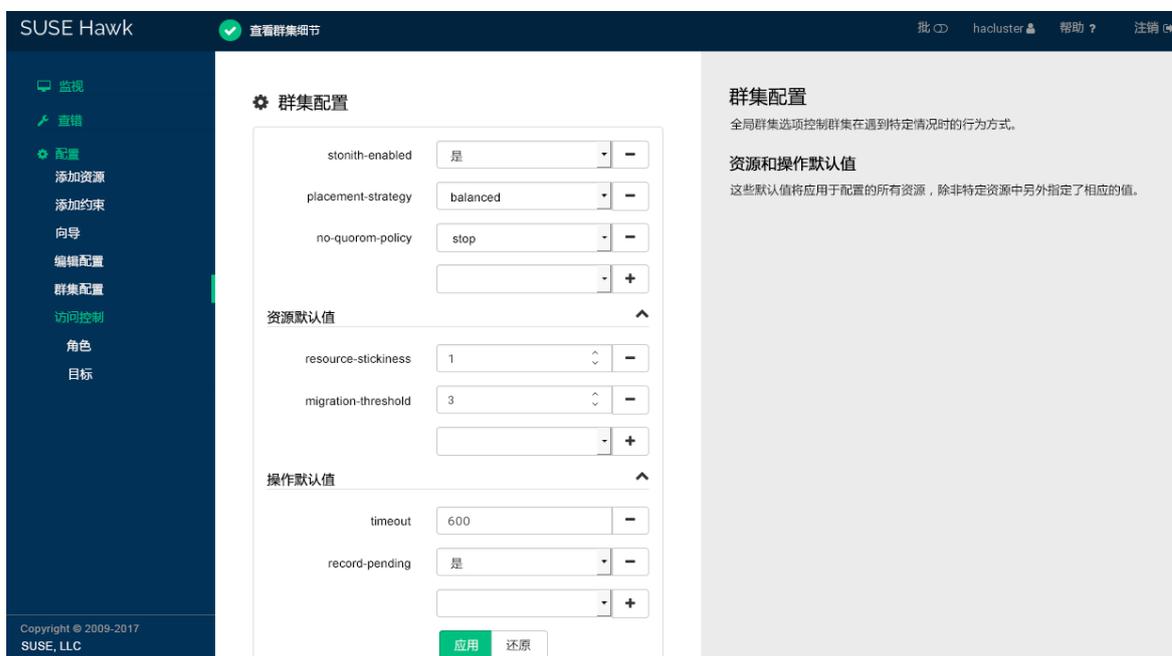


图 7.1：HAWK2 - 群集配置

3. 检查 no-quorum-policy 和 stonith-enabled 的值并根据需要进行调整。

- a. 将 no-quorum-policy 设置为合适的值。有关详细信息，请参见第 6.2.2 节“全局选项 no-quorum-policy”。
- b. 如果出于某些原因需要禁用屏蔽，请将 stonith-enabled 设置为 no。默认情况下，该参数设置为 true，因为执行常规的群集操作必须要使用 STONITH 设备。根据默认值，如果未配置 STONITH 资源，群集将拒绝启动任何资源。

❗ 重要：不支持无 STONITH 的配置

- 您必须为群集配置节点屏蔽机制。
- 全局群集选项 stonith-enabled 和 startup-fencing 必须设置为 true。如果您更改这些选项，将会失去支持。

- c. 要从群集配置中去除某个参数，请单击该参数旁边的减号图标。如果删除了某个参数，则群集的表现方式就像该参数采用默认值一样。
 - d. 要向群集配置添加新参数，请从下拉框中选择一个参数。
 4. 如果您需要更改资源默认值或操作默认值，请执行以下步骤：
 - a. 要调整某个值，请从下拉框中选择一个不同的值，或直接编辑该值。
 - b. 要添加新的资源默认值或操作默认值，请从空下拉框中选择一项，然后输入值。如果有默认值，Hawk2 会自动建议这些值。
 - c. 要去除某个参数，请单击该参数旁边的减号图标。如果没有为资源默认值和操作默认值指定值，群集会使用第 6.3.6 节“资源选项（元属性）”和第 6.3.8 节“资源操作”中所述的默认值。
 5. 确认更改。

7.5 配置群集资源

群集管理员需要为群集中服务器上运行的每个资源或应用程序创建群集资源。群集资源可包括网站、邮件服务器、数据库、文件系统、虚拟机和任何其他基于服务器的应用程序或对用户随时都可用的服务。

有关可创建的资源类型的概述，请参见第 6.3.3 节“资源类型”。当您指定资源基本信息（ID、类、提供程序和类型）后，Hawk2 会显示以下类别：

参数（实例属性）

确定资源控制的服务实例。有关更多信息，请参见第 6.3.7 节“实例属性（参数）”。

创建资源时，Hawk2 会自动显示所有必要的参数。对这些参数进行编辑，以便拥有有效的资源配置。

操作

为监视资源所需。有关更多信息，请参见第 6.3.8 节“资源操作”。

创建资源时，Hawk2 会显示最重要的资源操作（`monitor`、`start` 和 `stop`）。

元属性

告知 CRM 如何处理特定资源。有关更多信息，请参见第 6.3.6 节“资源选项（元属性）”。

创建资源时，Hawk2 会自动列出该资源的重要元属性，例如，定义资源初始状态的 `target-role` 属性。默认情况下，该属性设置为 `Stopped`，因此资源不会立即启动。

利用率

告知 CRM 某个资源需从节点获取的容量。有关更多信息，请参见第 7.6.8 节“根据负载影响配置资源放置”。

您可以在创建资源期间或之后调整这些类别中的项和值。

7.5.1 显示当前群集配置 (CIB)

群集管理员有时需要知道群集配置。Hawk2 可以 `crm` 外壳语法、XML 和图表形式显示当前配置。要查看 `crm` 外壳语法形式的群集配置，请从左侧导航栏中选择配置 > 编辑配置，并单击显示。要改为以原始 XML 显示配置，请单击 XML。单击示意图会以图表显示 CIB 中配置的节点和资源。它还会显示各资源之间的关系。

7.5.2 使用向导添加资源

Hawk2 向导是设置简单资源（如虚拟 IP 地址或 SBD STONITH 资源）的便捷方式。对于包含多个资源的复杂配置（例如 DRBD 块设备或 Apache Web 服务器的资源配置）而言，这种方法也十分有用。向导会引导您完成所有配置步骤，并提供您需要输入的参数的相关信息。

过程 7.4：使用资源向导

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 向导。

3. 单击各个类别旁边的向下箭头图标将其展开，然后选择所需的向导。

4. 按照屏幕指导执行操作。完成最后的配置步骤后，校验您所输入的值。

Hawk2 会显示它将执行的操作以及配置的最终成果。根据配置，您可能会收到提示要求输入 `root` 口令才能应用配置。

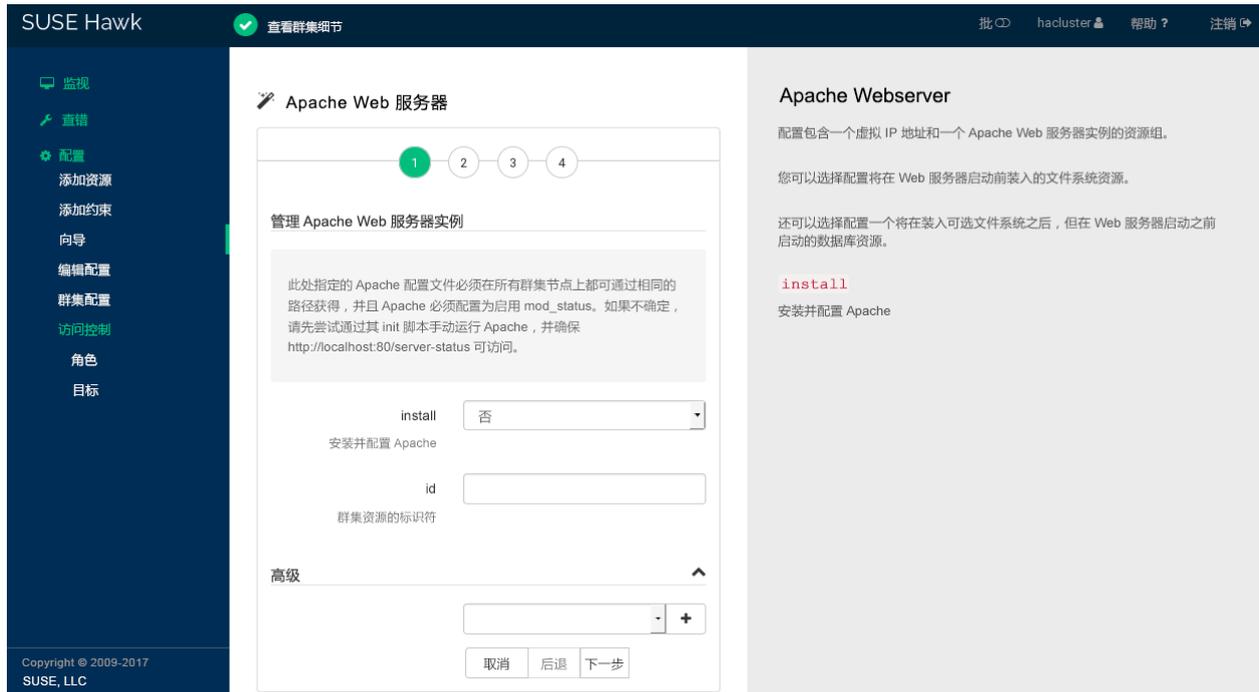


图 7.2：HAWK2 - 用于 APACHE WEB 服务器的向导

7.5.3 添加简单资源

要创建最基本类型的资源，请执行以下操作：

过程 7.5：添加原始资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加资源 > 原始资源。

3. 输入唯一的资源 ID。

4. 如果存在您要在其基础上设置资源配置的资源模板，请选择相应的模板。有关配置模板的细节，请参见过程 7.6 “添加资源模板”。

5. 选择要使用的资源代理类：`lsb`、`ocf`、`service`、`stonith` 或 `systemd`。有关详细信息，请参见第 6.3.2 节“支持的资源代理类”。
 6. 如果选择了 `ocf` 作为类，则指定 OCF 资源代理的提供程序。OCF 规范允许多个供应商供应相同的资源代理。
 7. 从类型列表中，选择要使用的资源代理（例如 `IPaddr` 或 `Filesystem`）。将显示该资源代理的简短描述。
- 如此，资源基本信息即指定完成。

注意

类型列表中提供的选项取决于您选择的类（对于 OCF 资源还取决于提供程序中选择的内容）。

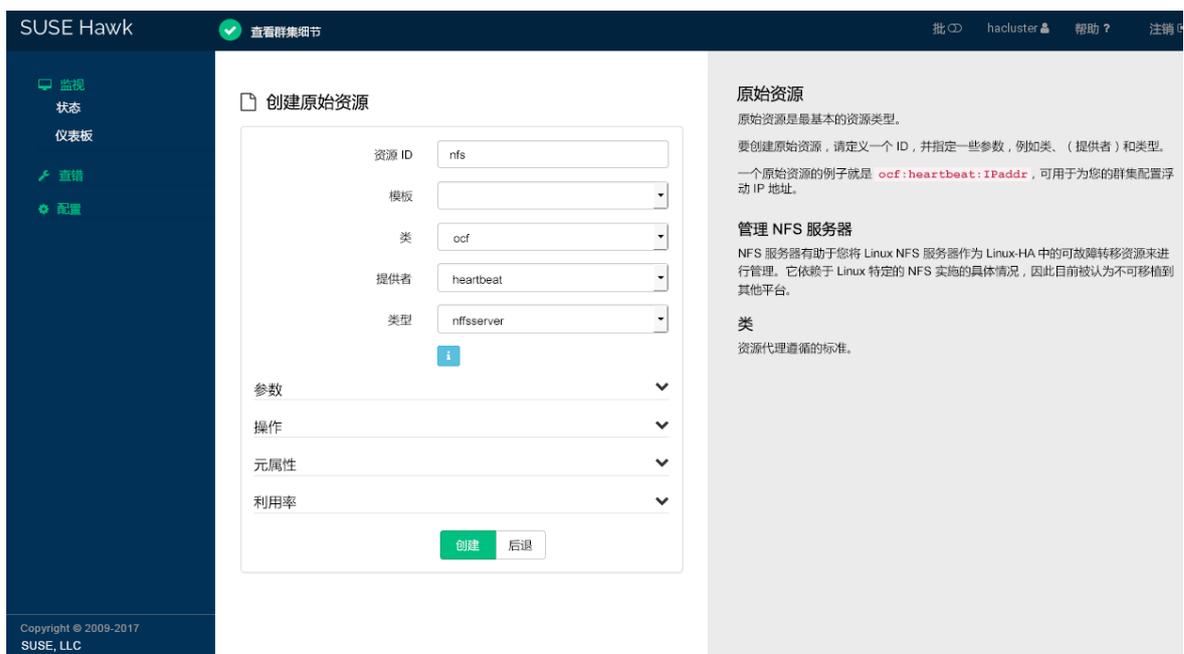


图 7.3：HAWK2 — 原始资源

8. 要保留 Hawk2 建议的参数、操作和元属性，请单击创建完成配置。如果操作成功，屏幕顶部会显示一条讯息。
- 要调整参数、操作或元属性，请参见第 7.5.5 节“修改资源”。要配置资源的利用率属性，请参见过程 7.21“配置资源所需的容量”。

7.5.4 添加资源模板

要创建大量具有类似配置的资源，定义资源模板是最简单的方式。定义后，便可在基元或特定类型的约束中引用它。有关功能及使用资源模板的详细信息，请参见第 6.5.3 节“资源模板和约束”。

过程 7.6：添加资源模板

配置资源模板就如同配置原始资源一样。

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加资源 > 模板。
3. 输入唯一的资源 ID。
4. 按照过程 7.5 “添加原始资源” 中的指导从步骤 5 开始。

7.5.5 修改资源

如果您之前创建了一个资源，可以随时根据需要调整参数、操作或元属性来编辑其配置。

过程 7.7：修改资源的参数、操作或元属性

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 在 Hawk2 的状态屏幕中，转到资源列表。
3. 在操作列中，单击要修改的资源或组旁边的向下箭头图标，然后选择编辑。
资源配置屏幕即会打开。

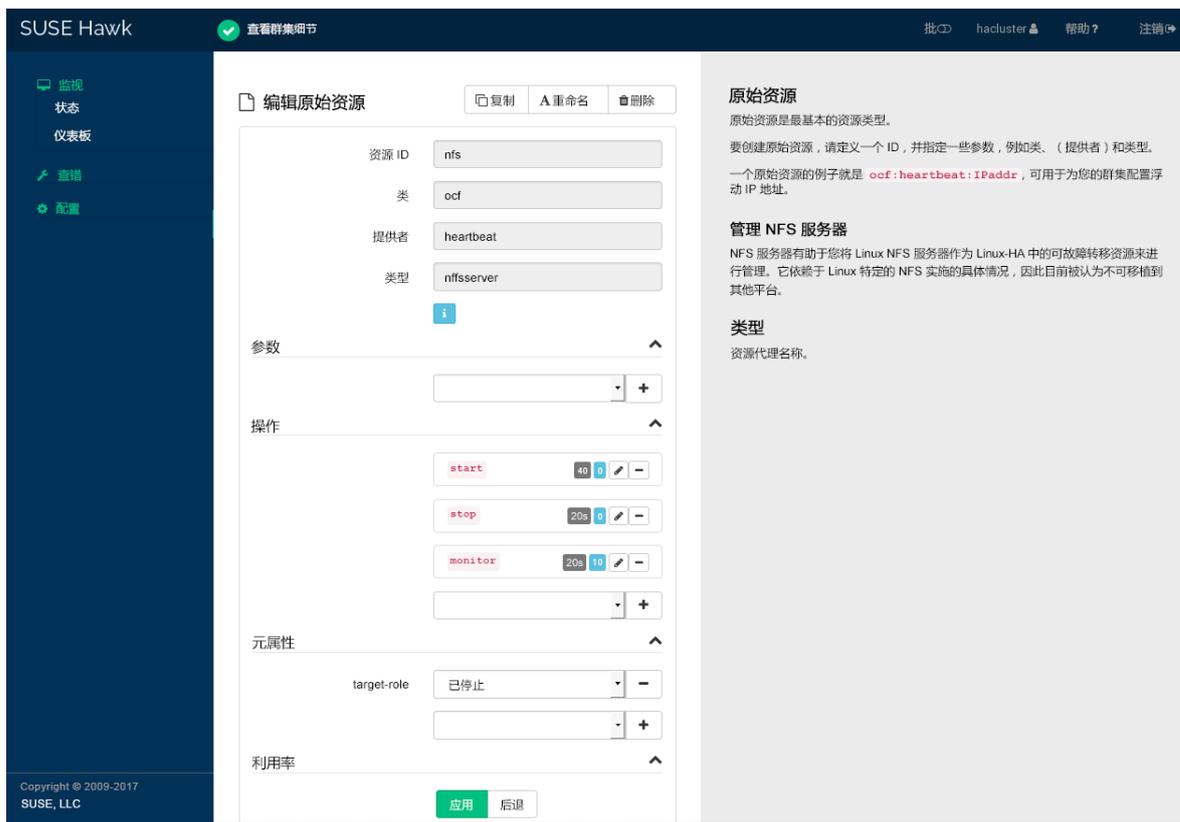


图 7.4：HAWK2 — 编辑原始资源

4. 要添加新参数、操作或元属性，请从空下拉框中选择一项。
5. 要编辑操作类别中的任何值，请单击相应项的编辑图标，为该操作输入不同的值，然后单击应用。
6. 完成时，单击资源配置屏幕中的应用按钮确认对参数、操作或元属性所做的更改。如果操作成功，屏幕顶部会显示一条讯息。

7.5.6 添加 STONITH 资源

! 重要：不支持无 STONITH 的配置

- 您必须为群集配置节点屏蔽机制。
- 全局群集选项 `stonith-enabled` 和 `startup-fencing` 必须设置为 `true`。如果您更改这些选项，将会失去支持。

默认情况下，全局群集选项 `stonith-enabled` 设置为 `true`。如果未定义任何 STONITH 资源，群集将拒绝启动任何资源。配置一个或多个 STONITH 资源以完成 STONITH 设置。要为 SBD、libvirt (KVM/Xen) 或 vCenter/ESX 服务器添加 STONITH 资源，最简单的方式就是使用 Hawk2 向导（请参见第 7.5.2 节“使用向导添加资源”）。虽然 STONITH 资源的配置过程与其他资源类似，但它们的行为在某些方面有所不同。有关细节，请参见第 10.3 节“STONITH 资源和配置”。

过程 7.8：添加 STONITH 资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加资源 > 原始资源。

3. 输入唯一的资源 ID。

4. 从类列表，选择资源代理类 `stonith`。

5. 从类型列表中，选择用于控制 STONITH 设备的 STONITH 插件。该插件的简短描述即会显示。

6. Hawk2 会自动显示该资源必需的参数。为每个参数输入值。

7. Hawk2 会显示最重要的资源操作并建议默认值。如果在此处不修改任何设置，Hawk2 会在您确认后立即添加建议的操作及其默认值。

8. 如无更改必要，请保留默认的元属性设置。

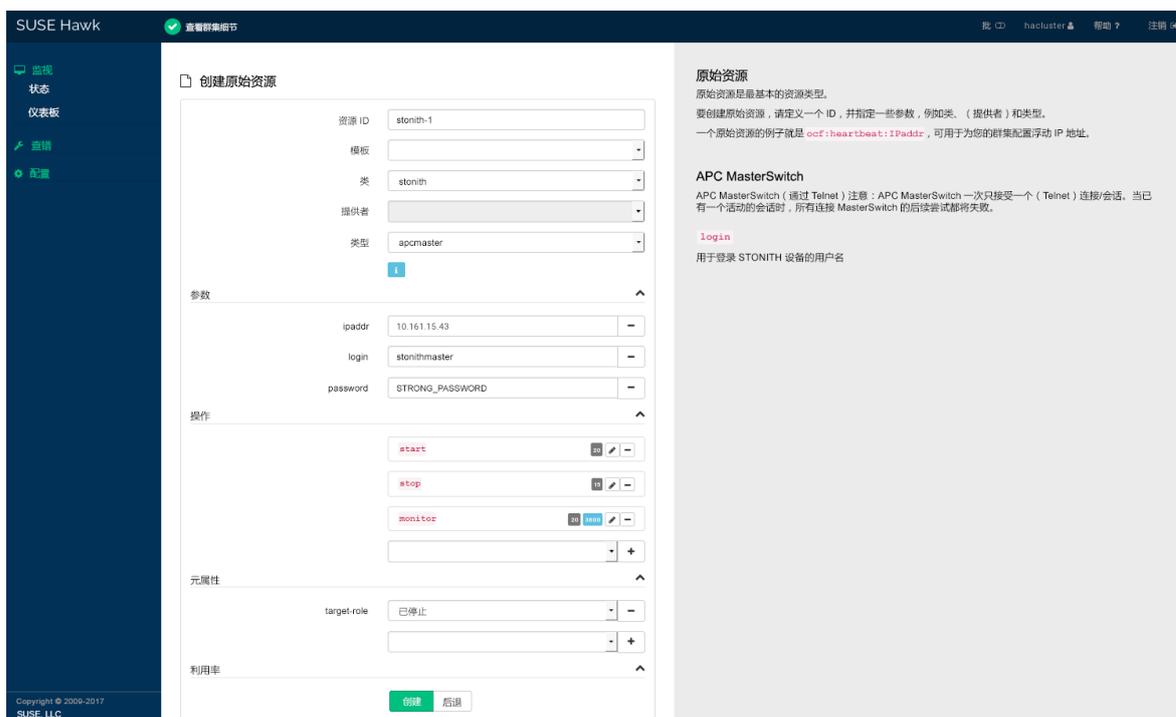


图 7.5 : HAWK2 – STONITH 资源

9. 确认更改以创建 STONITH 资源。

如果操作成功，屏幕顶部会显示一条讯息。

要完成屏蔽配置，请添加约束。有关详细信息，请参见第 10 章 “屏障和 STONITH”。

7.5.7 添加群集资源组

某些群集资源依赖于其他组件或资源，这些资源要求每个组件或资源按特定顺序启动，并在同一服务器上运行。为简化此配置，SUSE Linux Enterprise High Availability Extension 支持使用组。

资源组包含一组需要放在一起并按顺序启动和按相反顺序停止的资源。有关资源组的示例以及组及其属性的更多信息，请参见第 6.3.5.1 节 “组”。



注意：空组

组必须包含至少一个资源，否则配置无效。创建组时，Hawk2 允许您创建多个原始资源并将它们添加到组中。有关细节，请参见第 7.7.1 节 “编辑资源和组”。

过程 7.9：添加资源组

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加资源 > 组。

3. 输入唯一的组 ID。

4. 要定义组成员，请选择子项列表中的一项或多项。通过使用右侧的“手柄”图标将组成员拖放为需要的顺序对其进行重新排序。

5. 根据需要修改或添加元属性。

6. 单击创建以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

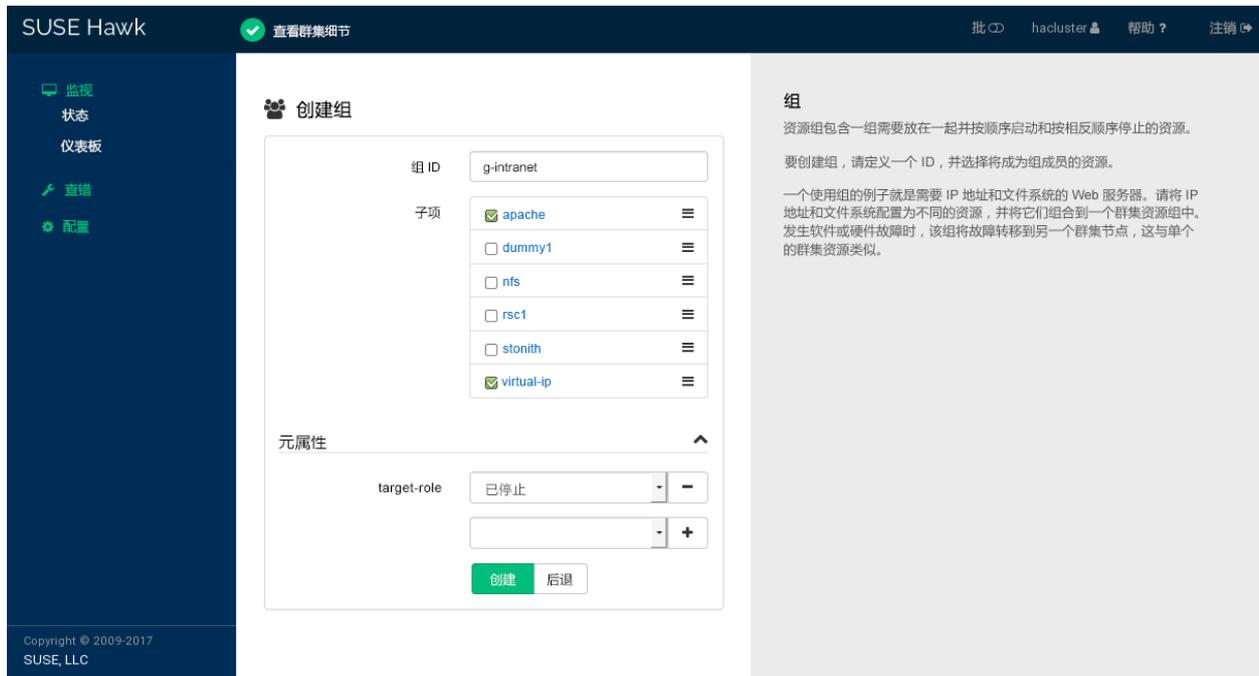


图 7.6：HAWK2 — 资源组

7.5.8 添加克隆资源

如果希望特定资源同时在群集中的多个节点上运行，请将这些资源配置为克隆。一个可配置为克隆的资源例子就是要作为克隆使用的 OCFS2 之类群集文件系统的 `ocf:pacemaker:controld` 任何普通的资源或资源组均可克隆。克隆资源的实例的行为方式可能一样。但其配置也可能有所不同，具体取决于克隆资源所在的节点。有关可用类型的资源克隆的概述，请参见第 6.3.5.2 节“克隆资源”。



注意：克隆资源的子资源

克隆资源可以包含原始资源或组作为子资源。在 Hawk2 中，创建克隆资源时不能创建或修改子资源。添加克隆资源之前，先创建子资源并根据需要配置它们。有关细节，请参见第 7.5.3 节“添加简单资源”或第 7.5.7 节“添加群集资源组”。

过程 7.10：添加克隆资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加资源 > 克隆。
3. 输入唯一的克隆 ID。
4. 从子资源列表中，选择原始资源或组作为克隆资源的子资源。
5. 根据需要修改或添加元属性。
6. 单击创建以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

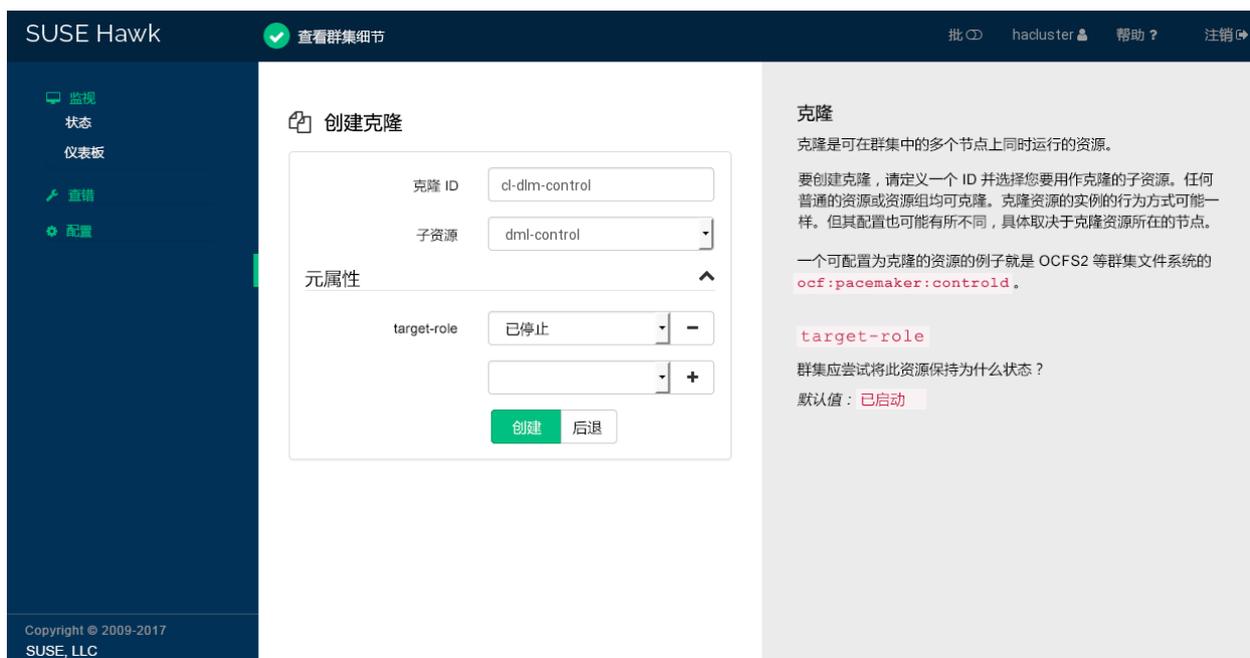


图 7.7：HAWK2 – 克隆资源

7.5.9 添加多状态资源

多状态资源是克隆的特殊形式。它们允许实例处于两种运行模式中的其中一种模式（称为 主动/被动、主要/次要、主/从）。多状态资源只能包含一个组或一个常规资源。

配置资源监视或约束时，多状态资源与简单资源具有不同的要求。有关细节，请参见 <http://www.clusterlabs.org/pacemaker/doc/> 上的《Pacemaker Explained》（Pacemaker 配置说明）。请参见“Multi-state - Resources That Have Multiple Modes”（多状态 - 具有多个节点的资源）一节。

注意：多状态资源的子资源

多状态资源可以包含原始资源或组作为子资源。在 Hawk2 中，在创建多状态资源时不能创建或修改子资源。添加多状态资源之前，先创建子资源并根据需要配置它们。有关细节，请参见第 7.5.3 节“添加简单资源”或第 7.5.7 节“添加群集资源组”。

过程 7.11：添加多状态资源

1. 登录 Hawk2:

https://HAWKSERVER:7630/

2. 从左侧导航栏中，选择配置 > 添加资源 > 多状态。
3. 输入唯一的多状态 ID。
4. 从子资源列表中，选择原始资源或组作为多状态资源的子资源。
5. 根据需要修改或添加元属性。
6. 单击创建以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

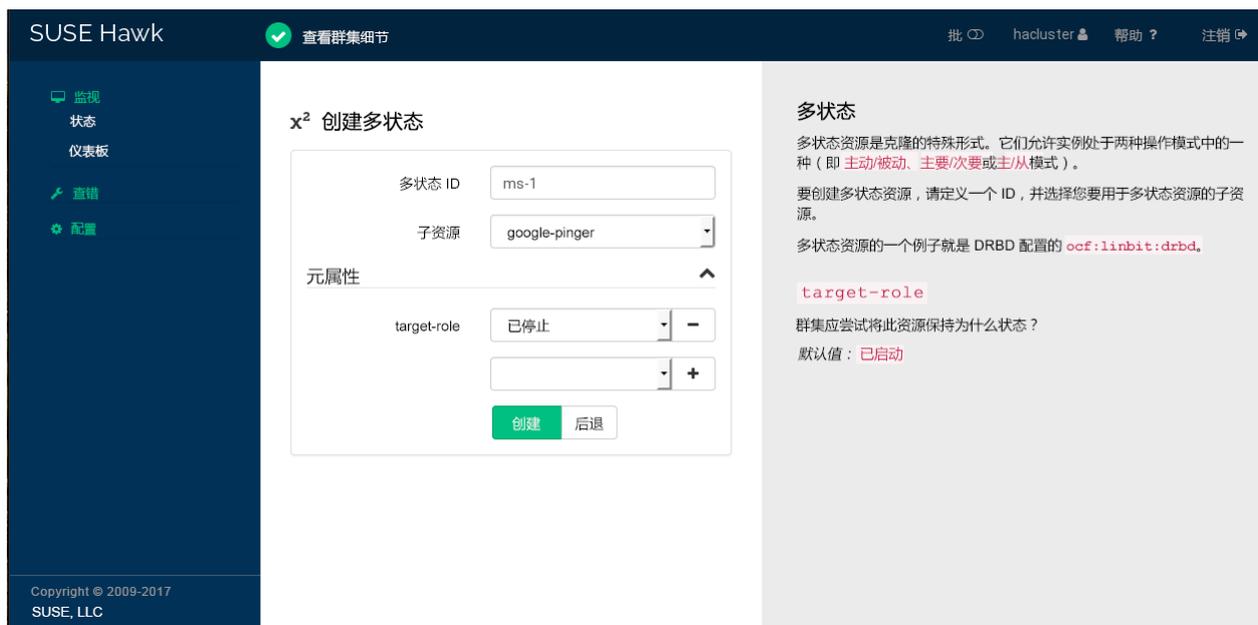


图 7.8：HAWK2 – 多状态资源

7.5.10 使用标记分组资源

使用标记可以一次性引用多个资源，而无需在这些资源之间创建任何共置或顺序关系。您可以使用标记对概念上相关的资源分组。例如，如果您有多个资源与某个数据库相关，可以将所有相关的资源添加到名为 数据库 的标记中。

属于一个标记的所有资源都可通过一个命令启动或停止。

过程 7.12：添加标记

1. 登录 Hawk2:

https://HAWKSERVER:7630/

2. 从左侧导航栏中，选择配置 > 添加资源 > 标记。
3. 输入唯一的标记 ID。
4. 从对象列表中，选择要使用标记引用的资源。
5. 单击创建以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

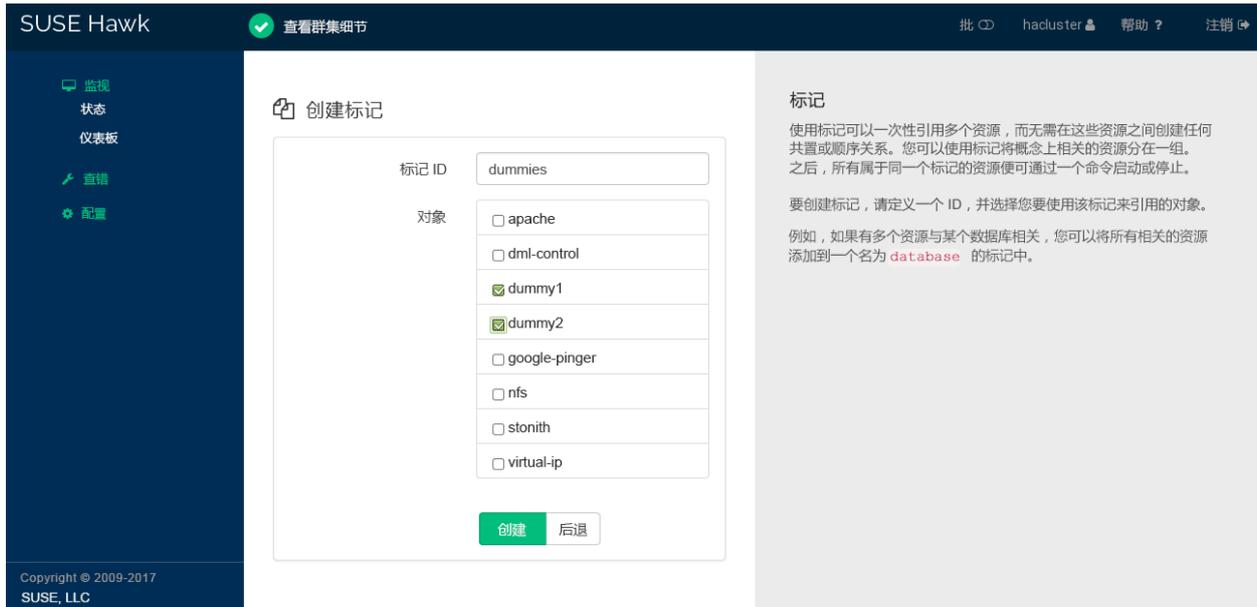


图 7.9：HAWK2 - 标记

7.5.11 配置资源监视

High Availability Extension 不仅会检测到节点故障，还会检测到节点上各资源的失败情况。如果要确保资源正在运行，则为其配置资源监视。通常，资源仅会在运行时受到群集的监视。但是，为了检测并发违例，还需为停止的资源配置监视。要进行资源监视，请指定超时和/或启动延迟值及间隔。间隔告诉 CRM 检查资源状态的频率。您还可以设置特定参数，如为 `start` 或 `stop` 操作设置 `timeout`。

过程 7.13：添加和修改操作

1. 登录 Hawk2:

- 按过程 7.5 “添加原始资源” 中所述添加资源，或选择要编辑的现有原始资源。Hawk2 会自动显示最重要的操作（start、stop、monitor）并建议默认值。要查看属于每个建议值的属性，请将鼠标悬停在相应的值上。



- 要更改针对 start 或 stop 操作建议的 timeout 值，请执行以下操作：
 - 单击操作旁边的钢笔图标。
 - 在打开的对话框中，为 timeout 参数输入不同的值，例如 10，然后确认您的更改。
- 要更改针对 monitor 操作建议的间隔值，请执行以下操作：
 - 单击操作旁边的钢笔图标。
 - 在打开的对话框中，为监控 interval 输入不同的值。
 - 要配置资源停止时针对资源的监视，请执行以下操作：
 - 从下面的空下拉框中选择 role 这一项。
 - 从 角色 下拉框中，选择 已停止。
 - 单击应用确认更改并关闭操作对话框。
- 在资源配置屏幕中确认更改。如果操作成功，屏幕顶部会显示一条讯息。

有关在资源监视程序检测到故障时将发生的流程，请参见第 6.4 节 “资源监视”。

要查看资源故障，请切换到 Hawk2 中的状态屏幕，然后选择您感兴趣的资源。在操作列中，单击向下箭头图标并选择最近的事件。随后打开的对话框会列出对资源执行的最近操作。失败事件显示为红色。要查看资源细节，请单击操作列中的放大镜图标。

Q nfs
原始资源

代理 ocf:heartbeat:nfsserver

元属性 ▼

target-role	已停止
-------------	-----

操作 ▼

名称	超时	间隔
起始	40	0
停止	20s	0
显示器	20s	10

约束 ▼

ID	类型	分数	针对

关闭

图 7.10 : HAWK2 - 资源细节

7.6 配置约束

配置所有资源后，指定群集应如何正确地处理它们。使用资源约束可指定资源可以在哪些群集节点上运行、以何顺序装载资源，以及特定资源依赖于其他哪些资源。

有关约束的可用类型的概述，请参见第 6.5.1 节“约束类型”。定义约束时，还需要指定分数。有关分数及其在群集中的含义的更多信息，请参见第 6.5.2 节“分数和无限值”。

7.6.1 添加位置约束

位置约束决定资源可在哪个节点上运行、优先在哪个节点上运行，或者不能在哪个节点上运行。将与某个数据库相关的所有资源存放在同一个节点上，就是位置约束的一个示例。

过程 7.14：添加位置约束

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加约束 > 位置。

3. 输入唯一的约束 ID。

4. 从资源列表中，选择要为其定义约束的一个或多个资源。

5. 输入一个分数。分数表示您指派给此资源约束的值。正值表示资源可以在下一步中指定的节点上运行。负值表示它不应在该节点上运行。分数较高的约束先应用，分数较低的约束后应用。

也可以通过下拉框设置某些常用值：

- 要强制资源在该节点上运行，请单击箭头图标并选择 始终。如此会将分数设置为 INFINITY。
- 如果不想让资源在该节点上运行，请单击箭头图标并选择 从不。如此会将分数设置为 -INFINITY，表示资源不得在该节点上运行。
- 要将分数设置为 0，请单击箭头图标并选择 建议。这样便会禁用约束。如果您要设置资源发现，但又不想约束资源，便可使用此方法。

6. 选择一个节点。

7. 单击创建以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

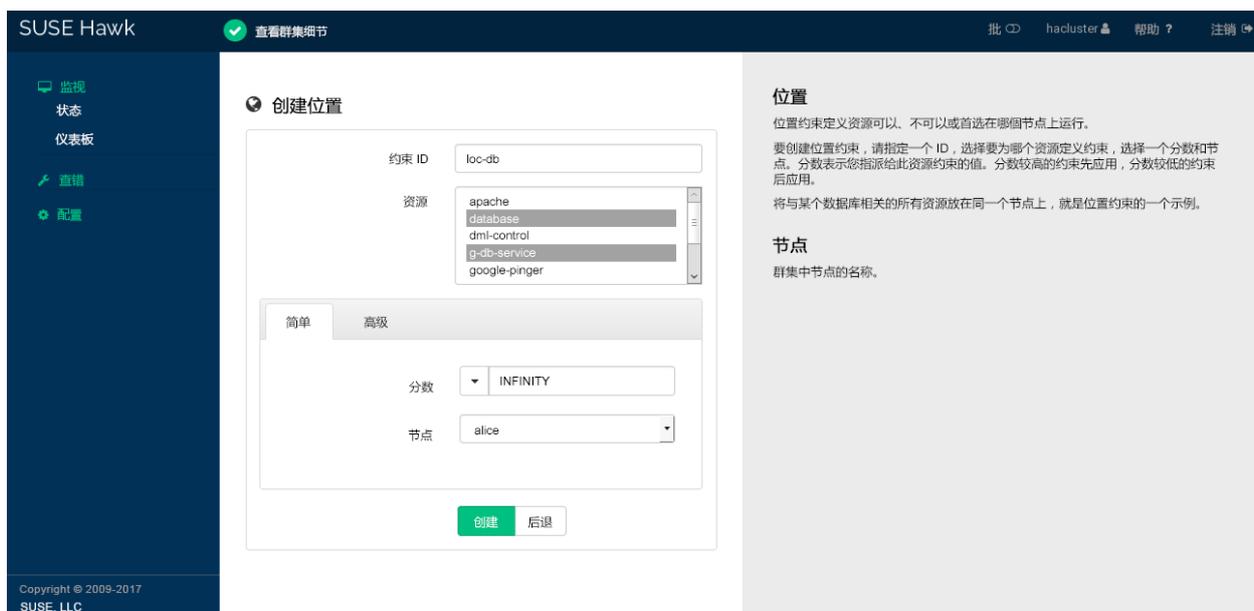


图 7.11 : HAWK2 — 位置约束

7.6.2 添加共置约束

共置约束告知群集哪些资源可以或不可以在一个节点上一起运行。由于共置约束定义了资源之间的依赖性，因此您至少需要两个资源才能创建共置约束。

过程 7.15 : 添加共置约束

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 添加约束 > 共置。

3. 输入唯一的约束 ID。

4. 输入一个分数。分数决定资源之间的位置关系。正值表示多个资源应在同一个节点上运行。负值表示多个资源不应在同一个节点上运行。分数将与其他因数结合使用，以确定放置资源的位置。

也可以通过下拉框设置某些常用值：

- 要强制资源在同一个节点上运行，请单击箭头图标并选择 始终。如此会将分数设置为 INFINITY。
 - 如果不想让多个资源在同一个节点上运行，请单击箭头图标并选择 从不。如此会将分数设置为 -INFINITY，表示资源不得在同一个节点上运行。
5. 要为约束定义资源，请执行以下步骤：
 - a. 从资源类别的下拉框中，选择某个资源（或模板）。
系统即会添加该资源，并且下面会出现一个新的空下拉框。
 - b. 重复此步骤添加更多资源。
由于最上面的资源依赖于下一个资源，依此类推，因此群集将首先决定最后一个资源的放置位置，然后再根据该决定放置依赖的资源。如果无法满足约束，群集可能不允许运行依赖资源。
 - c. 要交换共置约束中资源的顺序，请单击一个资源旁边的向上箭头图标，将其与上方的项目加以交换。
 6. 如果需要，请指定每个资源的更多参数（例如 Started、Stopped、Master、Slave、Promote、Demote）：单击资源旁边的空下拉框并选择所需项。
 7. 单击 创建 以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

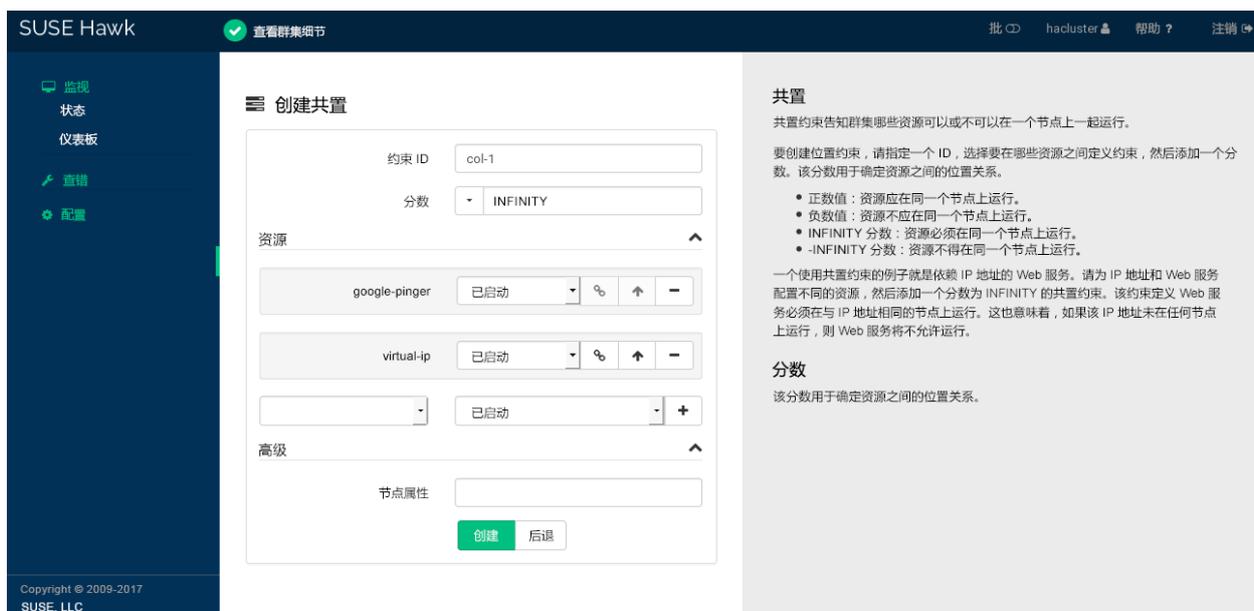


图 7.12：HAWK2 — 共置约束

7.6.3 添加顺序约束

顺序约束定义启动和停止资源的顺序。由于顺序约束定义了资源之间的依赖性，因此您至少需要两个资源才能创建顺序约束。

过程 7.16：添加顺序约束

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 在左侧导航栏中，选择配置 > 添加约束 > 顺序。
3. 输入唯一的约束 ID。
4. 输入一个分数。如果分数大于零，则顺序约束为强制性的，否则为选择性的。
也可以通过下拉框设置某些常用值：

- 如果要将顺序约束设为强制性，请单击箭头图标并选择 强制。
 - 如果只想将顺序约束作为建议，请单击箭头图标并选择 可选。
 - 序列化：要确保不会同时对资源执行两个停止/启动操作，请单击箭头图标并选择 序列化。如此可确保一个资源完成启动操作后，另一个资源方可启动。典型的使用案例是启动期间在主机上产生高负载的资源。
5. 对于顺序约束，通常可将选项对称保持为启用状态。这指定了资源以相反顺序停止。
 6. 要为约束定义资源，请执行以下步骤：
 - a. 从资源类别的下拉框中，选择某个资源（或模板）。
系统即会添加该资源，并且下面会出现一个新的空下拉框。
 - b. 重复此步骤添加更多资源。
排在最前面的资源最先启动，然后是排在第二的资源，以此类推。通常资源将以相反顺序停止。
 - c. 要交换顺序约束中资源的顺序，请单击一个资源旁边的向上箭头图标，将其与上方的项目加以交换。
 7. 如果需要，请指定每个资源的更多参数（例如 Started、Stopped、Master、Slave、Promote、Demote）：单击资源旁边的空下拉框并选择所需项。
 8. 确认更改以完成配置。如果操作成功，屏幕顶部会显示一条讯息。

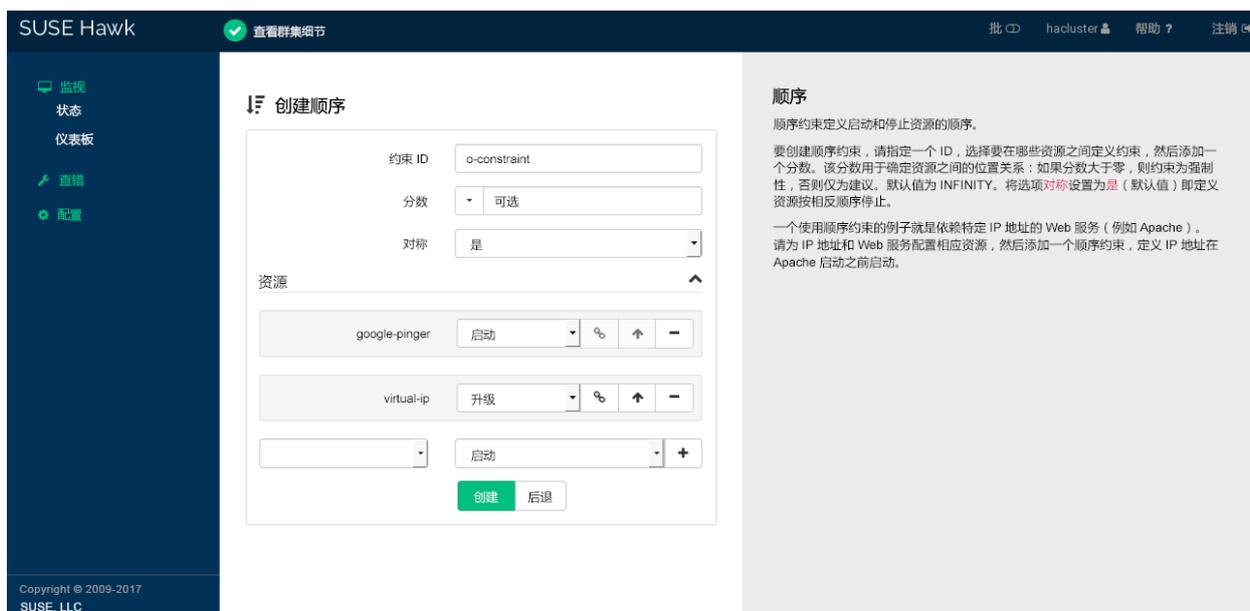


图 7.13 : HAWK2 — 顺序约束

7.6.4 在约束中使用资源集

可以使用资源集作为定义约束的备用形式。它们有与组相同的排序语义。

过程 7.17 : 为约束使用资源集

1. 要在位置约束中使用资源集，请执行以下操作：
 - a. 按过程 7.14 “添加位置约束”中所述操作，但步骤 4 除外：不要选择单个资源，而是按住 **Ctrl** 或 **Shift** 同时单击鼠标选择多个资源。这样便会在位置约束中创建一个资源集。
 - b. 要从位置约束中去除某个资源，请按住 **Ctrl** 并再次单击该资源，以将其取消选中。
2. 要在共置或顺序约束中使用资源集，请执行以下操作：
 - a. 按过程 7.15 “添加共置约束”或过程 7.16 “添加顺序约束”中所述操作，但为约束定义资源的步骤（步骤 5.a 或步骤 6.a）除外：
 - b. 添加多个资源。

- c. 要创建资源集，请单击某个资源旁边的链形图标将其与上方的资源链接起来。资源集通过属于集合的资源周围的框架显现。
- d. 您可以在一个资源集中组合多个资源，或创建多个资源集。

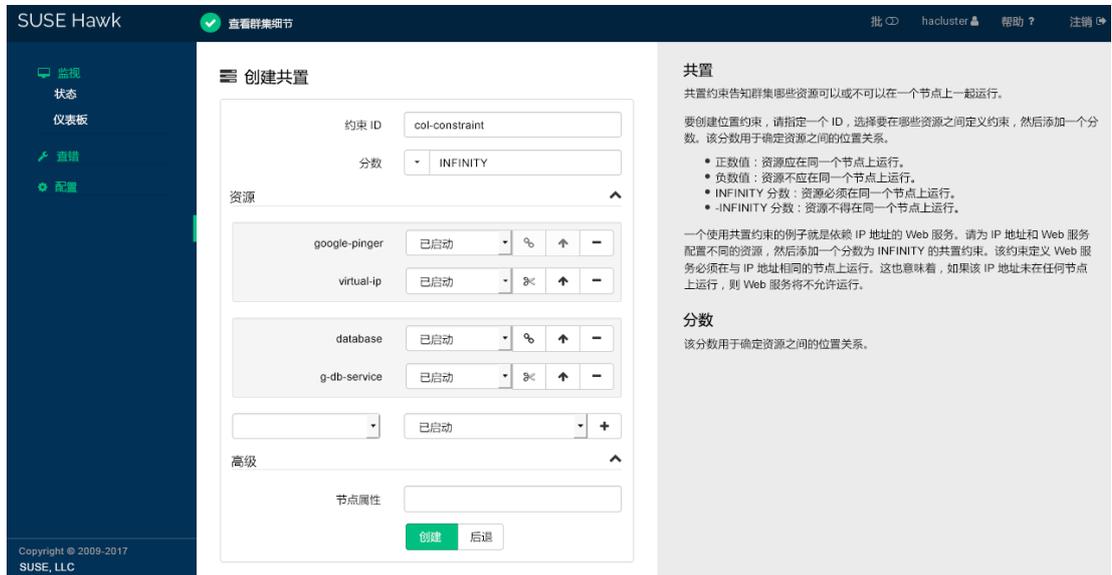


图 7.14 : HAWK2 - 一个共置约束中的两个资源集

- e. 要将某个资源与其上方的资源解除链接，请单击该资源旁边的剪刀图标。

3. 确认更改以完成约束配置。

7.6.5 更多信息

有关配置约束的更多信息以及顺序和共置基本概念的详细背景信息，请参见 <http://www.clusterlabs.org/pacemaker/doc/> 上提供的文档：

- 《Pacemaker Explained》（Pacemaker 配置说明）中的“Resource Constraints”（资源约束）一章
- 《Colocation Explained》（共置说明）
- 《Ordering Explained》（顺序说明）

7.6.6 指定资源故障转移节点

资源在出现故障时会自动重新启动。如果在当前节点上无法实现此操作，或者此操作在当前节点上失败了 N 次，它将尝试故障转移到其他节点。您可以定义资源的故障次数（migration-threshold），在该值之后资源会迁移到新节点。如果群集中存在两个以上的节点，则由 High Availability 软件选择特定资源故障转移的节点。

可按照以下步骤指定资源将故障转移到的特定节点：

过程 7.18：指定故障转移节点

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 按过程 7.14 “添加位置约束” 中所述，为资源配置位置约束。

3. 按过程 7.7: 修改资源的参数、操作或元属性中的步骤 4 所述为资源添加 migration-threshold 元属性，并输入 migration-threshold 的值。值应是小于 INFINITY 的正数。

4. 如果希望资源的失败计数自动失效，请按过程 7.5: 添加原始资源中的步骤 4 所述为该资源添加 failure-timeout 元属性，并输入 failure-timeout 的值。

要创建原始资源，请定义一个 ID，并指定一些参数，例如类、（提供者）和类型。一个原始资源的例子就是 `ocf:heartbeat:IPaddr`，可用于为您的群集配置浮动 IP 地址。

示例无状态资源代理

这是一个虚拟的资源代理。它除了会跟踪自己是否在运行外，实际上并不执行任何操作。运行它的目的是为了测试以及为 RA 编写器提供一个模板。

注意：请注意下面的操作部分指定的超时。它们应该对代理所管理的资源种类有意义。它们应该是建议的最小超时，但不应不能涵盖所有可能的资源实例。因此，请避免过度宽松，也不要太过严格，而是应保持适度。最小超时值不得小于 10 秒。

类型

资源代理名称

5. 如果希望为资源指定更多的首选故障转移节点，请创建更多的位置约束。

例 6.8 “迁移阈值 - 流程” 中演示了有关迁移阈值和失败计数的流程。

您可以随时手动清理资源的失败计数，而不是让资源的失败计数自动失效。有关详细信息，请参见第 7.7.3 节 “清理资源”。

7.6.7 指定资源故障回复节点（资源粘性）

当原始节点恢复联机并位于群集中时，资源可能会故障回复到该节点。要防止这种情况发生或指定其他节点供资源故障回复，请更改资源的粘性值。您可以在创建资源时或之后指定资源粘性。

有关不同资源粘性值的含义，请参见第 6.5.5 节 “故障回复节点”。

过程 7.19：指定资源粘性

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 按过程 7.7: 修改资源的参数、操作或元属性步骤 4 所述，为资源添加 `resource-stickiness` 元属性。

3. 为 `resource-stickiness` 指定介于 `-INFINITY` 和 `INFINITY` 之间的值。

The screenshot shows the SUSE Hawk web interface for editing a resource. The main content area is titled "编辑原始资源" (Edit Original Resource). It contains several input fields for resource configuration:

- 资源 ID (Resource ID): simple-testresource
- 类 (Class): ocf
- 提供者 (Provider): heartbeat
- 类型 (Type): 虚拟 (Virtual)

Below these fields are sections for "参数" (Parameters), "操作" (Operations), "元属性" (Metadata), and "利用率" (Usage). The "元属性" section is expanded to show the "resource-stickiness" parameter, which is currently set to 5000. There are plus and minus buttons next to the input field to adjust the value.

At the bottom of the configuration area are buttons for "应用" (Apply), "还原" (Reset), and "后退" (Back).

On the right side of the interface, there is a help panel for the "resource-stickiness" parameter. It explains that this is a virtual resource agent and provides a default value of 0. The text reads: "资源继续保留在所在位置的可能性? 默认值: 0".

7.6.8 根据负载影响配置资源放置

并非所有资源都相等。某些资源（如 Xen guest）需要托管它们的节点满足其容量要求。如果资源的放置导致其组合需求超过了提供的容量，则资源的性能会降低，或者失败。

要考虑此情况，可使用 High Availability Extension 指定以下参数：

1. 特定节点提供的容量。
2. 特定资源需要的容量。
3. 资源放置整体策略。

有关更多细节和配置示例，请参见第 6.5.6 节“根据资源负载影响放置资源”。

利用率属性用于配置资源的要求及节点提供的容量。您需要先配置节点的容量，然后才能配置资源所需的容量。

过程 7.20：配置节点提供的容量

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 状态。
3. 在节点选项卡上，选择要配置其容量的节点。
4. 在操作列中，单击向下箭头图标并选择编辑。
编辑节点屏幕即会打开。
5. 在利用率下，将利用率属性的名称输入到空下拉框中。
该名称可以是任意的（例如 RAM_in_GB）。
6. 单击添加图标添加属性。
7. 在属性旁边的空文本框中，输入一个属性值。该值必须是整数。



8. 添加所需数量的利用率属性，并为其添加相应的值。
9. 确认更改。如果操作成功，屏幕顶部会显示一条讯息。

过程 7.21：配置资源所需的容量

请在创建原始资源或编辑现有原始资源时配置特定资源需从节点中获取的容量。

您需要先按过程 7.20 中所述设置群集节点的利用率属性，之后才能将利用率属性添加到资源。

1. 登录 Hawk2:

`https://HAWKSERVER:7630/`

2. 要将利用率属性添加到现有资源：按第 7.7.1 节“编辑资源和组”中所述转到管理 > 状态并打开资源配置对话框。
如果要创建新资源：转到配置 > 添加资源，然后按第 7.5.3 节“添加简单资源”中所述继续操作。
3. 在资料配置对话框中，转到利用率类别。
4. 从空下拉框中，选择您在过程 7.20 中已为节点配置的其中一个利用率属性。
5. 在属性旁边的空文本框中，输入一个属性值。该值必须是整数。

6. 添加所需数量的利用率属性，并为其添加相应的值。
7. 确认更改。如果操作成功，屏幕顶部会显示一条讯息。

配置节点提供的容量以及资源所需的容量之后，请在全局群集选项中设置布局策略。否则，容量配置不起作用。可使用多个策略来调度负载：例如，可以将负载集中到尽可能少的节点上，或使其均匀分布在所有可用节点上。有关更多信息，请参见第 6.5.6 节“根据资源负载影响放置资源”。

过程 7.22：设置放置策略

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 群集配置以打开相应的屏幕。该屏幕会显示全局群集选项和资源，以及操作默认值。
3. 从屏幕上部的空下拉框中选择 `placement-strategy`。
默认情况下，其值会设置为默认，这表示不考虑利用率属性和值。
4. 根据要求，将放置策略设置为适当值。
5. 确认更改。

7.7 管理群集资源

除了配置群集资源以外，Hawk2 还可用于从状态屏幕管理现有资源。有关该屏幕的一般性概述，请参见第 7.8.1 节“监视单个群集”。

7.7.1 编辑资源和组

如果您需要编辑现有资源，请转到状态屏幕。在操作列中，单击要修改的资源或组旁边的向下箭头图标，然后选择编辑。

编辑屏幕即会显示。如果要编辑原始资源，则以下操作可用：

针对原始资源的操作

- 复制资源。

- 重命名资源（更改其 ID）。
- 删除资源。

如果要编辑组，则以下操作可用：

针对组的操作

- 创建要添加到此组的新原始资源。
- 重命名组（更改其 ID）。
- 通过使用右侧的“手柄”图标将组成员拖放为需要的顺序对其进行排序。

7.7.2 启动资源

启动群集资源之前，应确保资源设置正确。例如，如果使用 Apache 服务器作为群集资源，请先设置 Apache 服务器。完成 Apache 配置，然后再启动群集中的相应资源。



注意：不要处理由群集管理的服务

通过 High Availability Extension 管理资源时，不得以其他方式（例如，在群集外手动或者引导时或重引导时）启动或停止该资源。High Availability Extension 软件负责所有服务的启动或停止操作。

但如果要检查服务是否配置正确，可手动启动该服务，不过请确保在 High Availability Extension 接管前再次停止该服务。

要对群集当前管理的资源进行干预，请先将资源设置为 维护模式。有关细节，请参见[过程 16.5 “使用 Hawk2 将资源置于维护模式”](#)。

使用 Hawk2 创建资源时，可通过 target-role 元属性设置其初始状态。如果将其值设置为 stopped，则该资源在创建后不会自动启动。

过程 7.23：启动新资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 状态。资源列表还会显示状态。
3. 选择要启动的资源。在其操作列中，单击启动图标。要继续，请确认显示的消息。资源启动后，Hawk2 会将资源的状态变为绿色，并显示当前运行该资源的节点。

7.7.3 清理资源

如果资源失败，它会自动重新启动，但每次失败都会增加资源的失败计数。

如果已为资源设置 `migration-threshold`，当失败次数达到迁移阈值时，节点将不再运行该资源。

资源的失败计数可以通过系统自动重置（通过设置资源的 `failure-timeout` 选项），也可如下所述手动重置。

过程 7.24：清理资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择状态。资源列表还会显示状态。
3. 转到要清理的资源。在操作列中，单击向下箭头按钮并选择清理。要继续，请确认显示的消息。
如此即会执行 `crm resource cleanup` 命令并在所有节点上清理该资源。

7.7.4 删除群集资源

如果需要从群集中删除资源，请遵循以下过程以免出现配置错误：

过程 7.25：删除群集资源

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 按过程 7.24 “清理资源” 中所述清理所有节点上的资源。
3. 停止资源:

- a. 从左侧导航栏中，选择监视 > 状态。资源列表还会显示状态。
- b. 在操作列中，单击资源旁边的停止按钮。
- c. 要继续，请确认显示的消息。
资源停止后，状态列将会反映此变化。

4. 删除资源：

- a. 从左侧导航栏中，选择配置 > 编辑配置。
- b. 在资源列表中，转到相应资源。在操作列中，单击资源旁边的删除图标。
- c. 要继续，请确认显示的消息。

7.7.5 迁移群集资源

如第 7.6.6 节“指定资源故障转移节点”中所述，如果软件或硬件发生故障，群集会根据可自定义的特定参数（例如，迁移阈值或资源粘性），自动进行资源故障转移（迁移）。您还可以手动将资源迁移到群集中的其他节点。或者，可以将它移出当前节点，并让群集决定要将它放置到哪个位置。

过程 7.26：手动迁移资源

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 状态。资源列表还会显示状态。
3. 在资源列表中，选择相应资源。
4. 在操作列中，单击向下箭头按钮并选择迁移。
5. 随后打开的窗口中会提供以下选项：
 - 离开当前节点：此选项会为当前节点创建一个分数为 -INFINITY 的位置约束。
 - 或者，您也可以将资源移到另一节点上。此选项将为目标节点创建分数为 INFINITY 的位置约束。

6. 确认您的选择。

要使资源重新移回，请按如下操作：

过程 7.27：取消迁移资源

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 状态。资源列表还会显示状态。

3. 在资源列表中，转到相应资源。

4. 在操作列中，单击向下箭头按钮并选择清除。要继续，请确认显示的消息。

Hawk2 使用 `crm_resource --clear` 命令。资源可以移回到其原始位置，也可以留在当前位置（取决于资源粘性）。

有关详细信息，请参见 <http://www.clusterlabs.org/pacemaker/doc/> 上的《Pacemaker Explained》（Pacemaker 配置说明）。请参见“Resource Migration”（资源迁移）一节。

7.8 监视群集

Hawk2 提供不同的屏幕用于监视单个群集和多个群集：状态和仪表盘屏幕。

7.8.1 监视单个群集

要监视单个群集，请使用状态屏幕。当您登录 Hawk2 后，默认会显示状态屏幕。右上角的图标会一目了然地显示群集状态。如需更多细节，请查看以下类别：

错误

如果发生了错误，会显示在页面顶部。

资源

显示配置的资源，包括它们的状态、名称 (ID)、位置（运行资源的节点）和资源代理类型。在操作列中，您可以启动或停止资源，触发多个操作或查看细节。可以触发的操作包括将资源设置为维护模式（或去除维护模式）、将其迁移到其他节点、清理资源、显示任何最近的事件，或编辑资源。

节点

显示属于您登录的群集站点的节点，包括节点的状态和名称。在维护和待机列中，您可以为节点设置或去除 维护 或 待机 标志。操作列可用于查看节点的最近事件或更多细节：例如，查看是否为相应节点设置了 utilization、standby 或 maintenance 属性。

票据

仅当已配置了票据的情况下才显示（用于与 Geo 群集配合使用）。

状态	名称	位置	类型	操作
+	apache		ocf:heartbeat:Dummy	
+	database	kemter-3	ocf:heartbeat:Dummy	
+	dml-control	kemter-3	ocf:heartbeat:Dummy	
+	g-db-service		组 (2)	
+	google-pinger		ocf:heartbeat:Dummy	
+	nfs		ocf:heartbeat:nfsserver	
+	stonith		stonith:external/ipmi	
+	virtual-ip		ocf:heartbeat:Dummy	

图 7.15 : HAWK2 - 群集状态

7.8.2 监视多个群集

要监视多个群集，请使用 Hawk2 仪表盘。仪表盘屏幕中显示的群集信息储存在服务器端。群集节点之间会同步这些信息（如果已配置群集节点之间的无口令 SSH 访问权限）。有关详细信息，请参见第 D.2 节“配置无口令 SSH 帐户”。不过，运行 Hawk2 的计算机甚至不需要属于任何群集也可实现该目的，它可以是不相关的独立系统。

除了一般的 Hawk2 要求之外，还需要满足以下先决条件才能使用 Hawk2 监视多个群集：

先决条件

- 要通过 Hawk2 的仪表板监视的所有群集必须运行 SUSE Linux Enterprise High Availability Extension 15 SP2。
- 如果您之前未在每个群集节点上用自己的证书（或官方证书颁发机构签名的证书）替换 Hawk2 的自我签名证书，请执行以下操作：在每个群集的每个节点上至少登录 Hawk2 一次。验证证书（或在浏览器中添加例外以绕过警告）。否则，Hawk2 将无法连接到群集。

过程 7.28：使用仪表板监视多个群集

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 仪表板。

Hawk2 会显示当前群集站点的资源和节点的概述。此外，它还会显示已配置为与 Geo 群集配合使用的所有票据。如需有关此视图中所用图标的信息，请单击图例。要搜索资源 ID，请在搜索文本框中输入名称 (ID)。若只想显示特定节点，请单击过滤器图标并选择一个过滤选项。

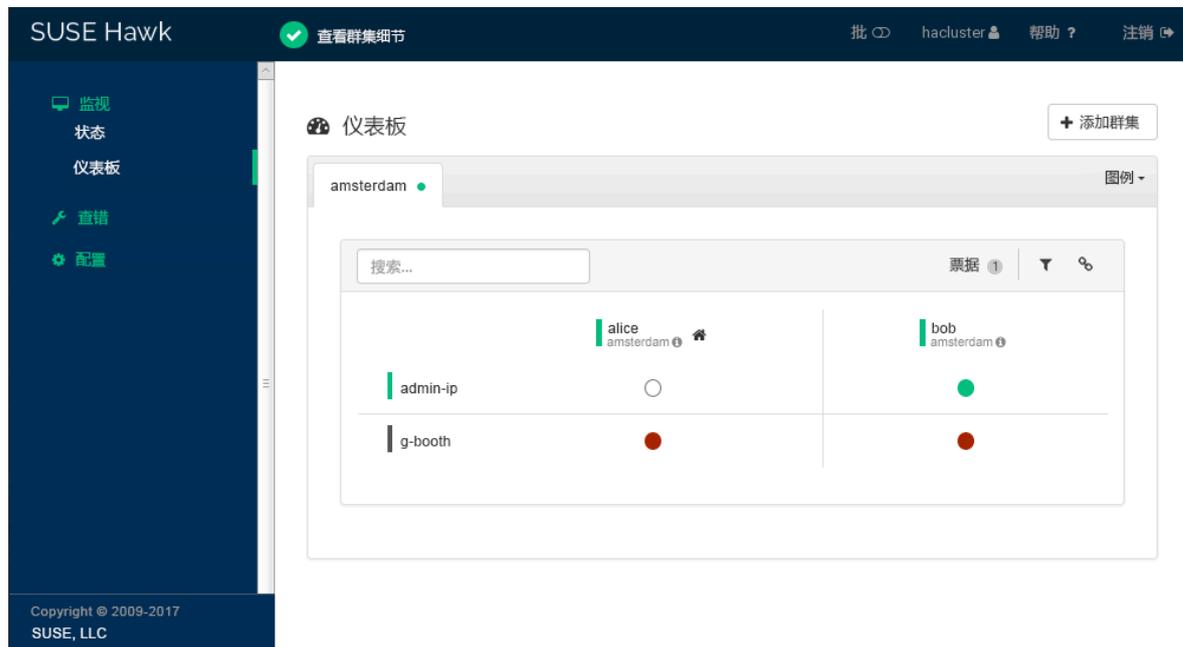
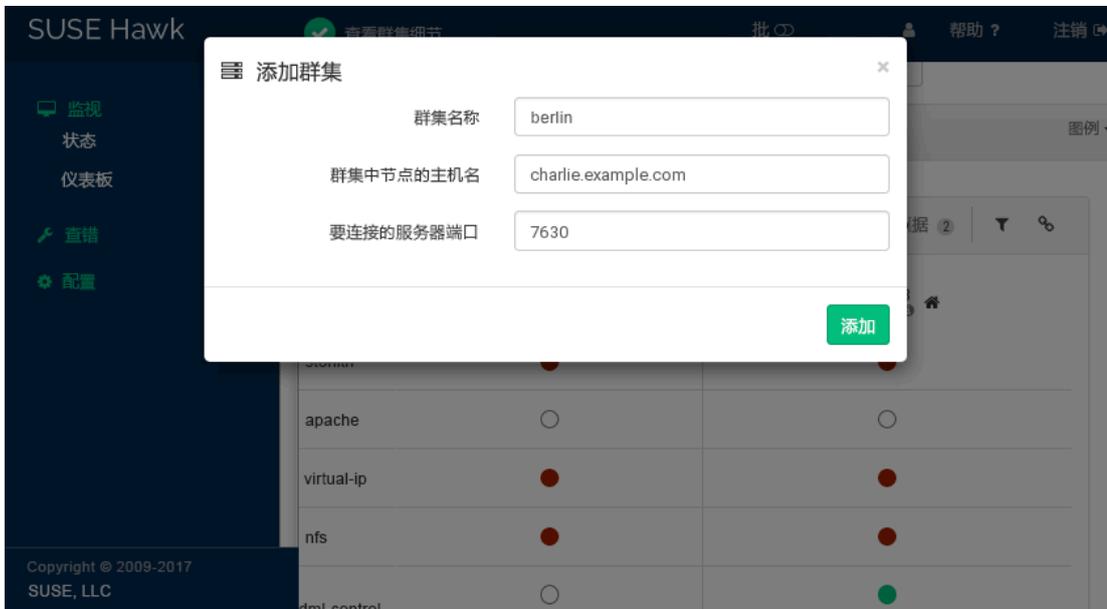


图 7.16：包含一个群集站点 (amsterdam) 的 HAWK2 仪表板

3. 要为多个群集添加仪表板，请执行以下操作：

- a. 单击添加群集。
- b. 输入用于在仪表板中标识该群集的群集名称。例如，berlin。
- c. 输入第二个群集的其中一个节点的完全限定主机名。例如，charlie。



- d. 单击添加。Hawk2 会为新添加的群集站点显示另一个选项卡，其中会列出该群集站点的节点和资源的概述。

注意：连接错误

如果系统提示您输入口令来登录此节点，则表明您可能未连接到该节点且未替换自我签名证书。在此情况下，即使在输入口令之后，连接也将失败，并显示以下讯息：连接服务器时出错。每隔 5 秒钟重试一次...

要继续，请参见[替换自我签名证书](#)。

4. 要查看群集站点的更多细节或管理群集站点，请切换到站点的选项卡并单击锁链图标。Hawk2 会在新的浏览器窗口或选项卡中打开此站点的状态视图。在此视图中，您可以管理 Geo 群集的这部分内容。
5. 要从仪表板中去除某个群集，请单击该群集细节右侧的 x 图标。

7.9 使用批模式

Hawk2 提供批模式，包括群集模拟器。该模式可用于以下操作：

- 对群集进行分阶段更改并通过单次事务应用这些更改，而不是让每项更改立即生效。
- 模拟更改和群集事件，例如，了解可能失败的情况。

例如，在创建相互依赖的资源组时，可以使用批模式。通过使用批模式，您可以避免将中间或不完整的配置应用到群集。

启用批模式后，您可以添加或编辑资源和约束，或更改群集配置。此外，还可以模拟群集中的事件，包括变为联机或脱机的节点、资源操作，以及要授予或撤消的票据。有关详细信息，请参见[过程 7.30 “插入节点、资源或票据事件”](#)。

群集模拟器会在每次更改后自动运行，并在用户界面上显示预期效果。举例而言，这还意味着当您在批模式下停止某资源时，用户界面上会将该资源显示为已停止，但实际上，该资源仍在运行中。

重要：在线系统的向导和更改

某些向导包含除纯群集配置以外的其他操作。在批模式下使用这些向导时，群集配置以外的任何其他更改都将立即应用到在线系统。

因此，需要 `root` 权限的向导无法在批模式下执行。

过程 7.29：使用批模式

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 要激活批模式，请从顶层行选择批。

顶层行下方即会另外显示一栏，指出批模式处于活动状态，且包含指向您可在批模式下执行的操作的链接。

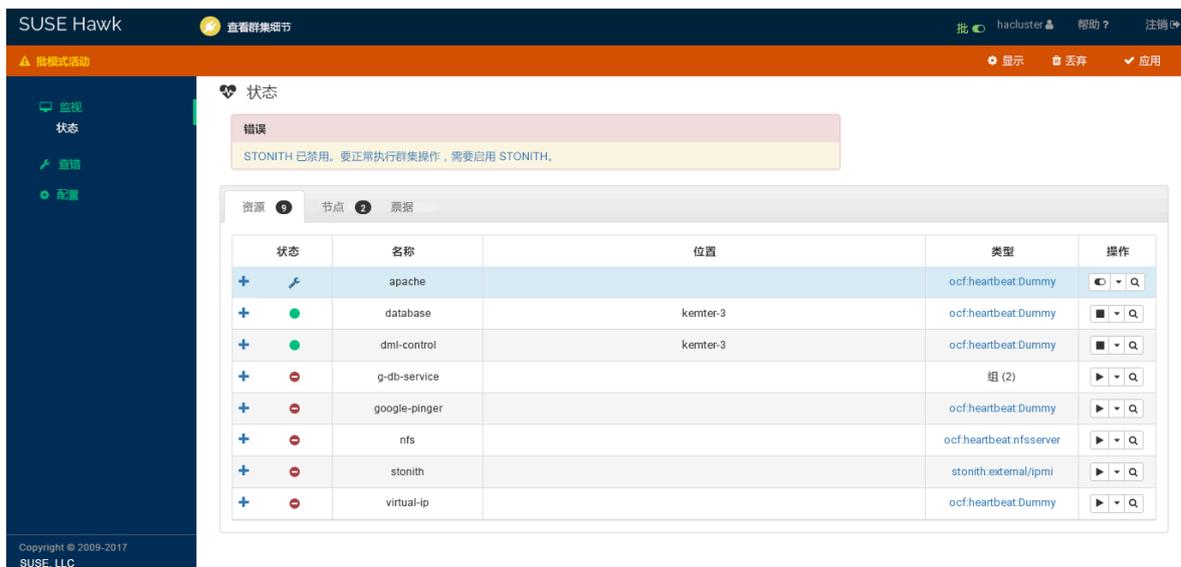


图 7.17：HAWK2 批模式已激活

- 当批模式处于活动状态时，对群集执行任意更改，例如添加或编辑资源和约束，或编辑群集配置。
系统将会模拟更改，并将其显示在所有屏幕上。
- 要查看所做更改的细节，请从批模式栏中选择显示。批模式窗口即会打开。
对于任何配置更改，该模式会以 crmsh 语法显示在线状态与模拟更改之间的差异：以 - 字符开头的行表示当前状态，而以 + 开头的行则显示目标状态。
- 要插入事件或查看更多细节，请参见过程 7.30。否则请关闭窗口。
- 选择丢弃或应用模拟的更改，并确认您的选择。此操作还会停用批模式，使您回到正常模式。

在批模式下运行时，Hawk2 还允许您插入节点事件和资源事件。

节点事件

可让您更改节点的状态。可用的状态有联机、脱机和不干净。

资源事件

可让您更改资源的一些属性。例如，您可以设置操作（如 start、stop、monitor）、其要应用到的节点，以及要模拟的预期结果。

票据事件

可让您测试授予和撤消票据（用于 Geo 群集）的影响。

过程 7.30：插入节点、资源或票据事件

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 如果批模式未启动，请单击顶层行上的批切换到批模式。

3. 在批模式栏中，单击显示打开批模式窗口。

4. 要模拟节点的状态更改:

- a. 单击插入 > 节点事件。
- b. 选择要操作的节点，然后选择其目标状态。
- c. 确认更改。您的事件便会添加到批模式对话框中所列的事件队列中。

5. 模拟资源操作:

- a. 单击插入 > 资源事件。
- b. 选择要操作的资源和要模拟的操作。
- c. 如果必要，请定义间隔。
- d. 选择要运行操作的节点及目标结果。您的事件便会添加到批模式对话框中所列的事件队列中。
- e. 确认更改。

6. 要模拟票据操作，请执行以下操作:

- a. 单击插入 > 票据事件。
- b. 依次选择要操作的票据和要模拟的操作。
- c. 确认更改。您的事件便会添加到批模式对话框中所列的事件队列中。

7. 批模式对话框（图 7.18）会为每个插入的事件显示新的一行。此处列出的所有事件都会立即被模拟并反映到状态屏幕上。

如果您还执行了任何配置更改，在线状态和模拟更改之间的差异会显示在所插入事件的下方。



图 7.18：HAWK2 批模式 - 插入的事件和配置更改

8. 要去除插入的事件，请单击该事件旁边的去除图标。Hawk2 会相应地更新状态屏幕。
9. 要查看模拟运行的更多细节，请单击模拟器并选择以下选项之一：

摘要

显示详细的摘要。

CIB (输入) /CIB (输出)

CIB (输入) 会显示初始的 CIB 状态。CIB (输出) 会显示转换后 CIB 的情况。

转换图

显示转换的图形表示形式。

交付

显示转换的 XML 表示形式。

10. 如果您已审阅模拟的更改，请关闭批模式窗口。
11. 要退出批模式，请应用或丢弃模拟的更改。

7.10 查看群集历史记录

Hawk2 提供了以下用于查看群集上的过去事件（按不同级别和不同详细程度）的功能：

- 第 7.10.1 节 “查看节点或资源的最近事件”
- 第 7.10.2 节 “使用历史记录浏览器获得群集报告”
- 第 7.10.3 节 “在历史记录浏览器中查看转换细节”

7.10.1 查看节点或资源的最近事件

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择监视 > 状态。它会列出资源和节点。

3. 要查看资源的最近事件:

- 单击资源并选择相应的资源。
- 在资源的操作列中，单击向下箭头按钮并选择最近的事件。
Hawk2 会打开一个新窗口，显示最近事件的表视图。

4. 要查看节点的最近事件:

- 单击节点并选择相应的节点。
- 在节点的操作列中，选择最近的事件。
Hawk2 会打开一个新窗口，显示最近事件的表视图。

🔄 最近的事件 : alice

RC	资源	操作	上次更改	状态	调用	执行时间	完成
0	dummy1	dummy1_start_0	2016-10-25 (周二) 18:10:49	已启动	18	20ms	✓
0	dummy1	dummy1_monitor_10000	2016-10-25 (周二) 18:10:49	已启动	19	26ms	✓
0	dummy2	dummy2_stop_0	2016-10-25 (周二) 18:10:08	已停止 (已禁用)	15	23ms	✓
0	dummy2	dummy2_monitor_10000	2016-10-25 (周二) 18:09:51	已停止 (已禁用)	13	19ms	✓

7.10.2 使用历史记录浏览器获得群集报告

从左侧导航栏中，选择查错 > 历史记录，以访问历史记录浏览器。历史记录浏览器可让您创建详细的群集报告并查看转换信息。它提供以下选项：

生成

创建特定时间内的群集报告。Hawk2 会调用 `crm report` 命令来生成报告。

上载

允许您上载直接使用 `crm` 外壳创建的或位于不同群集上的 `crm report` 存档。

生成或上载报告后，它们会显示在报告下方。在报告列表中，您可以显示报告的细节，下载或删除报告。

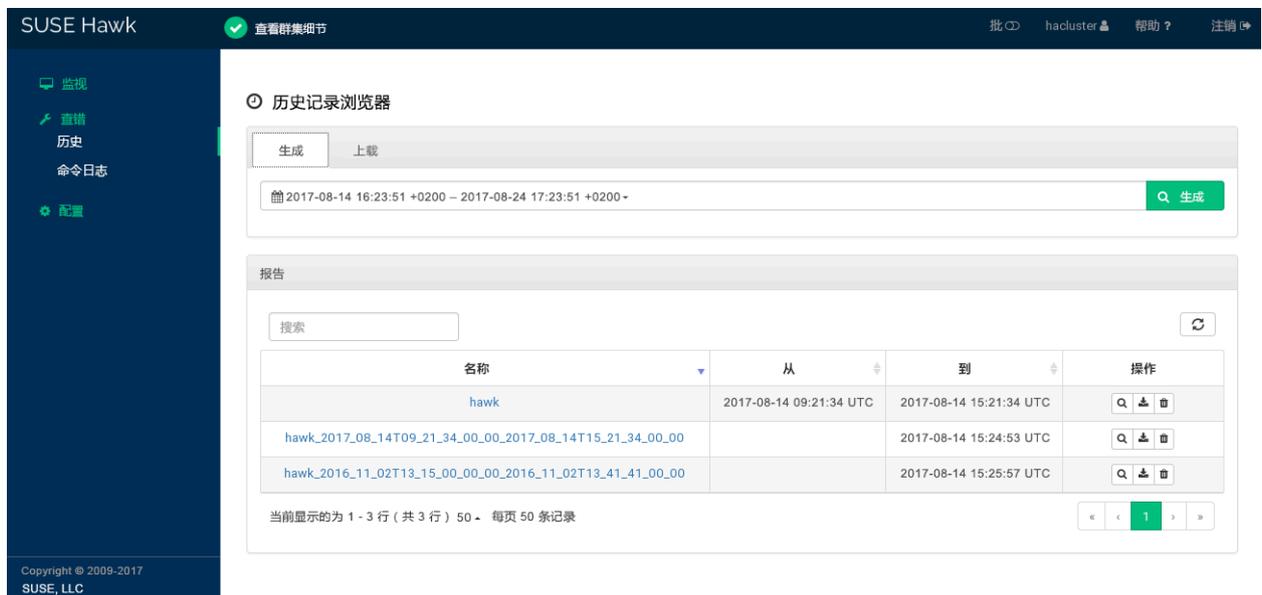


图 7.19：HAWK2 - 历史记录浏览器主视图

过程 7.31：生成或上载群集报告

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择查错 > 历史记录。

历史记录浏览器屏幕会在生成视图中打开。默认情况下，报告的建议时间段为过去 1 小时。

3. 要创建群集报告：

- a. 要立即启动报告，请单击生成。
- b. 要修改报告的时间段，请单击建议时间段的任意位置并从下拉框中选择另一个选项。您还可以分别输入自定义的开始日期、结束日期及小时。要启动报告，请单击生成。
报告生成后会显示在报告下方。

4. 要上传群集报告， crm report 存档必须位于您可通过 Hawk2 访问的文件系统中。按如下所示继续：

- a. 切换到上传选项卡。
- b. 浏览群集报告存档并单击上传。
报告上传后会显示在报告下方。

5. 要下载或删除报告，请在操作列中单击报告旁边的相应图标。

6. 要查看历史记录浏览器中的报告细节，请单击报告的名称，或从操作列中选择显示。

SUSE Hawk 查看群集细节 hacluster 帮助? 注销

历史记录浏览器

名称 hawk_2017_08_14T09_21_34_00_00_2017_08_14T15_21_34_00_00

从 2017-08-14 09:23:19 UTC

到 2017-08-14 15:21:31 UTC

转换 18

节点事件

资源事件

```
2017-08-14T11:23:19.465929+02:00 kenter-2 lrmd[1892]: notice: executing - rsc:nfs action:stop call_id:113
2017-08-14T11:23:19.907638+02:00 kenter-2 nfsserver(nfs)[2441]: INFO: Stopping NFS server ...
2017-08-14T11:23:19.942108+02:00 kenter-2 nfsserver(nfs)[2441]: INFO: Stop: threads
2017-08-14T11:23:19.468195+02:00 kenter-3 crmd[1894]: notice: Initiating stop operation nfs_stop_0 on kenter-2
2017-08-14T11:23:19.959297+02:00 kenter-2 nfsserver(nfs)[2441]: INFO: Stop: rpc-stad
2017-08-14T11:23:19.978677+02:00 kenter-2 nfsserver(nfs)[2441]: INFO: Stop: nfs-idmapd
2017-08-14T11:23:20.076367+02:00 kenter-3 crmd[1894]: notice: Initiating start operation nfs_start_0 locally on kenter-3
2017-08-14T11:23:20.133732+02:00 kenter-3 lrmd[1891]: notice: executing - rsc:nfs action:start call_id:94
2017-08-14T11:23:20.011716+02:00 kenter-2 nfsserver(nfs)[2441]: INFO: stop: nfs-mountd
2017-08-14T11:23:20.649564+02:00 kenter-3 nfsserver(nfs)[16256]: ERROR: rpcbind is not running
```

7. 单击报告按钮返回到报告列表。

历史记录浏览器中的报告细节

- 报告的名称。
- 报告的开始时间。
- 报告的结束时间。
- 报告所涵盖的群集中的转换次数以及所有转换的时间表。要了解如何查看转换的更多细节，请参见第 7.10.3 节。
- 节点事件。
- 资源事件。

7.10.3 在历史记录浏览器中查看转换细节

对于每个转换，群集都会保存其所提供的状态副本，作为对 `pacemaker-schedulerd` 的输入。会记录此存档的路径。所有 `pe-*` 文件都会在指定协调器 (DC) 上生成。由于群集中的 DC 可能会更换，因此可能存在来自多个节点的 `pe-*` 文件。所有 `pe-*` 文件都是保存的 CIB 快照，`pacemaker-schedulerd` 在执行计算时会将其用作输入。

在 Hawk2 中，您可以显示每个 `pe-*` 文件的名称、创建时间以及创建各文件所在的节点。此外，历史记录浏览器可以根据相应的 `pe-*` 文件以可视化形式显示以下细节：

历史记录浏览器中的转换细节

细节

显示属于转换的日志记录数据片段。显示以下命令的输出（包括资源代理的日志讯息）：

```
crm history transition peinput
```

配置

显示创建 `pe-*` 文件时的群集配置。

差别

显示选定 `pe-*` 文件与下一个文件之间的配置和状态差异。

登录

显示属于转换的日志记录数据片段。显示以下命令的输出：

```
crm history transition log peinput
```

这包括来自以下守护程序的细节：[pacemaker-schedulerd](#)、[pacemaker-control](#) 和 [pacemaker-execd](#)。

图形

显示转换的图形表示形式。如果您单击图形，则会模拟计算（与 [pacemaker-schedulerd](#) 执行的计算完全一样），并生成图形可视化表示形式。

过程 7.32：查看转换细节

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择查错 > 历史记录。

如果报告已生成或上载，它们会显示在报告列表中。否则，请按[过程 7.31](#)中所述生成或上载报告。

3. 单击报告的名称或从操作列中选择显示以打开[历史记录浏览器中的报告细节](#)。

4. 要访问转换细节，您需要在下面显示的转换时间表中选择一个转换点。使用上一个和下一个以及放大和缩小图标查找您感兴趣的转换。

5. 要显示 `pe-input*` 文件的名称、创建时间和创建该文件所在的节点，请将鼠标指针悬停在时间表的转换点上。

6. 要查看[历史记录浏览器中的转换细节](#)，请单击要了解其详细信息的转换点。

7. 要显示细节、配置、差异、日志或示意图，请单击相应的按钮以显示[历史记录浏览器中的转换细节](#)中所述的内容。

8. 要返回报告列表，请单击报告按钮。

7.11 校验群集状态

Hawk2 提供了可检查和检测群集存在的问题的向导。分析完成后，Hawk2 会创建包含更多细节的群集报告。要校验群集状态并生成报告，Hawk2 需要具有在节点之间进行无口令 SSH 访问的权限。否则，它只能从当前节点收集数据。如果您已使用 `ha-cluster-bootstrap` 包提供的引导脚本设置群集，则无口令 SSH 访问权限已配置好。如果您需要手动配置，请参见第 D.2 节“配置无口令 SSH 帐户”。

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 从左侧导航栏中，选择配置 > 向导。
3. 展开基本类别。
4. 选择校验状态和配置向导。
5. 单击校验进行确认。
6. 输入群集的 root 口令，然后单击应用。Hawk2 将生成报告。

8 配置和管理群集资源（命令行）

要配置和管理群集资源，可以使用 `crm` 外壳 (`crmsh`) 命令行实用程序或 Hawk2（基于 Web 的用户界面）。

本章介绍了命令行工具 `crm`，并包含此工具的概述以及如何使用模板，主要介绍如何配置和管理群集资源：创建基本和高级类型的资源（组和克隆资源）、配置约束、指定故障转移节点和故障回复节点、配置资源监视以及手动启动、清理、删除和迁移资源。



注意：用户特权

需要足够的特权才能管理群集。`crm` 命令及其子命令都需要以 `root` 用户或 CRM 拥有者用户（通常为 `hacluster` 用户）的身份来运行。

但是，`user` 选项允许您作为普通（非特权）用户运行 `crm` 及其子命令，而且必要时能使用 `sudo` 更改其 ID。例如，在以下命令中，`crm` 将使用 `hacluster` 作为特权用户 ID：

```
root # crm options user hacluster
```

请注意，您需要将 `/etc/sudoers` 设置为 `sudo` 不要求提供密码。

8.1 crmsh - 概述

`crm` 命令有多个子命令，这些子命令用于管理资源、CIB、节点和资源代理等。它提供了全面的帮助系统，并嵌入了示例。所有示例都遵循附录 B 中所述的命名约定。



提示：交互式 `crm` 提示符

使用不带自变量（或只带一个 `sublevel` 自变量）的 `crm`，`crm` 外壳将进入交互式模式。此模式由以下提示符指示：

```
crm(live/HOSTNAME)
```

为了容易阅读，我们的文档在交互式 `crm` 提示符中省略了主机名。仅当您需要在特定的节点（如 `alice`）上运行交互式外壳时，才包含主机名，例如：

```
crm(live/alice)
```

8.1.1 获得帮助

可通过以下方式之一访问帮助：

- 输出 `crm` 及其命令行选项的用法：

```
root # crm --help
```

- 列出所有可用的命令：

```
root # crm help
```

- 访问其他帮助部分，而不只是命令参考：

```
root # crm help topics
```

- 查看 `configure` 子命令的完整帮助文本：

```
root # crm configure help
```

- 要列显 `configure` 的 `group` 子命令的语法、用法及示例：

```
root # crm configure help group
```

以下命令的作用相同：

```
root # crm help configure group
```

几乎所有 `help` 子命令（请不要与 `--help` 选项混淆）的输出都会打开文本编辑器。此文本编辑器允许您向上/向下滚动，以便更加方便地阅读帮助文本。要退出文本编辑器，请按 **Q** 键。



提示：在 Bash 和交互式外壳中使用 Tab 键补全

crmsh 不仅为交互式外壳提供 Tab 键补全，还全面支持在 Bash 中直接使用此功能。例如，键入 `crm help config -l` 会补全文字（就像在交互式外壳中一样）。

8.1.2 执行 crmsh 的子命令

`crm` 命令本身可按以下方式使用：

- **直接：** 将所有子命令连接到 `crm` 中，按 `Enter`，您将立即看到输出。例如，输入 `crm help ra` 可获取有关 `ra` 子命令（资源代理）的信息。

可以缩写子命令，只要缩写后的子命令是唯一的即可。例如，可以将 `status` 缩写为 `st`，crmsh 可以识别该缩写。

另一项功能是缩写参数。通常，您是通过 `params` 关键字添加参数的。如果 `params` 部分是第一个且是唯一存在的部分，则您可以省略它。例如，下面一行：

```
root # crm primitive ipaddr ocf:heartbeat:IPaddr2 params ip=192.168.0.55
```

相当于下行：

```
root # crm primitive ipaddr ocf:heartbeat:IPaddr2 ip=192.168.0.55
```

- **作为 crm 外壳脚本：** Crm 外壳脚本包含 `crm` 的子命令。有关详细信息，请参见第 8.1.4 节“使用 crmsh 的外壳脚本”。
- **作为 crmsh 群集脚本：** 此类脚本是元数据、对 RPM 包的参照、配置文件及多个 crmsh 子命令捆绑在一起并以单个描述性名称命名的集合。可以通过 `crm script` 命令管理这些内容。
请不要将它们与 crmsh 外壳脚本相混淆：尽管两者具有一些共同的目标，但 crm 外壳脚本只包含子命令，而群集脚本所包含的远远不只是简单的命令枚举。有关详细信息，请参见第 8.1.5 节“使用 crmsh 的群集脚本”。
- **作为内部外壳交互：** 输入 `crm` 以进入内壳。提示更改为 `crm(live)`。使用 `help` 可获取可用子命令的概述。由于内壳具有不同级别的子命令，您可以键入一个子命令然后按 `Enter` “进入”相应的级别。

例如，如果输入 `resource`，则进入资源管理级别。提示符将更改为

`crm(live)resource#`。要退出内壳，可使用命令 `quit`、`bye` 或 `exit`。如果需要返回上一个级别，可使用 `back`、`up`、`end` 或 `cd`。

您可以键入 `crm` 和相应的子命令（不含任何选项）直接输入级别，然后按 `Enter`。

内壳还支持使用 `Tab` 键完成子命令和资源。输入命令的开头，按 `-|` 和 `crm` 完成相应对象。

除了前面说明的方法外，`crmsh` 还支持执行同步命令。使用 `-w` 选项可以激活该命令。如果已启动不带 `-w` 选项的 `crm`，则可以稍后通过将用户自选设置的 `wait` 设为 `yes` (`options wait yes`) 来启用它。如果此选项已启用，则 `crm` 将会等到事务完成为止。事务一经启用，就会打印出点以指示进度。同步命令执行仅适用于 `resource start` 之类的命令。



注意：管理和配置子命令之间的区别

`crm` 工具有管理功能（子命令 `resource` 和 `node`），可用于配置（`cib` 和 `configure`）。

以下小节概述了 `crm` 工具的一些重要方面。

8.1.3 显示有关 OCF 资源代理的信息

由于在群集配置中一直需要处理资源代理，`crm` 工具包含了 `ra` 命令。使用该命令可以显示有关资源代理的信息并对其进行管理（如需其他信息，另请参见第 6.3.2 节“支持的资源代理类”）：

```
root # crm ra
crm(live)ra#
```

命令 `classes` 可列出所有类和提供程序：

```
crm(live)ra# classes
lsb
ocf / heartbeat linbit lvm2 ocfs2 pacemaker
service
stonith
```

```
systemd
```

要获取某个类（和提供程序）的所有可用资源的概述，可使用 **list** 命令：

```
crm(live)ra# list ocf
AoEtarget          AudibleAlarm      CTDB              ClusterMon
Delay              Dummy             EvmsSCC           Evmsd
Filesystem         HealthCPU         HealthSMART       ICP
IPaddr             IPaddr2           IPsrcaddr         IPv6addr
LVM                LinuxSCSI         MailTo            ManageRAID
ManageVE           Pure-FTPd         Raid1             Route
SAPDatabase        SAPInstance       SendArp           ServerAID
...
```

可使用 **info** 查看资源代理的概述：

```
crm(live)ra# info ocf:linbit:drbd
This resource agent manages a DRBD* resource
as a master/slave resource. DRBD is a shared-nothing replicated storage
device. (ocf:linbit:drbd)

Master/Slave OCF Resource Agent for DRBD

Parameters (* denotes required, [] the default):

drbd_resource* (string): drbd resource name
    The name of the drbd resource from the drbd.conf file.

drbdconf (string, [/etc/drbd.conf]): Path to drbd.conf
    Full path to the drbd.conf file.

Operations' defaults (advisory minimum):

start          timeout=240
promote        timeout=90
demote         timeout=90
notify         timeout=90
stop           timeout=100
monitor_Slave_0 interval=20 timeout=20 start-delay=1m
```

```
monitor_Master_0 interval=10 timeout=20 start-delay=1m
```

按 **Q** 退出查看器。

提示：直接使用 **crm**

在之前的示例中，我们使用了 **crm** 命令的内壳。但是您不一定非要使用它。将相应子命令添加到 **crm** 中也可获得相同的结果。例如，在外壳中输入 **crm ra list ocf** 可以列出所有 OCF 资源代理。

8.1.4 使用 **crmsh** 的外壳脚本

Crmsh 外壳脚本提供了将 **crmsh** 子命令枚举到文件中的便捷方式。如此，您便可轻松地注释特定行或稍后重新运行这些行。请注意，**crmsh** 外壳脚本只能包含 **crmsh** 子命令，不允许包含任何其他命令。

您需要先创建包含特定命令的文件，然后才能使用 **crmsh** 外壳脚本。例如，下面的文件会列显群集的状态并提供所有节点的列表：

例 8.1：简单 **CRMSH** 外壳脚本

```
# A small example file with some crm subcommands
status
node list
```

以井字符号 (**#**) 开头的所有行都是注释，可忽略。如果行过长，可在结尾处插入反斜杠 (****)，然后在下一行继续。建议缩进属于特定子命令的行以便于阅读。

要使用此脚本，请使用以下其中一种方法：

```
root # crm -f example.cli
root # crm < example.cli
```

8.1.5 使用 **crmsh** 的群集脚本

从所有群集节点收集信息并部署任何更改是一项关键的群集管理任务。您不必在不同的节点上手动执行相同的过程（这很容易出错），可以使用 **crmsh** 群集脚本来代替该过程。

请不要将它们与 crmsh 外壳脚本相混淆，第 8.1.4 节 “使用 crmsh 的外壳脚本” 中对后者进行了介绍。

对比 crmsh 外壳脚本，群集脚本另外会执行如下任务：

- 安装特定任务所需的软件。
- 创建或修改任何配置文件。
- 收集信息并报告群集的潜在问题。
- 将更改部署到所有节点。

crmsh 群集脚本并不能取代其他群集管理工具，它只是提供了一种集成的方式用于在群集中执行上述任务。有关详细信息，请参见<http://crmsh.github.io/scripts/>。

8.1.5.1 使用

要获取所有可用群集脚本的列表，请运行：

```
root # crm script list
```

要查看脚本的组成部分，请使用 `show` 命令和群集脚本的名称，例如：

```
root # crm script show mailto
mailto (Basic)
MailTo

This is a resource agent for MailTo. It sends email to a sysadmin
whenever a takeover occurs.

1. Notifies recipients by email in the event of resource takeover

id (required) (unique)
    Identifier for the cluster resource
email (required)
    Email address
subject
    Subject
```

`show` 的输出包含标题、简要说明和过程。每个过程分为一系列按给定顺序执行的步骤。

每个步骤都包含一份必要参数与可选参数及其简要说明和默认值的列表。

每个群集脚本都可识别一组通用参数。这些参数可传递给任何脚本：

表 8.1：通用参数

参数	自变量	描述
<u>action</u>	<u>INDEX</u>	如果设置此参数，则只会执行单个操作（verify 会返回索引）
<u>dry_run</u>	<u>BOOL</u>	如果设置此参数，则只会模拟执行（默认值：no）
<u>nodes</u>	<u>LIST</u>	列出要对其执行脚本的节点
<u>port</u>	<u>NUMBER</u>	要连接的端口
<u>statefile</u>	<u>FILE</u>	在以单一步进方式执行时，状态将保存在给定文件中
<u>sudo</u>	<u>BOOL</u>	如果设置此参数，crm 将在适当的情况下提示输入 sudo 口令并使用 sudo（默认值：no）
<u>timeout</u>	<u>NUMBER</u>	以秒为单位的执行超时（默认值：600）
<u>user</u>	<u>USER</u>	以给定用户的身份运行脚本

8.1.5.2 校验和运行群集脚本

在运行某个群集脚本之前，请检查该脚本将要执行的操作并校验其参数，以免出现问题。群集脚本可能会执行一系列操作，并且可能会出于各种原因而失败。因此，在运行脚本之前校验参数有助于避免出现问题。

例如，mailto 资源代理需要唯一的标识符和一个电子邮件地址。要校验这些参数，请运行：

```
root # crm script verify mailto id=sysadmin email=tux@example.org
1. Ensure mail package is installed

    mailx

2. Configure cluster resources

    primitive sysadmin ocf:heartbeat:MailTo
        email="tux@example.org"
        op start timeout="10"
        op stop timeout="10"
        op monitor interval="10" timeout="10"

    clone c-sysadmin sysadmin
```

verify 将会列显步骤，并将所有占位符替换为您的给定参数。如果 **verify** 发现任何问题，将会报告问题。如果一切正常，请将 **verify** 命令替换为 **run**：

```
root # crm script run mailto id=sysadmin email=tux@example.org
INFO: MailTo
INFO: Nodes: alice, bob
OK: Ensure mail package is installed
OK: Configure cluster resources
```

使用 **crm status** 检查您的资源是否已集成到群集中：

```
root # crm status
[...]
Clone Set: c-sysadmin [sysadmin]
    Started: [ alice bob ]
```

8.1.6 使用配置模板



注意：弃用通告

配置模板已弃用，将来会被去除。配置模板将由群集脚本取代，具体请参见第 8.1.5 节“使用 crmsh 的群集脚本”。

配置模板可为 crmsh 提供即时可用的群集配置。请不要将其与资源模板（如第 8.3.3 节“创建资源模板”中所述）混淆。资源模板只适用于群集，而不适用于 crm 外壳。

配置模板只需稍作更改，即可满足特定用户的需要。每次使用模板创建配置时，都会出现警告消息，提示您哪些可以稍后编辑以供将来自定义。

以下步骤显示了如何创建简单有效的 Apache 配置：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 从配置模板创建一个新配置：

- a. 切换到 `template` 子命令：

```
crm(live)configure# template
```

- b. 列出可用的配置模板：

```
crm(live)configure template# list templates
gfs2-base  filesystem  virtual-ip  apache    clvm      ocfs2     gfs2
```

- c. 确定需要的配置模板。由于我们需要 Apache 配置，因此选择了 `apache` 模板并将其命名为 `g-intranet`：

```
crm(live)configure template# new g-intranet apache
INFO: pulling in template apache
INFO: pulling in template virtual-ip
```

3. 定义参数：

- a. 列出您创建的配置：

```
crm(live)configure template# list
g-intranet
```

- b. 显示需要由您填充的最少的必要更改：

```
crm(live)configure template# show
```

```
ERROR: 23: required parameter ip not set
ERROR: 61: required parameter id not set
ERROR: 65: required parameter configfile not set
```

- c. 调用首选的文本编辑器，填写显示为错误（如步骤 3.b 中所示）的所有行：

```
crm(live)configure template# edit
```

4. 显示配置并检查配置是否有效（粗体文本取决于您在步骤 3.c 中进入的配置）：

```
crm(live)configure template# show
primitive virtual-ip ocf:heartbeat:IPaddr \
  params ip="192.168.1.101"
primitive apache ocf:heartbeat:apache \
  params configfile="/etc/apache2/httpd.conf"
  monitor apache 120s:60s
group g-intranet \
  apache virtual-ip
```

5. 应用配置：

```
crm(live)configure template# apply
crm(live)configure# cd ..
crm(live)configure# show
```

6. 将更改提交到 CIB：

```
crm(live)configure# commit
```

如果知道细节，可以更加简化命令。上述过程可汇总为外壳上的以下命令：

```
root # crm configure template \
  new g-intranet apache params \
  configfile="/etc/apache2/httpd.conf" ip="192.168.1.101"
```

如果在 **crm** 内壳中，可使用以下命令：

```
crm(live)configure template# new intranet apache params \
  configfile="/etc/apache2/httpd.conf" ip="192.168.1.101"
```

但是，前一条命令仅会从配置模板创建其配置。它不会将其应用或提交到 CIB。

8.1.7 使用阴影配置进行测试

阴影配置可用于测试不同的配置方案。如果创建了多个阴影配置，则可逐一测试这些配置，以查看更改的影响。

一般的流程显示如下：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 创建新的阴影配置：

```
crm(live)configure# cib new myNewConfig  
INFO: myNewConfig shadow CIB created
```

如果省略阴影 CIB 的名称，则会创建临时名称 `@tmp@`。

3. 要将当前的活动配置复制到阴影配置中，可使用以下命令，否则请跳过此步骤：

```
crm(myNewConfig)# cib reset myNewConfig
```

使用上面的命令便于稍后修改现有资源。

4. 照常进行更改。创建阴影配置后，会应用所有更改。要保存所有更改，请使用以下命令：

```
crm(myNewConfig)# commit
```

5. 如果再次需要活动群集配置，可使用以下命令切换回此配置：

```
crm(myNewConfig)configure# cib use live  
crm(live)#
```

8.1.8 调试配置更改

将配置更改装载回群集之前，建议使用 `ptest` 复查更改。使用 `ptest` 命令可显示提交更改后产生的操作图。您需要 `graphviz` 包才能显示该图形。以下示例是一个抄本，添加了监视操作：

```

root # crm configure
crm(live)configure# show fence-bob
primitive fence-bob stonith:apcsmart \
    params hostlist="bob"
crm(live)configure# monitor fence-bob 120m:60s
crm(live)configure# show changed
primitive fence-bob stonith:apcsmart \
    params hostlist="bob" \
    op monitor interval="120m" timeout="60s"
crm(live)configure# ptest
crm(live)configure# commit

```

8.1.9 群集图表

要输出群集图表，请使用命令 `crm configure graph`。它会在当前的窗口上显示当前配置，因此需要配备 X11。

如果您希望使用可缩放矢量图 (SVG)，请使用以下命令：

```

root # crm configure graph dot config.svg svg

```

8.2 管理 Corosync 配置

Corosync 是大多数 HA 群集的基础讯息交换层。`corosync` 子命令提供了用于编辑和管理 Corosync 配置的命令。

例如，要列出群集的状态，请使用 `status`：

```

root # crm corosync status
Printing ring status.
Local node ID 175704363
RING ID 0
    id      = 10.121.9.43
    status  = ring 0 active with no faults
Quorum information
-----

```

```
Date: Thu May 8 16:41:56 2014
Quorum provider: corosync_votequorum
Nodes: 2
Node ID: 175704363
Ring ID: 4032
Quorate: Yes
```

Votequorum information

```
-----
Expected votes: 2
Highest expected: 2
Total votes: 2
Quorum: 2
Flags: Quorate
```

Membership information

```
-----
Nodeid      Votes Name
175704363   1 alice.example.com (local)
175704619   1 bob.example.com
```

diff 命令非常有用：它可以比较所有节点上的 Corosync 配置（如果未另行指定）并打印出各自的差异：

```
root # crm corosync diff
--- bob
+++ alice
@@ -46,2 +46,2 @@
-     expected_votes: 2
-     two_node: 1
+     expected_votes: 1
+     two_node: 0
```

有关细节，请参见 http://crmsh.nongnu.org/crm.8.html#cmdhelp_corosync。

8.3 配置群集资源

作为群集管理员，您需要在群集中为服务器上运行的每个资源或应用程序创建群集资源。群集资源可以包括网站、电子邮件服务器、数据库、文件系统、虚拟机和任何其他基于服务器的应用程序或在任意时间对用户都可用的服务。

有关可创建的资源类型的概述，请参见第 6.3.3 节“资源类型”。

8.3.1 从文件装载群集资源

可从本地文件或网络 URL 装载部分或全部配置。可定义三种不同方法：

replace

此选项会将当前配置替换为新的源配置。

update

此选项会尝试导入源配置。它会向当前配置添加新项目或更新现有项目。

push

此选项会将内容从来源导入到当前配置中（与 update 相同）。不过，它会去除在新配置中不可用的对象。

要从文件 mycluster-config.txt 装载新配置，请使用以下语法：

```
root # crm configure load push mycluster-config.txt
```

8.3.2 创建群集资源

有三种 RA（资源代理）类型可用于群集（有关背景信息，请参见第 6.3.2 节“支持的资源代理类”）。要将新资源添加到群集，请按如下操作：

1. 以 root 用户身份登录，然后启动 crm 工具：

```
root # crm configure
```

2. 配置原始 IP 地址：

```
crm(live)configure# primitive myIP ocf:heartbeat:IPaddr \  
    params ip=127.0.0.99 op monitor interval=60s
```

上一命令配置了名称为 `myIP` 的“原始资源”。需要选择一个类（此处为 `ocf`）、提供程序（`heartbeat`）和类型（`IPaddr`）。此外，此原始资源还需要其他参数，如 IP 地址。根据设置更改地址。

3. 显示您所做的更改并进行复查：

```
crm(live)configure# show
```

4. 提交更改使其生效：

```
crm(live)configure# commit
```

8.3.3 创建资源模板

如果希望使用类似的配置创建多个资源，则资源模板可以简化此项任务。有关一些基本背景信息，另请参见第 6.5.3 节“资源模板和约束”。不要将它们与第 8.1.6 节“使用配置模板”中的“常规”模板相混淆。使用 `rsc_template` 命令可以熟悉其语法：

```
root # crm configure rsc_template  
usage: rsc_template <name> [<class>:[<provider>:]]<type>  
    [params <param>=<value> [<param>=<value>...]]  
    [meta <attribute>=<value> [<attribute>=<value>...]]  
    [utilization <attribute>=<value> [<attribute>=<value>...]]  
    [operations id_spec  
        [op op_type [<attribute>=<value>...] ...]]
```

例如，以下命令将会根据 `ocf:heartbeat:Xen` 资源和一些默认值及操作新建一个名称为 `BigVM` 的资源模板：

```
crm(live)configure# rsc_template BigVM ocf:heartbeat:Xen \  
    params allow_mem_management="true" \  
    op monitor timeout=60s interval=15s \  
    op stop timeout=10m \  
    op start timeout=10m
```

```
op start timeout=10m
```

定义了新的资源模板后，可以将其用作原始资源或在顺序、共置或 `rscticket` 约束中引用。要引用资源模板，请使用 `@` 符号：

```
crm(live)configure# primitive MyVM1 @BigVM \  
  params xfile="/etc/xen/shared-vm/MyVM1" name="MyVM1"
```

新的原始资源 `MyVM1` 将继承 `BigVM` 资源模板中的所有配置。例如，上述两者的等效配置有：

```
crm(live)configure# primitive MyVM1 ocf:heartbeat:Xen \  
  params xfile="/etc/xen/shared-vm/MyVM1" name="MyVM1" \  
  params allow_mem_management="true" \  
  op monitor timeout=60s interval=15s \  
  op stop timeout=10m \  
  op start timeout=10m
```

如果希望重写一些选项或操作，只需将它们添加到您的（原始）定义中。例如，下面这个新的原始资源 `MyVM2` 会让监视操作的超时增加一倍，而其他值保持不变：

```
crm(live)configure# primitive MyVM2 @BigVM \  
  params xfile="/etc/xen/shared-vm/MyVM2" name="MyVM2" \  
  op monitor timeout=120s interval=30s
```

资源模板可以在约束中引用，以表示所有原始资源都派生自该模板。这有助于生成更加清晰明了的群集配置。除了位置约束外，允许在所有约束中进行资源模板引用。共置约束不能包含多次模板引用。

8.3.4 创建 STONITH 资源

就 `crm` 而言，STONITH 设备只是另一种资源。要创建 STONITH 资源，请执行以下操作：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 使用以下命令获取所有 STONITH 类型的列表：

```
crm(live)# ra list stonith
```

apcmaster	apcmastersnmp	apcsmart
baytech	bladehpi	cyclades
drac3	external/drac5	external/dracmc-
telnet		
external/hetzner	external/hmchttp	external/ibmrsa
external/ibmrsa-telnet	external/ipmi	external/ippower9258
external/kdumpcheck	external/libvirt	external/nut
external/rackpdu	external/riloe	external/sbd
external/vcenter	external/vmware	external/xen0
external/xen0-ha	fence_legacy	ibmhmc
ipmilan	meatware	nw_rpc100s
rcd_serial	rps10	suicide
wti_mpc	wti_nps	

3. 从以上列表中选择 STONITH 类型并查看可用的选项列表。使用以下命令：

```
crm(live)# ra info stonith:external/ipmi
IPMI STONITH external device (stonith:external/ipmi)

ipmitool based power management. Apparently, the power off
method of ipmitool is intercepted by ACPI which then makes
a regular shutdown. If case of a split brain on a two-node
it may happen that no node survives. For two-node clusters
use only the reset method.

Parameters (* denotes required, [] the default):

hostname (string): Hostname
    The name of the host to be managed by this STONITH device.
...
```

4. 使用 `stonith` 类（您在步骤 3 中选择的类型）和相应的参数（如果需要）创建 STONITH 资源，例如：

```
crm(live)# configure
crm(live)configure# primitive my-stonith stonith:external/ipmi \
    params hostname="alice" \
    ipaddr="192.168.1.221" \
    userid="admin" passwd="secret" \
```

```
op monitor interval=60m timeout=120s
```

8.3.5 配置资源约束

配置所有资源只是任务的一部分。即使群集了解所有需要的资源，它仍然不能正确处理它们。例如，尽量不要在 DRBD 的从属节点上装入文件系统（事实上，这将导致 DRBD 出现故障）。定义约束以使这些信息可用于群集。

有关约束的更多信息，请参见第 6.5 节“资源约束”。

8.3.5.1 位置约束

location 命令定义资源可以、不可以或首选在哪些节点上运行。

每个资源可多次添加此类约束。对于给定资源，将评估所有 **location** 约束。下面是个简单的示例，它将首选在名为 alice 的节点上运行资源 fs1 的值设置为 100：

```
crm(live)configure# location loc-fs1 fs1 100: alice
```

另一个示例是使用 pingd 的位置：

```
crm(live)configure# primitive pingd pingd \  
    params name=pingd dampen=5s multiplier=100 host_list="r1 r2"  
crm(live)configure# location loc-node_pref internal_www \  
    rule 50: #uname eq alice \  
    rule pingd: defined pingd
```

位置约束的另一个用例是将基元资源分组为资源集。例如，如果多个资源依赖于 ping 属性来进行网络连接，则此功能会十分有用。以前，需要在配置中复制 -inf/ping 规则数次，因此不必要地增加了复杂性。

以下示例将创建资源集 loc-alice，该资源集引用虚拟 IP 地址 vip1 和 vip2：

```
crm(live)configure# primitive vip1 ocf:heartbeat:IPaddr2 params ip=192.168.1.5  
crm(live)configure# primitive vip2 ocf:heartbeat:IPaddr2 params ip=192.168.1.6  
crm(live)configure# location loc-alice { vip1 vip2 } Mandatory: alice
```

在某些情况下，为 **location** 命令使用资源模式会有效且方便得多。资源模式是用两个斜杠括起的正则表达式。例如，可以使用以下命令全部匹配上述虚拟 IP 地址：

```
crm(live)configure# location loc-alice /vip.*/ Mandatory: alice
```

8.3.5.2 共置约束

colocation 命令用于定义哪些资源应在相同主机上运行，哪些资源应在不同主机上运行。

只能设置 +inf 或 -inf 的分数，定义必须始终或不得在相同节点上运行的资源。还可以使用有限分数。在这种情况下，共置将称为建议，群集可决定不遵循它们，从而在出现冲突时不停止其他资源。

例如，要始终在同一个主机上运行 ID 为 `filesystem_resource` 和 `nfs_group` 的两个资源，可使用以下约束：

```
crm(live)configure# colocation nfs_on_filesystem Mandatory: nfs_group  
filesystem_resource
```

对于主从属配置，除在本地运行资源以外，还有必要了解当前节点是否为主节点。

8.3.5.3 共置没有依赖性的资源集

有时，将一组资源放置在同一个节点上（定义共置约束）会很有用，但前提是这些资源之间不存在硬依赖性。

如果您想要将多个资源放置在同一个节点上，但不想疲于应对每一个资源的故障，那么，您可以使用命令 **weak-bond**。

```
root # crm configure assist weak-bond RES1 RES2
```

weak-bond 的实施将使用给定的资源自动创建虚设资源和共置约束。

8.3.5.4 顺序约束

order 命令定义操作顺序。

有时必需提供资源操作顺序。例如，在设备可用于系统之前，您不能装入文件系统。使用顺序约束可在另一个资源满足某个特殊条件之前或之后启动或停止某项服务，如已启动、已停止或已升级到主资源。

在 `crm` 外壳中使用以下命令配置顺序约束：

```
crm(live)configure# order nfs_after_filesystem mandatory: filesystem_resource
nfs_group
```

8.3.5.5 示例配置约束

本节中使用的示例必须与其他约束结合使用。其中最基本的就是让所有资源与 DRBD 资源的主资源在同一台计算机上运行。在启动其他资源前，DRBD 资源必须是主资源。在 DRBD 设备不是主资源时尝试装入 DRBD 只会失败。必须实现以下约束：

- 文件系统必须始终与 DRBD 资源的主资源位于同一节点上。

```
crm(live)configure# colocation filesystem_on_master Mandatory: \
filesystem_resource drbd_resource:Master
```

- NFS 服务器及 IP 地址必须与文件系统位于相同的节点上。

```
crm(live)configure# colocation nfs_with_fs Mandatory: \
nfs_group filesystem_resource
```

- NFS 服务器及 IP 地址在装入文件系统后启动：

```
crm(live)configure# order nfs_second mandatory: \
filesystem_resource:start nfs_group
```

- 必须在 DRBD 资源提升为节点上的主资源后才能在此节点上装入文件系统。

```
crm(live)configure# order drbd_first Mandatory: \
drbd_resource:promote filesystem_resource:start
```

8.3.6 指定资源故障转移节点

要确定资源故障转移，可使用元属性 `migration-threshold`。如果所有节点上的故障计数超过 `migration-threshold`，资源将处于停止状态。例如：

```
crm(live)configure# location rsc1-alice rsc1 100: alice
```

通常，`rsc1` 首选在 `alice` 上运行。如果失败，将检查 `migration-threshold` 并与将它与故障计数进行比较。如果故障计数 \geq `migration-threshold`，则会将该资源迁移到具有下一个最佳自选设置的节点。

根据 `start-failure-is-fatal` 选项，启动失败会将失败计数设置为 `inf`。停止故障可导致屏蔽。如果未定义 `STONITH`，将不会迁移资源。

有关概述，请参见第 6.5.4 节“故障转移节点”。

8.3.7 指定资源故障回复节点（资源粘性）

当原始节点恢复联机并位于群集中时，资源可能会故障回复到该节点。为防止资源故障回复到之前运行它的节点，或者要指定让该资源故障回复到其他节点，请更改其资源粘性值。可以在创建资源时或之后指定资源粘性。

有关概述，请参见第 6.5.5 节“故障回复节点”。

8.3.8 根据负载影响配置资源放置

某些资源可能具有特定的容量要求，如最低内存量。如果无法满足要求，资源可能无法完全启动或运行时性能下降。

要考虑此情况，可使用 `High Availability Extension` 指定以下参数：

1. 特定节点提供的容量。
2. 特定资源需要的容量。
3. 资源放置整体策略。

有关参数的详细背景信息和配置示例，请参见第 6.5.6 节“根据资源负载影响放置资源”。

要配置资源要求和节点提供的容量，请使用利用率属性。可根据个人喜好命名利用率属性，并根据配置需要定义多个名称/值对。在某些情况下，某些代理（例如 `VirtualDomain`）将自行更新利用率。

在下例中，我们假定您已有群集节点和资源的基本配置，现在想要配置特定节点提供的容量以及特定资源需要的容量。

过程 8.1：使用 `crm` 添加或修改利用率属性

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 要指定节点提供的容量，请使用以下命令并将占位符 `NODE_1` 替换为节点名称：

```
crm(live)configure# node NODE_1 utilization memory=16384 cpu=8
```

上例中的这些值将假定 `NODE_1` 向资源提供 16 GB 内存和 8 个 CPU 核心。

3. 要指定资源需要的容量，请使用：

```
crm(live)configure# primitive xen1 ocf:heartbeat:Xen ... \  
utilization memory=4096 cpu=4
```

这会使资源消耗 `NODE_1` 的 4096 个内存单元以及 4 个 CPU 单元。

4. 使用 `property` 命令配置放置策略：

```
crm(live)configure# property ...
```

可用值如下：

`default` (默认值)

不考虑利用率值。根据位置得分分配资源。如果分数相等，资源将均匀分布在节点中。

`utilization`

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。但仍会根据分配给节点的资源数执行负载平衡。

`minimal`

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。尝试将资源集中到尽可能少的节点上（以节省其余节点上的能耗）。

balanced

在确定节点是否有足够的可用容量来满足资源要求时考虑利用率值。尝试均匀分布资源，从而优化资源性能。

注意：配置资源优先级

可用的放置策略是最佳方法 - 它们不使用复杂的启发式解析程序即可始终实现最佳分配结果。确保正确设置资源优先级，以便首选调度最重要的资源。

5. 退出 crmsh 之前提交更改：

```
crm(live)configure# commit
```

以下示例演示了配有四台虚拟机、节点数相等的三节点群集：

```
crm(live)configure# node alice utilization memory="4000"  
crm(live)configure# node bob utilization memory="4000"  
crm(live)configure# node charlie utilization memory="4000"  
crm(live)configure# primitive xenA ocf:heartbeat:Xen \  
utilization hv_memory="3500" meta priority="10" \  
params xmfile="/etc/xen/shared-vm/vm1"  
crm(live)configure# primitive xenB ocf:heartbeat:Xen \  
utilization hv_memory="2000" meta priority="1" \  
params xmfile="/etc/xen/shared-vm/vm2"  
crm(live)configure# primitive xenC ocf:heartbeat:Xen \  
utilization hv_memory="2000" meta priority="1" \  
params xmfile="/etc/xen/shared-vm/vm3"  
crm(live)configure# primitive xenD ocf:heartbeat:Xen \  
utilization hv_memory="1000" meta priority="5" \  
params xmfile="/etc/xen/shared-vm/vm4"  
crm(live)configure# property placement-strategy="minimal"
```

如果三个节点都处于正常状态，那么 xenA 将首先放置到一个节点上，然后是 xenD。xenB 和 xenC 将分配在一起或者其中一个与 xenD 分配在一起。

如果一个节点出现故障，可用的总内存将不足以托管所有资源。将确保分配 xenA，xenD 也同样如此。但是，只能再分配 xenB 和 xenC 中的一个，由于它们的优先级相同，结果不确定。要解决这种不确定性，需要为其中一个资源设置更高的优先级。

8.3.9 配置资源监视

要监视资源，有两种可能性：使用 **op** 关键字或 **monitor** 命令定义监视操作。以下示例使用 **op** 关键字配置 Apache 资源并且每 60 分钟监视一次：

```
crm(live)configure# primitive apache apache \  
  params ... \  
  op monitor interval=60s timeout=30s
```

同样也可以使用以下方式来实现：

```
crm(live)configure# primitive apache apache \  
  params ...  
crm(live)configure# monitor apache 60s:30s
```

有关概述，请参见第 6.4 节“资源监视”。

8.3.10 配置群集资源组

群集的一个最常见元素是需要放置在一起的一组资源。按顺序启动，并按相反顺序停止。为了简化此配置，我们支持组的概念。以下示例创建了两个原始资源（一个 IP 地址和一个电子邮件资源）：

1. 以系统管理员的身份运行 **crm** 命令。提示符更改为 **crm(live)**。
2. 配置这两个原始资源：

```
crm(live)# configure  
crm(live)configure# primitive Public-IP ocf:heartbeat:IPaddr \  
  params ip=1.2.3.4 id= Public-IP  
crm(live)configure# primitive Email systemd:postfix \  
  params id=Email
```

3. 以正确顺序按其相关标识符对原始资源进行分组：

```
crm(live)configure# group g-mailsvc Public-IP Email
```

要更改组成员的顺序，请使用 **configure** 子命令中的 **modgroup** 命令。使用以下命令可将原来的 **Email** 移到 **Public-IP** 前面。（只供展示功能之用）：

```
crm(live)configure# modgroup g-mailsvc add Email before Public-IP
```

如果您要从某个组去除资源（例如 `Email`），则使用以下命令：

```
crm(live)configure# modgroup g-mailsvc remove Email
```

有关概述，请参见第 6.3.5.1 节“组”。

8.3.11 配置克隆资源

最初将克隆构想成便于启动一个 IP 地址的 N 个实例并使它们分布在群集各处以保持负载平衡的一种方法。事实证明，它们可用于多种用途，包括与 DLM 集成、屏蔽子系统和 OCFS2。您可以克隆资源代理支持的任何资源。

要了解有关克隆资源的更多信息，请参见第 6.3.5.2 节“克隆资源”。

8.3.11.1 创建匿名克隆资源

要创建匿名克隆资源，首先要创建一个原始资源，然后使用 `clone` 命令来引用它。执行下列操作：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 配置原始资源，例如：

```
crm(live)configure# primitive Apache ocf:heartbeat:apache
```

3. 克隆原始资源：

```
crm(live)configure# clone cl-apache Apache
```

8.3.11.2 创建可升级克隆资源

可升级克隆资源（以前称为多状态资源）是一种特殊的克隆。此类型允许实例处于两种运行模式中的其中一种，可以是主动/被动、主要/次要或主/从模式。

要创建可升级克隆资源，首先要创建一个原始资源，然后再创建可升级克隆资源。可升级克隆资源必须至少支持升级和降级操作。

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 配置原始资源。必要时更改时间间隔：

```
crm(live)configure# primitive my-rsc ocf:myCorp:myAppl \  
    op monitor interval=60 \  
    op monitor interval=61 role=Master
```

3. 创建可升级克隆资源：

```
crm(live)configure# ms ms-rsc my-rsc
```

8.4 管理群集资源

除可用于配置群集资源外，`crm` 工具还可用于管理现有资源。以下小节进行了概述。

8.4.1 显示群集资源

当管理群集时，`crm configure show` 命令会列出诸如群集配置、全局选项、原始资源及其其他的当前 CIB 对象：

```
root # crm configure show  
node 178326192: alice  
node 178326448: bob  
primitive admin_addr IPAddr2 \  
    params ip=192.168.2.1 \  
    op monitor interval=10 timeout=20  
primitive stonith-sbd stonith:external/sbd \  
    params pcmk_delay_max=30  
property cib-bootstrap-options: \  
    have-watchdog=true \  

```

```

dc-version=1.1.15-17.1-e174ec8 \
cluster-infrastructure=corosync \
cluster-name=hacluster \
stonith-enabled=true \
placement-strategy=balanced \
standby-mode=true
rsc_defaults rsc-options: \
  resource-stickiness=1 \
  migration-threshold=3
op_defaults op-options: \
  timeout=600 \
  record-pending=true

```

如果您有许多资源，`show` 的输出会十分冗长。为限制输出，请使用资源名称。例如，如果只想列出原始资源 `admin_addr` 的属性，请将资源名称追加到 `show` 后：

```

root # crm configure show admin_addr
primitive admin_addr IPAddr2 \
  params ip=192.168.2.1 \
  op monitor interval=10 timeout=20

```

但在某些情况下，您可能希望更精确地限制特定资源的输出。那么，您可以使用过滤器。过滤器可将输出限定到特定组件。例如，要想仅列出节点，可使用 `type:node`：

```

root # crm configure show type:node
node 178326192: alice
node 178326448: bob

```

如果您还对原始资源感兴趣，请使用 `or` 运算符：

```

root # crm configure show type:node or type:primitive
node 178326192: alice
node 178326448: bob
primitive admin_addr IPAddr2 \
  params ip=192.168.2.1 \
  op monitor interval=10 timeout=20
primitive stonith-sbd stonith:external/sbd \
  params pcmk_delay_max=30

```

此外，要搜索以特定字符串开头的对象，请使用以下表示法：

```
root # crm configure show type:primitive and and 'admin*'
primitive admin_addr IPAddr2 \
    params ip=192.168.2.1 \
    op monitor interval=10 timeout=20
```

要列出所有可用类型，请输入 `crm configure show type:` 并按 `→|` 键。Bash 补全会为您提供所有类型的列表。

8.4.2 启动新的群集资源

要启动新的群集资源，您需要相应的标识符。按如下所示继续：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm
```

2. 切换到资源级别：

```
crm(live)# resource
```

3. 使用 `start` 启动资源，然后按 `→|` 键显示所有已知资源：

```
crm(live)resource# start ID
```

8.4.3 停止群集资源

要停止一个或多个现有群集资源，需要提供相应的标识符。按如下所示继续：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm
```

2. 切换到资源级别：

```
crm(live)# resource
```

3. 使用 `stop` 停止资源，然后按 `→|` 键显示所有已知资源：

```
crm(live)resource# stop ID
```

可以一次性停止多个资源：

```
crm(live)resource# stop ID1 ID2 ...
```

8.4.4 清理资源

如果资源失败，它会自动重启动，但每次失败都会增加资源的失败计数。如果为资源设置了 `migration-threshold`，当故障计数达到该迁移阈值时，将不再允许节点运行该资源。

1. 打开外壳并以 `root` 用户身份登录。

2. 获取所有资源的列表。

```
root # crm resource list
...
Resource Group: dlm-clvm:1
    dlm:1 (ocf:pacemaker:controld) Started
    clvm:1 (ocf:heartbeat:clvm) Started
```

3. 例如，要清理资源 `dlm`，请执行以下操作：

```
root # crm resource cleanup dlm
```

8.4.5 删除群集资源

请按如下操作以删除群集资源：

1. 以 `root` 用户身份登录，然后启动 `crm` 交互式外壳：

```
root # crm configure
```

2. 运行以下命令来获取您的资源列表：

```
crm(live)# resource status
```

例如，输出可能类似于以下内容（其中 myIP 是资源的相关标识符）：

```
myIP (ocf:IPaddr:heartbeat) ...
```

3. 删除具有相关标识符的资源（也暗指 `commit`）：

```
crm(live)# configure delete YOUR_ID
```

4. 提交更改：

```
crm(live)# configure commit
```

8.4.6 迁移群集资源

虽然资源已配置为在发生硬件或软件故障时自动故障转移（或迁移）到群集的其他节点，但您也可以使用 Hawk2 或命令行将资源手动迁移到其他节点。

对此任务使用 `migrate` 命令。例如，要将 `ipaddress1` 资源迁移到名为 `bob` 的群集节点，请使用以下命令：

```
root # crm resource  
crm(live)resource# migrate ipaddress1 bob
```

8.4.7 分组/标记资源

使用标记可以一次性引用多个资源，而无需在这些资源之间创建任何共置或顺序关系。此功能十分适用于对概念上相关的资源进行分组。例如，如果有多个资源与某个数据库相关，您可以创建一个名为 `databases` 的标记，并将与该数据库相关的所有资源都添加到此标记：

```
root # crm configure tag databases: db1 db2 db3
```

这样，只需使用一条命令就能启动所有这些资源：

```
root # crm resource start databases
```

同样，也可以一次性停止所有这些资源：

```
root # crm resource stop databases
```

8.4.8 获取运行状态

可以使用所谓的脚本来显示群集或节点的“运行”状态。脚本可以执行不同的任务，并不局限于显示运行状态。不过，本节重点介绍如何获取运行状态。

要获取有关 `health` 命令的所有细节，请使用 `describe`：

```
root # crm script describe health
```

该命令将显示所有参数及其默认值的说明和列表。要执行脚本，请使用 `run`：

```
root # crm script run health
```

如果您希望只运行整套命令中的一个步骤，可以使用 `describe` 命令列出步骤类别中的所有可用步骤。

例如，以下命令将执行 `health` 命令的第一个步骤。将在 `health.json` 文件中储存输出以供做进一步调查：

```
root # crm script run health
statefile='health.json'
```

您也可以使用 `crm cluster health` 运行以上命令。

有关脚本的更多信息，请参见 <http://crmsh.github.io/scripts/>。

8.5 设置独立于 `cib.xml` 的密码

如果群集配置包含密码之类的敏感信息，应将其储存在本地文件中。这样的话，这些参数将永远不会记录到或导入支持报告中。

使用 `secret` 前，最好先运行 `show` 命令了解一下所有资源的概况：

```
root # crm configure show
primitive mydb ocf:heartbeat:mysql \
  params replication_user=admin ...
```

要为上面的 `mydb` 资源设置口令，请使用以下命令：

```
root # crm resource secret mydb set passwd linux
INFO: syncing /var/lib/heartbeat/lrm/secrets/mydb/passwd to [your node list]
```

使用以下命令可以取回保存的密码：

```
root # crm resource secret mydb show passwd
linux
```

请注意，节点之间需要同步参数，使用 `crm resource secret` 命令可以帮助您处理好同步问题。强烈建议仅使用此命令管理机密参数。

8.6 检索历史记录信息

调查群集的历史记录是一项复杂的任务。为简化此任务，`crmsh` 包含了 `history` 命令及其子命令。假定已正确配置 SSH。

每个群集都会移动状态、迁移资源或启动重要进程。这些操作均可通过 `history` 子命令进行检索。

默认情况下，所有 `history` 命令会查看最近一小时的事件。要更改此时间段，请使用 `limit` 子命令。语法为：

```
root # crm history
crm(live)history# limit FROM_TIME [TO_TIME]
```

有效示例如下所示：

```
limit 4:00pm ,
```

```
limit 16:00
```

上述两个命令表达同一个意思：今天下午 4 点。

```
limit 2012/01/12 6pm
```

2012 年 1 月 12 日下午 6 点

```
limit "Sun 5 20:46"
```

当年当月 5 日（星期日）晚上 8:46

要查找更多示例以及如何创建时间段的信息，请访问 <http://labix.org/python-dateutil>。

info 子命令显示 **crm report** 涵盖的所有参数：

```
crm(live)history# info
Source: live
Period: 2012-01-12 14:10:56 - end
Nodes: alice
Groups:
Resources:
```

要将 **crm report** 限制为使用特定参数，请使用 **help** 子命令查看可用的选项。

要降低细节级别，请使用 **detail** 子命令及级别：

```
crm(live)history# detail 1
```

级别数字越高，报告就越详细。默认值为 0（零）。

设置上述参数后，使用 **log** 显示日志消息。

要显示上次转换操作，请使用以下命令：

```
crm(live)history# transition -1
INFO: fetching new logs, please wait ...
```

此命令会提取日志并运行 **dotty**（从 **graphviz** 包）以显示转换图。外壳会打开日志文件，您可以在其中使用 **↓** 和 **↑** 光标键浏览内容。

如果希望不要打开转换图，请使用 **nograph** 选项：

```
crm(live)history# transition -1 nograph
```

8.7 更多信息

- [crm 手册页](#)。
- 访问 <http://crmsb.github.io/documentation> 中的上游项目文档。
- 有关详尽示例，请参见《使用 DRBD 和 Pacemaker 的高度可用 NFS 储存》文章。

9 添加或修改资源代理

需由群集管理的所有任务都必须可用作资源。在此处需要考虑两个主要组：资源代理和 STONITH 代理。对于这两个类别，您都可以添加自己的代理，根据需要扩展群集的功能。

9.1 STONITH 代理

群集有时会检测到某个节点行为异常，需要删除此节点。这称为屏蔽，通常使用 STONITH 资源实现。



警告：不支持外部 SSH/STONITH

由于无法了解 SSH 可能对其他系统问题如何做出反应。出于此原因，生产环境不支持外部 SSH/STONITH 代理（例如 `stonith:external/ssh`）。如果您仍要使用此类代理进行测试，请安装 `libglue-devel` 包中提供。

要（从软件端）获取所有当前可用的 STONITH 设备列表，请使用 `crm ra list stonith` 命令。如果您找不到收藏的代理，请安装 `-devel` package. 有关 STONITH 设备和资源代理的详细信息，请参见第 10 章“[屏障和 STONITH](#)”。

目前尚无有关写入 STONITH 代理的文档。如果要编写新的 STONITH 代理，请参见以下包的源代码中提供的示例：`cluster-glue` 包中提供。

9.2 写入 OCF 资源代理

所有 OCF 资源代理 (RA) 都可在 `/usr/lib/ocf/resource.d/` 中找到，请参见第 6.3.2 节“[支持的资源代理类](#)”了解更多信息。每个资源代理都必须支持以下操作才能进行控制：

start

启动或启用资源

stop

停止或禁用资源

status

返回资源状态

monitor

与 **status** 类似，但还会检查是否存在意外状态

validate

验证资源配置

meta-data

返回有关资源代理的 XML 格式的信息

创建 OCF RA 的常规过程大概如下：

1. 将文件 `/usr/lib/ocf/resource.d/pacemaker/Dummy` 装载为模板。
2. 为每个新资源代理创建新的子目录，以避免发生命名冲突。例如，如果您的一个资源组 `kitchen` 具有资源 `coffee_machine`，可将此资源添加到目录 `/usr/lib/ocf/resource.d/kitchen/`。要访问此资源代理，请执行命令 `crm`：

```
root # crm configure primitive coffee_1 ocf:coffee_machine:kitchen ...
```

3. 实施其他外壳功能，并用不同名称保存文件。

可在 http://www.linux-ha.org/wiki/Resource_Agents 中找到有关写入 OCF 资源代理的更多细节。在第 1 章“产品概述”中可以找到有关若干概念的特殊信息。

9.3 OCF 返回码和故障恢复

根据 OCF 规范，有一些关于操作必须返回的退出代码的严格定义。群集会始终检查返回代码与预期结果是否相符。如果结果与预期值不匹配，则将操作视为失败，并将启动恢复操作。有三种类型的故障恢复：

表 9.1：故障恢复类型

恢复类型	描述	群集执行的操作
软	发生临时错误。	重新启动资源或将它移到新位置。
硬	发生非临时错误。错误可能特定于当前节点。	将资源移到别处，避免在当前节点上重试该资源。
致命	发生所有群集节点共有的非临时错误。这表示指定了错误配置。	停止资源，避免在任何群集节点上启动该资源。

假定某个操作被视为已失败，下表概括了不同的 OCF 返回代码。此外，该表还显示了收到相应的错误代码时群集将启动的恢复类型。

表 9.2：OCF 返回代码

OCF 返回代码	OCF 别名	描述	恢复类型
0	OCF_SUCCESS	成功。命令成功完成。这是所有启动、停止、升级和降级命令的所需结果。	软
1	OCF_ERR_GENERIC	通用“出现问题”错误代码。	软
2	OCF_ERR_ARGS	资源配置在此计算机上无效（例如，它引用了在节点上找不到的位置/工具）。	硬
3	OCF_ERR_UNIMPLEMENTED	请求的操作未实现。	硬
4	OCF_ERR_PERM	资源代理没有足够的特权，不能完成此任务。	硬
5	OCF_ERR_NOT_INSTALLED	资源所需的工具未安装在此计算机上。	硬

OCF 返回代码	OCF 别名	描述	恢复类型
6	OCF_ERR_- CONFIGURED	资源配置无效（例如，缺少必需的参数）。	致命
7	OCF_NOT_- RUNNING	资源未运行。群集将不会尝试停止为任何操作返回此代码的资源。 此 OCF 返回代码可能需要或不需要资源恢复，这取决于所需的资源状态。如果出现意外，则执行软恢复。	不适用
8	OCF_RUNNING_- MASTER	资源正在主节点中运行。	软
9	OCF_FAILED_- MASTER	资源在主节点中，但已失败。资源将再次被降级、停止再重新启动（然后也可能升级）。	软
其他	不适用	自定义错误代码。	软

10 屏障和 STONITH

屏障在 HA（高可用性）计算机群集中是一个非常重要的概念。群集有时会检测到某个节点行为异常，需要删除此节点。这称为屏障，通常使用 STONITH 资源实现。屏障可以定义为一种使 HA 群集具有已知状态的方法。

群集中的每个资源均带有状态。例如：“资源 r1 已在 alice 上启动”。在 HA 群集中，这种状态暗示了“资源 r1 在除 alice 外的所有节点上都处于停止状态”，因为群集必须确保每个资源只能在一个节点上启动。每个节点都必须报告资源发生的每个更改。这样群集状态就是资源状态和节点状态的集合。

当节点或资源的状态无法十分肯定地确立时，将进行屏障。即使在群集未感知到给定节点上发生的事件时，屏障也可确保此节点不会运行任何重要资源。

10.1 屏障分类

有两类屏障：资源级别屏障和节点级别屏障。后者是本章的主题。

资源级别屏障

资源级别屏障可确保对给定资源的排它访问。此情况的常见示例就是通过 SAN 光纤通道开关（用于锁定节点不让访问其磁盘）或 SCSI 保留之类的方法更改节点的区域。有关示例，请参见第 11.10 节“储存保护的其他机制”。

节点级别屏障

节点级别屏障可彻底防止故障节点访问共享资源。这种屏障通常采用一种简单但却粗暴的方式来完成，即重设置或关闭节点。

10.2 节点级别屏障

在 Pacemaker 群集中，节点级别屏障的实施方式是 STONITH（关闭其他节点）。High Availability Extension 包括 `stonith` 命令行工具，一个能远程关闭群集中节点的可扩展界面。有关可用选项的概述，请运行 `stonith --help` 或参见 `stonith` 的手册页了解更多信息。

10.2.1 STONITH 设备

要使用节点级屏蔽，首先需要屏蔽设备。要获取 High Availability Extension 所支持的 STONITH 设备的列表，请在任何节点上运行以下命令之一：

```
root # stonith -L
```

或者

```
root # crm ra list stonith
```

STONITH 设备可分为以下类别：

电源分配单元 (PDU)

电源分发单元是管理关键网络、服务器和数据中心设备的电源容量和功能的基本元素。它可以提供对已连接设备的远程负载监视和独立电源出口控制，以实现远程电源循环。

不间断电源 (UPS)

公共用电出现故障时，通过其他来源供电的稳定电源可为连接的设备提供应急电源。

刀片电源控制设备

如果是在刀片组上运行群集，则刀片外壳中的电源控制设备就是提供屏蔽的唯一候选。当然，此设备必须能够管理单个刀片计算机。

无人值守设备

无人值守设备（IBM RSA、HP iLO 和 Dell DRAC）正变得越来越普遍，在未来它们甚至可能成为现成可用计算机上的标准配置。然而，它们相比 UPS 设备有一点不足，因为它们与主机（群集节点）共享一个电源。如果节点持续断电，则认为控制该节点的设备失去作用。在这种情况下，CRM 将继续无限期地尝试屏蔽节点，而所有其他资源操作都将等待屏蔽/STONITH 操作完成。

测试设备

测试设备仅用于测试目的。它们通常对硬件更加友好。将群集投放到生产环境之前，必须以真实的屏蔽设备进行替换。

对 STONITH 设备的选择主要取决于您的预算和所用硬件的种类。

10.2.2 STONITH 实施

SUSE® Linux Enterprise High Availability Extension 的 STONITH 实施由两个组件组成：

pacemaker-fenced

`pacemaker-fenced` 是可由本地进程或通过网络访问的守护程序。它接受与屏蔽操作（重置、关闭电源和打开电源）对应的命令。它还可以检查屏蔽设备的状态。

`pacemaker-fenced` 守护程序在高可用性群集中的每个节点上运行。在 DC 节点上运行的 `pacemaker-fenced` 实例从 `pacemaker-controld` 接收屏蔽请求。由此实例及其他 `pacemaker-fenced` 程序决定是否要执行所需的屏蔽操作。

STONITH 插件

对于每个受支持的屏蔽设备，都有一个能够控制所述设备的 STONITH 插件。STONITH 插件是屏障设备的界面。STONITH 插件包含在 `cluster-glue` 包中，位于每个节点上的 `/usr/lib64/stonith/plugins` 下。（如果您同时安装了 `fence-agents` 包，则会将其中包含的插件安装在 `/usr/sbin/fence_*` 中。）对于 `pacemaker-fenced` 来说，所有 STONITH 插件看上去都一样，但在反映屏蔽设备性质的方面则大不相同。

某些插件支持多个设备。`ipmilan`（或 `external/ipmi`）就是一个典型的示例，它实施 IPMI 协议并可以控制任何支持此协议的设备。

10.3 STONITH 资源和配置

要设置屏蔽，需要配置一个或多个 STONITH 资源 - `pacemaker-fenced` 守护程序不需要配置。所有配置都储存在 CIB 中。STONITH 资源属于 `stonith` 类的资源（请参见第 6.3.2 节“支持的资源代理类”）。STONITH 资源是 STONITH 插件在 CIB 中的代表。除了屏蔽操作，还可以启动、停止和监视 STONITH 资源，就像任何其他资源一样。启动或停止 STONITH 资源意味着装载或卸载节点上的 STONITH 设备驱动程序。启动和停止仅是管理操作，不会转换成对屏蔽设备自身的任何操作。然而，监视操作却会转换成将其记录到设备（以校验设备能否在需要时正常运行）。STONITH 资源故障转移到另一个节点时，它通过装载相应的驱动程序允许当前节点与 STONITH 设备对话。

STONITH 资源可像任何其他资源一样进行配置。有关如何使用首选群集管理工具执行此操作的细节：

- Hawk2: 第 7.5.6 节 “添加 STONITH 资源”
- crmsh: 第 8.3.4 节 “创建 STONITH 资源”

参数（属性）列表取决于相应的 STONITH 类型。要查看特定设备的参数列表，请使用 **stonith** 命令：

```
stonith -t stonith-device-type -n
```

例如，要查看 `ibmhmc` 设备类型的参数，请输入以下命令：

```
stonith -t ibmhmc -n
```

要获取设备的简短帮助文本，请使用 `-h` 选项：

```
stonith -t stonith-device-type -h
```

10.3.1 STONITH 资源配置示例

在以下部分中，可了解一些用 `crm` 命令行工具的语法编写的示例配置。要应用这些配置，请将示例放进文本文件（例如 `sample.txt`）并运行：

```
root # crm < sample.txt
```

有关使用 `crm` 命令行工具配置资源的更多信息，请参见第 8 章 “配置和管理群集资源（命令行）”。

例 10.1：IBM RSA 无人值守设备的配置

可以如下配置 IBM RSA 无人值守设备：

```
configure
primitive st-ibmrsa-1 stonith:external/ibmrsa-telnet \
params nodename=alice ip_address=192.168.0.101 \
username=USERNAME password=PASSWORD
primitive st-ibmrsa-2 stonith:external/ibmrsa-telnet \
params nodename=bob ip_address=192.168.0.102 \
username=USERNAME password=PASSWORD
location l-st-alice st-ibmrsa-1 -Mandatory: alice
location l-st-bob st-ibmrsa-2 -Mandatory: bob
commit
```

在此示例中，由于以下原因使用了位置约束：STONITH 操作始终有失败的可能性。因此，在同时兼作执行程序节点上操作 STONITH 并不可靠。如果重设置节点，则它将无法发送有关屏蔽操作结果的通知。唯一的方法是假设操作会成功并提前发送通知。不过，如果操作失败，可能会出现问題。因此，按惯例 `pacemaker-fenced` 会拒绝终止其主机。

例 10.2：UPS 屏蔽设备的配置

UPS 类型屏蔽设置的配置类似于上面的示例。此处不作详细介绍。所有 UPS 设备均使用相同的机制屏蔽。访问设备的方式有所不同。旧的 UPS 设备只有一个串行端口，通常使用特殊的串行电缆以 1200 波特的速率进行连接。许多新的 UPS 设备仍有一个串行端口，但它们一般还使用 USB 或以太网接口。可以使用的连接类型取决于插件支持的连接。

例如，通过使用 `stonith -t stonith 设备类型 -n` 命令比较 `apcmaster` 与 `apcsmart` 设备：

```
stonith -t apcmaster -h
```

返回以下信息：

```
STONITH Device: apcmaster - APC MasterSwitch (via telnet)
NOTE: The APC MasterSwitch accepts only one (telnet)
connection/session a time. When one session is active,
subsequent attempts to connect to the MasterSwitch will fail.
For more information see http://www.apc.com/
List of valid parameter names for apcmaster STONITH device:
    ipaddr
    login
    password

For Config info [-p] syntax, give each of the above parameters in order as
the -p value.
Arguments are separated by white space.
Config file [-F] syntax is the same as -p, except # at the start of a line
denotes a comment
```

使用

```
stonith -t apcsmart -h
```

得到以下输出：

```
STONITH Device: apcsmart - APC Smart UPS
(via serial port - NOT USB!).
Works with higher-end APC UPSes, like
Back-UPS Pro, Smart-UPS, Matrix-UPS, etc.
(Smart-UPS may have to be >= Smart-UPS 700?).
See http://www.networkupstools.org/protocols/apcsmart.html
for protocol compatibility details.
For more information see http://www.apc.com/
List of valid parameter names for apcsmart STONITH device:
ttydev
hostlist
```

第一个插件支持带有一个网络端口的 APC UPS 和 telnet 协议。第二个插件使用 APC SMART 协议（通过许多 APC UPS 产品系列都支持的串行线路）。

例 10.3：KDUMP 设备的配置

Kdump 属于**特殊的屏蔽设备**，实际上与屏蔽设备相反。该插件检查节点上是否正在进行内核转储。如果是，它将返回 true，并如同节点已被屏蔽那样进行操作。

必须以与其他真实的 STONITH 设备（例如 `external/ipmi`）一致的方式使用 Kdump 插件。要正常运行屏蔽机制，必须在触发真实的 STONITH 设备之前，指定 Kdump 已经过检查。请按以下过程中所述，使用 `crm configure fencing_topology` 来指定屏蔽设备的顺序。

1. 使用 `stonith:fence_kdump` 资源代理（由包 `fence-agents` 提供）以监视启用 Kdump 功能的所有节点。下面提供了资源的配置示例：

```
configure
primitive st-kdump stonith:fence_kdump \
  params nodename="alice" \ ❶
  pcmk_host_check="static-list" \
  pcmk_reboot_action="off" \
  pcmk_monitor_action="metadata" \
  pcmk_reboot_retries="1" \
  timeout="60"
commit
```

- ❶ 要监控的节点的名称。如果您需要监视多个节点，请配置更多 STONITH 资源。要防止特定节点使用屏蔽设备，请添加位置约束。

当资源超时后，屏蔽操作将会启动。

2. 在每个节点上的 `/etc/sysconfig/kdump` 中，将 `KDUMP_POSTSCRIPT` 配置为在 Kdump 进程完成后向所有节点发送通知。例如：

```
/usr/lib/fence_kdump_send -i INTERVAL -p PORT -c 1 alice bob charlie  
[...]
```

执行 Kdump 的节点将在完成 Kdump 后自动重新启动。

3. 写入新的 `initrd`，以包含支持网络的 `fence_kdump_send` 库。使用 `-f` 选项重写现有文件，以便在下一次引导过程中使用新文件：

```
root # dracut -f -a kdump
```

4. 在防火墙中针对 `fence_kdump` 资源打开一个端口。默认端口为 `7410`。
5. 为了能够在触发真实的屏蔽机制（例如 `external/ipmi`）之前检查 Kdump，请使用类似以下的配置：

```
fencing_topology \  
  alice: kdump-node1 ipmi-node1 \  
  bob: kdump-node2 ipmi-node2
```

有关 `fencing_topology` 的详细信息：

```
crm configure help fencing_topology
```

10.4 监视屏蔽设备

与任何其他资源一样，STONITH 类代理还支持使用监视操作检查状态。



注意：监视 STONITH 资源

请定期而谨慎地监视 STONITH 资源。对于大多数设备而言，至少 1800 秒（30 分钟）的监视间隔应已足够。

屏蔽设备是 HA 群集不可缺少的组成部分，但越少需要使用它们越好。电源管理设备常常会受广播流量过多的影响。某些设备无法处理每分钟多于十个左右连接的情况。如果两个客户端同时尝试进行连接，一些设备会分辨不清。大多数设备不能同时处理多个会话。

通常，每隔几小时检查一次屏蔽设备的状态应已足够。需要执行屏蔽操作和电源开关故障的情况是较少的。

有关如何配置监视操作的详细信息，请参见针对命令行方法的第 8.3.9 节“配置资源监视”。

10.5 特殊的屏蔽设备

除了处理真实 STONITH 设备的插件外，还有特殊用途的 STONITH 插件。



警告：仅供测试

下面提到的一些 STONITH 插件仅供演示和测试之用。不要在实际情境中使用以下任何设备，因为这可能导致数据损坏和无法预料的结果：

- [external/ssh](#)
- [ssh](#)

[fence_kdump](#)

此插件检查节点上是否正在进行内核转储。如果有，它将返回 `true`，并按节点已被屏蔽那样进行操作。在转储过程中，此节点不能运行任何资源。这可避免屏蔽已关闭但正在进行转储的节点，从而节省屏蔽所需时间。此插件必须与另一个实际 STONITH 设备一同使用。

有关配置细节，请参见例 10.3 “Kdump 设备的配置”。

[external/sbd](#)

这是一个自屏蔽设备。它对可以插入共享磁盘的所谓的“毒药”作出反应。当中断共享储存区连接时，它将停止节点运行。要了解如何使用此 STONITH 代理实施基于储存的屏蔽，请参见第 11 章、过程 11.7 “将群集配置为使用 SBD”。有关更多细节，另请参见 http://www.linux-ha.org/wiki/SBD_Fencing。

! 重要: external/sbd 和 DRBD

external/sbd 屏蔽机制要求能直接从每个节点读取 SBD 分区。因此，SBD 分区中不得使用 DRBD* 设备。

但是，如果 SBD 分区位于未镜像或未复制的共享磁盘上，则可以对 DRBD 群集使用该屏蔽机制。

external/ssh

另一个基于软件的“屏蔽”机制。节点必须能够以 root 身份相互登录，而且无需密码。它使用一个参数 hostlist 指定它将指向的目标节点。由于不能重置已确实失败的节点，它不得用于实际群集 - 仅供测试和演示之用。将其用于共享储存将导致数据损坏。

meatware

meatware 需要用户操作才能运行。调用 meatware 时，它会记录一条 CRIT 严重性消息，显示在节点的控制台上。然后操作员确认节点已关闭，并发出 meatclient(8) 命令。此命令指示 meatware 通知群集将该节点视为已出现故障。有关更多信息，请参见 /usr/share/doc/packages/cluster-glue/README.meatware。

suicide

这是一个仅有软件的设备，它可以使用 reboot 命令重引导它运行所处的节点。这需要节点的操作系统的操作，在某些情况下可能失败。因此，如果可能，请避免使用此设备。然而，在单节点群集上使用此设备是很安全的。

无磁盘 SBD

如果您想要建立一个不含共享储存的屏蔽机制，则此配置十分有用。在此无磁盘模式下，SBD 会使用硬件检查包来屏蔽节点，而不依赖于任何共享设备。不过，无磁盘 SBD 不能处理双节点群集的节点分裂情况。此选项仅适用于具有两个以上节点的群集。

suicide 是“I do not shoot my host”（我自己不关闭我的主机）规则的唯一例外。

10.6 基本建议

请查看以下建议列表以避免常见错误：

- 不要并行配置多个电源开关。
- 要测试 STONITH 设备及其配置，请从每个节点拔出一次插头，并校验该节点是否会被屏蔽。
- 在负载状态下测试资源，并校验超时值是否合适。超时值设置得过短会触发（不必要的）屏蔽操作。有关详细信息，请参见第 6.3.9 节“超时值”。
- 对您的设置使用合适的屏蔽设备。有关细节，另请参见第 10.5 节“特殊的屏蔽设备”。
- 配置一个或多个 STONITH 资源。默认情况下，全局群集选项 `stonith-enabled` 设置为 `true`。如果未定义任何 STONITH 资源，群集将拒绝启动任何资源。
- 不要将全局群集选项 `stonith-enabled` 设置为 `false`，原因如下：
 - 未启用 STONITH 的群集不受支持。
 - DLM/OCFS2 将会阻止一直等待不会发生的屏蔽操作。
- 不要将全局群集选项 `startup-fencing` 设置为 `false`。默认情况下，会因以下原因将其设置为 `true`：如果节点在群集启动期间状态未知，则它将会被屏蔽一次以明确其状态。

10.7 更多信息

</usr/share/doc/packages/cluster-glue>

在已安装系统中，此目录包含多个 STONITH 插件和设备的自述文件。

<http://www.linux-ha.org/wiki/STONITH> ↗

有关 STONITH 的信息位于 High Availability Linux 项目的主页中。

<http://www.clusterlabs.org/pacemaker/doc/> ↗

- 《Pacemaker Explained》（Pacemaker 配置说明）：说明用于配置 Pacemaker 的概念。包含全面而非常详细的信息供参考。

http://techthoughts.typepad.com/managing_computers/2007/10/split-brain-quo.html ↗

说明 HA 群集中节点分裂、法定票数和屏蔽的概念的文章。

11 储存保护和 SBD

SBD（STONITH 块设备）通过经由共享块储存（SAN、iSCSI、FCoE 等）交换讯息来为基于 Pacemaker 的群集提供节点屏蔽机制。此方法可以将屏蔽机制隔离开来，使其不受固件版本更改的影响或不依赖于特定固件控制器。SBD 需要在每个节点上安装一个检查包，以确保能确实停止行为异常的节点。在某些情况下，还可以通过无磁盘模式运行 SBD，以便使用不含共享储存的 SBD。

此 `ha-cluster-bootstrap` 脚本提供了一种自动设置群集的方式，并可选择使用 SBD 作为屏蔽机制。有关详细信息，请参见《安装和设置快速入门》文章。但是，手动设置 SBD 可为您提供个别设置的更多选项。

本章介绍 SBD 背后的概念。它将指导您完成 SBD 所需的组件配置，以保护您的群集防止在发生节点分裂情况下出现可能的数据损坏。

除了节点级别屏蔽，您还可以使用额外的储存保护机制，例如 LVM2 排它激活或 OCFS2 文件锁定支持（资源级别屏蔽）。它们可以保护您的系统，以防出现管理或应用程序故障。

11.1 概念概述

SBD 是 Storage-Based Death（基于储存区的终止）或 STONITH Block Device（STONITH 块设备）的缩写。

高可用性群集堆栈的最高优先级是保护数据完整性。此项保护通过防止对数据储存进行未协调的并行访问来实现。群集堆栈会使用几种控制机制来实现此目标。

但是，如果在群集中选出数个 DC，则可能导致网络分区或软件故障。如果允许出现这种所谓的“节点分裂”情况，则可能会发生数据损坏。

避免出现这种情况可采用的主要机制是通过 STONITH 实现节点屏蔽。如果使用 SBD 作为节点屏蔽机制可在节点分裂情况下关闭节点而无需使用外部关闭设备。

SBD 分区

在所有节点都可访问共享储存的环境中，设备的某个小分区会格式化，以用于 SBD。该分区的大小取决于所用磁盘的块大小（例如，对于块大小为 512 字节的标准 SCSI 磁盘，该分区大小为 1 MB；块大小为 4 KB 的 DASD 磁盘需要 4 MB 大小的分区）。初始化过程会在设备上创建讯息布局，配置最多 255 个节点的讯息槽。

SBD 守护程序

配置完相应的 SBD 守护程序后，在每个节点上使其联机，然后启动其余群集堆栈。它在所有其他群集组件都关闭之后才终止，从而确保了群集资源绝不会在没有 SBD 监督的情况下被激活。

讯息

此守护程序会自动将分区上的消息槽之一分配给其自身，并持续监视其中有无发送给它自己的消息。收到消息后，守护程序会立即执行请求，如启动关闭电源或重引导循环以进行屏蔽。

另外，此守护程序会持续监视与储存设备的连接性，当无法连接分区时会自行终止。这就保证了它不会从屏蔽消息断开连接。如果群集数据驻留在不同分区中的同一个逻辑单元，则一旦与储存失去连接，工作负载便将终止，因此不会增加额外的故障点。

检查包

只要使用 SBD，就必须确保检查包正常工作。新式系统支持硬件检查包，此功能需由软件组件来“激发”或“馈送数据”。软件组件（在此案例中为 SBD 守护程序）通过将服务脉冲定期写入检查包来“供给”检查包。如果守护程序停止供给检查包，硬件将强制系统重新启动。这可防止出现 SBD 进程本身的故障，如失去响应或由于 I/O 错误而卡住。

如果 Pacemaker 集成已激活，则当设备大多数节点丢失时，SBD 将不会进行自我屏蔽。例如，假设您的群集包含三个节点：A、B 和 C。由于网络分隔，A 只能看到它自己，而 B 和 C 仍可相互通讯。在此案例中，有两个群集分区，一个因节点占多数（B 和 C）而具有法定票数，而另一个则不具有 (A)。如果在大多数屏蔽设备无法访问时发生此情况，则节点 A 会立即自我关闭，而节点 B 和 C 将会继续运行。

11.2 手动设置 SBD 概述

手动设置基于储存的保护时必须执行以下步骤：必须以 `root` 身份执行这些步骤。在开始执行之前，请查看第 11.3 节“要求”。

1. 设置检查包
2. 根据您的情况，可将 SBD 与一到三个设备搭配使用，或以无磁盘模式使用。有关概述，请参见第 11.4 节“SBD 设备的数量”。有关详细的设置，请参见：
 - 设置 SBD 与设备
 - 设置无磁盘 SBD
3. 测试 SBD 和屏蔽

11.3 要求

- 最多可将三个 SBD 设备用于基于储存的屏蔽。使用一到三个设备时，必须可从所有节点访问共享储存。
- 群集中的所有节点上，共享存储设备的路径都必须永久且一致。使用稳定的设备名称，如 `/dev/disk/by-id/dm-uuid-part1-mpath-abcdef12345`。
- 可通过光纤通道 (FC)、以太网光纤通道 (FCoE) 甚至 iSCSI 来连接共享储存。
- 共享储存段不得使用基于主机的 RAID、LVM2 或 DRBD*。DRBD 可能已分割，这会导致 SBD 发生问题，因为 SBD 中不能存在两种状态。不能将群集多设备（群集 MD）用于 SBD。
- 但是，建议使用基于储存区的 RAID 和多路径，以提高可靠性。
- 可以在不同群集之间共享某个 SBD 设备，只要共享该设备的节点数不超过 255 个。
- 对于具有两个以上节点的群集，还可以在无磁盘模式下使用 SBD。

11.4 SBD 设备的数量

SBD 支持最多使用三个设备：

一个设备

最简单的实施。适用于所有数据位于同一个共享储存的群集。

两个设备

此配置主要用于使用基于主机的镜像但未提供第三个存储设备的环境。SBD 在丢失对某个镜像分支的访问权后将自我终止，以允许群集继续运行。但是，由于 SBD 不具备足够的知识可以检测到存储区的不对称分裂，因此在只有一个镜像分支可用时它不会屏蔽另一个分支。如此一来，就无法在存储阵列中的一个关闭时对第二个故障自动容错。

三个设备

最可靠的配置。它具有从一个设备中断（可能是因为故障或维护）的情况中恢复的能力。只有当一个以上设备丢失及必要时，SBD 才会自行终止，具体取决于群集分区或节点的状态。如果至少有两个设备仍然可访问，便能成功传输屏蔽讯息。此配置适用于存储未限制为单个阵列的更为复杂的环境。基于主机的镜像解决方案可以每个镜像分支拥有一个 SBD（不自我镜像），并在 iSCSI 上有一个额外的决定项。

无磁盘

如果您想要建立一个不含共享储存的屏蔽机制，则此配置十分有用。在此无磁盘模式下，SBD 会使用硬件检查包来屏蔽节点，而不依赖于任何共享设备。不过，无磁盘 SBD 不能处理双节点群集的节点分裂情况。此选项仅适用于具有两个以上节点的群集。

11.5 超时计算

使用 SBD 作为屏蔽机制时，必须考虑所有组件的超时，因为它们之间相互依赖。

检查包超时

此超时在初始化 SBD 设备期间设置。它主要取决于储存延迟。必须可在此时间内成功读取大多数设备。否则，节点可能会自我屏蔽。



注意：多路径或 iSCSI 设置

如果 SBD 设备驻留在多路径设置或 iSCSI 上，则应将超时设置为检测到路径故障并切换到下一个路径所需的时间。

这还意味着在 `/etc/multipath.conf` 中，`max_polling_interval` 的值必须小于 `watchdog` 超时。

`msgwait` 超时

此超时在初始化 SBD 设备期间设置。它定义了将讯息写入到 SBD 设备上的某个节点槽后经过多长时间视为已传递。该超时应设置的足够长，让节点有时间检测到它是否需要自我屏蔽。

但是，如果 `msgwait` 超时相对较长，已屏蔽的群集节点可能会在屏蔽操作返回之前便又重新加入。此情况可通过在 SBD 配置中设置 `SBD_DELAY_START` 参数来缓解，请参见步骤 4 中的过程 11.4。

CIB 中的 `stonith-timeout`

此超时在 CIB 中作为全局群集属性设置。它定义了等待 STONITH 操作（重引导、打开、关闭）完成的时间。

CIB 中的 `stonith-watchdog-timeout`

此超时在 CIB 中作为全局群集属性设置。如果未显式设置，则默认值为 `0`，此值适用于 SBD 与一到三个设备搭配使用的情况。若要以无磁盘模式使用 SBD，请参见过程 11.8 “配置无磁盘 SBD” 以获取详细信息。

如果您更改检查包超时，则需要同时调整另外两个超时。以下“公式”表达了这三个值之间的关系：

例 11.1：超时计算公式

```
Timeout (msgwait) >= (Timeout (watchdog) * 2)
stonith-timeout = Timeout (msgwait) + 20%
```

例如，如果您将检查包超时设置为 `120`，则请将 `msgwait` 超时设置为 `240`，并将 `stonith-timeout` 设置为 `288`。

如果您使用 `ha-cluster-bootstrap` 脚本设置群集并初始化 SBD 设备，则会自动考虑这些超时之间的关系。

11.6 设置检查包

SUSE Linux Enterprise High Availability Extension 随附了几个内核模块用于提供硬件特定的检查包驱动程序。有关最常用检查包驱动程序列表，请参见[常用检查包驱动程序](#)。

对于生产环境中的群集，建议使用硬件特定的检查包驱动程序。不过，如果没有与您的硬件匹配的检查包，则可以将 `softdog` 用作内核检查包模块。

High Availability Extension 使用 SBD 守护程序作为“供给”检查包的软件组件。

11.6.1 使用硬件检查包

查找给定系统的正确检查包内核模块并非没有意义。自动探测常常会失败。因此，在装载正确的模块之前，许多模块就已装载。

[表 11.1](#) 列出了最常用的检查包驱动程序。如果您的硬件未列在其中，也可从目录 `/lib/modules/KERNEL_VERSION/kernel/drivers/watchdog` 提供的列表中选择。或者，咨询您的硬件或系统供应商，获取有关特定于系统的检查包配置的细节。

表 11.1：常用检查包驱动程序

硬件	驱动程序
HP	<code>hpwdt</code>
Dell、Lenovo (Intel TCO)	<code>iTCO_wdt</code>
Fujitsu	<code>ipmi_watchdog</code>
IBM 大型机上的 VM 或 z/VM	<code>vmwatchdog</code>
Xen VM (DomU)	<code>xen_xdt</code>
通用	<code>softdog</code>

重要：访问检查包计时器

有些硬件供应商交付的系统管理软件（例如 HP ASR 守护程序）会使用检查包来进行系统重设置。如果 SBD 使用了检查包，请禁用此类软件。不能有其他任何软件在访问检查包计时器。

过程 11.1：装载正确的内核模块

要确保装载正确的检查包模块，请执行如下操作：

1. 列出已随内核版本安装的驱动程序：

```
root # rpm -ql kernel-VERSION | grep watchdog
```

2. 列出内核中当前装载的任何检查包模块：

```
root # lsmod | egrep "(wd|dog)"
```

3. 如果返回了结果，请卸载错误的模块：

```
root # rmmod WRONG_MODULE
```

4. 启用与您的硬件匹配的检查包模块：

```
root # echo WATCHDOG_MODULE > /etc/modules-load.d/watchdog.conf  
root # systemctl restart systemd-modules-load
```

5. 测试是否已正确装载检查包模块：

```
root # lsmod | grep dog
```

11.6.2 使用软件检查包 (softdog)

对于生产环境中的群集，建议使用硬件特定的检查包驱动程序。不过，如果没有与您的硬件匹配的检查包，则可以将 `softdog` 用作内核检查包模块。

! 重要：Softdog 限制

Softdog 驱动程序假设至少有一个 CPU 仍然在运行。如果所有 CPU 均已阻塞，则 softdog 驱动程序中应该重引导系统的代码永远都不会执行。相反地，即使所有 CPU 均已阻塞，硬件检查包也仍然会继续工作。

过程 11.2：装载 SOFTDOG 内核模块

1. 启用 softdog 驱动程序：

```
root # echo softdog > /etc/modules-load.d/watchdog.conf
```

2. 在 `/etc/modules-load.d/watchdog.conf` 中添加 `softdog` 模块并重启动服务:

```
root # echo softdog > /etc/modules-load.d/watchdog.conf
root # systemctl restart systemd-modules-load
```

3. 测试是否已正确装载 `softdog` 检查包模块:

```
root # lsmod | grep softdog
```

11.7 设置 SBD 与设备

进行该设置必须执行以下步骤:

1. 初始化 SBD 设备
2. 编辑 SBD 配置文件
3. 启用和启动 SBD 服务
4. 测试 SBD 设备
5. 将群集配置为使用 SBD

在开始之前, 请确保要用于 SBD 的一个或多个块设备满足在第 11.3 节中指定的要求。

设置 SBD 设备时, 您需要考虑几个超时值。有关细节, 请参见第 11.5 节“超时计算”。

如果节点上运行的 SBD 守护程序未足够快速地更新检查包计时器, 则节点会自行终止。设置超时后, 请在您的特定环境中予以测试。

过程 11.3: 初始化 SBD 设备

要将 SBD 与共享储存搭配使用, 必须先在一到三个块设备上创建讯息布局。`sbd create` 命令会将元数据头写入指定的一个或多个设备。它还将初始化最多 255 个节点的讯息槽。如果该命令不带任何其他选项, 则执行时将使用默认的超时设置。



警告：覆盖现有数据

确保要用于 SBD 的一个或多个设备未保存任何重要数据。执行 `sbd create` 命令时，会直接重写指定块设备的大约第一个 MB，而不会进一步发出请求或进行备份。

1. 决定要将哪个块设备或哪些块设备用于 SBD。
2. 使用以下命令初始化 SBD 设备：

```
root # sbd -d /dev/SBD create
```

(请将 `/dev/SBD` 替换为实际路径名称，例如：`/dev/disk/by-id/scsi-ST2000DM001-0123456_Wabcdfg`。)

要将多个设备用于 SBD，请多次指定 `-d` 选项，例如：

```
root # sbd -d /dev/SBD1 -d /dev/SBD2 -d /dev/SBD3 create
```

3. 如果您的 SBD 设备驻留在多路径组上，请使用 `-l` 和 `-4` 选项来调整要用于 SBD 的超时。有关详细信息，请参见第 11.5 节“[超时计算](#)”。所有超时均以秒为单位指定：

```
root # sbd -d /dev/SBD -4 180 ① -l 90 ② create
```

- ① `-4` 选项用于指定 `msgwait` 超时。在以上示例中，超时设置为 180 秒。
- ② `-l` 选项用于指定 `watchdog` 超时。在以上示例中，超时设置为 90 秒。模拟检查包的最小允许值为 15 秒。

4. 检查已写入设备的内容：

```
root # sbd -d /dev/SBD dump
Header version      : 2.1
UUID                : 619127f4-0e06-434c-84a0-ea82036e144c
Number of slots     : 255
Sector size         : 512
Timeout (watchdog)  : 5
Timeout (allocate)  : 2
Timeout (loop)      : 1
```

```
Timeout (msgwait) : 10
==Header on disk /dev/SBD is dumped
```

正如您看到的，超时数也储存在报头中，以确保所有参与的节点在这方面都一致。

初始化 SBD 设备之后，编辑 SBD 配置文件，然后启用并启动相应的服务以让更改生效。

过程 11.4：编辑 SBD 配置文件

1. 打开文件 `/etc/sysconfig/sbd`。
2. 搜索以下参数：`SBD_DEVICE`。
该参数指定要监视和要用于交换 SBD 讯息的设备。
3. 编辑此行，并用您的 SBD 设备替换 `SBD`：

```
SBD_DEVICE="/dev/SBD"
```

如果您需要在第一行中指定多个设备，请使用分号分隔设备（设备顺序无关紧要）：

```
SBD_DEVICE="/dev/SBD1; /dev/SBD2; /dev/SBD3"
```

如果无法访问 SBD 设备，守护程序将无法启动群集，并会禁止群集启动。

4. 搜索以下参数：`SBD_DELAY_START`。
启用或禁用延迟。设置 `SBD_DELAY_START` 为 `yes`（如果 `msgwait` 相对较长，而群集节点引导很快）。将此参数设置为 `yes` 可在引导时延迟 SBD 启动。虚拟机有时候需要此项延迟。

将您的 SBD 设备添加到 SBD 配置文件之后，启用 SBD 守护程序。SBD 守护程序是群集堆栈的关键部分。当群集堆栈正在运行时，需要运行该守护程序。因此，每当启动 `pacemaker` 服务时，作为依赖项，也要启动 `sbd` 服务。

过程 11.5：启用和启动 SBD 服务

1. 在每个节点，启用 SBD 服务：

```
root # systemctl enable sbd
```

每当启动 Pacemaker 服务时，SBD 服务将与 Corosync 服务一起启动。

2. 在每个节点上重新启动群集堆栈:

```
root # crm cluster restart
```

此操作会自动触发 SBD 守护程序的启动。

下一步是测试 SBD 设备，请参见[过程 11.6](#)。

过程 11.6：测试 SBD 设备

1. 以下命令会将节点槽及其当前消息从 SBD 设备进行转储:

```
root # sbd -d /dev/SBD list
```

现在，您应该会看到曾随 SBD 启动的所有群集节点都列在此处。例如，如果您拥有双节点群集，讯息槽对于两个节点都应显示 `clear`：

```
0      alice      clear
1      bob        clear
```

2. 尝试将测试消息发送到节点之一:

```
root # sbd -d /dev/SBD message alice test
```

3. 此节点将在系统日志文件中确认收到了该讯息:

```
May 03 16:08:31 alice sbd[66139]: /dev/SBD: notice: servant: Received
command test from bob on disk /dev/SBD
```

这就确认了 SBD 确实在节点上正常运行，并已准备好接收消息。

在最后一步中，您需要调整群集配置，请参见[过程 11.7](#)。

过程 11.7：将群集配置为使用 SBD

要配置在群集中使用 SBD，您需要在群集配置中执行以下操作：

- 将 `stonith-timeout` 参数设置为与您的设置相匹配的值。
- 配置 SBD STONITH 资源。

有关 `stonith-timeout` 的计算，请参见[第 11.5 节“超时计算”](#)。

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 运行 `crm configure`。
3. 输入以下内容：

```
crm(live)configure# property stonith-enabled="true" ❶  
crm(live)configure# property stonith-watchdog-timeout=0 ❷  
crm(live)configure# property stonith-timeout="40s" ❸
```

- ❶ 此为默认配置，因为不支持没有 STONITH 的群集。而如果出于测试目的停用了 STONITH，请确保再次将此参数设置为 `true`。
 - ❷ 如果未显式设置，此值默认为 `0`，适用于 SBD 与一到三个设备搭配使用的情况。
 - ❸ 如果将 SBD 的 `msgwait` 超时值设置为 `30` 秒，则适合将 `stonith-timeout` 值设置为 `40`。
4. 对于双节点群集，请确定您希望延迟是可预测的还是随机的。对于其他类型的群集，则不需要设置此参数。

可预测的静态延迟

此参数会在执行 STONITH 操作之前启用静态延迟。它可以确保在使用不同的屏蔽资源和不同的延迟值时，节点不会互相屏蔽。在发生“屏蔽资源争用”时，目标节点将会松脱。使用该参数可将特定的节点“标记”为在双节点群集出现节点分裂情况时能够幸存。要达到此目的，为每个节点创建两个原始 STONITH 设备至关重要。在以下配置中，如果出现节点分裂情况，alice 将会获胜并得以幸存：

```
crm(live)configure# primitive st-sbd-alice stonith:external/sbd params \  
 \  
    pcmk_host_list=alice pcmk_delay_base=20  
crm(live)configure# primitive st-sbd-bob stonith:external/sbd params \  
    pcmk_host_list=bob pcmk_delay_base=0
```

动态随机延迟

此参数防止在使用慢速设备（例如 SBD）时出现双重屏蔽。它会为屏蔽设备上的 STONITH 操作添加随机延迟。这对于双节点群集尤其重要，否则若发生节点分裂的情况，两个节点可能都会尝试相互屏蔽。

```
crm(live)configure# primitive stonith_sbd stonith:external/sbd
params pcmk_delay_max=30
```

5. 使用 `show` 复查更改。
6. 使用 `commit` 提交更改，并使用 `exit` 退出 crm 当前配置。

启动资源之后，群集即会成功配置为使用 SBD。如果需要屏蔽某个节点，将会使用此方法。

11.8 设置无磁盘 SBD

SBD 可在无磁盘模式下操作。在此模式下，发生以下情况时，将使用检查包设备来重置节点：如果失去法定票数、如果任何受监视的守护程序丢失且未恢复，或者如果 Pacemaker 决定节点需要屏蔽。无磁盘 SBD 基于节点的“自我屏蔽”，具体取决于群集的状态、法定票数和一些合理的假设。CIB 中不需要 STONITH SBD 原始资源。

! 重要：群集节点数

不要将无磁盘 SBD 用作双节点群集的屏蔽机制。请仅在三节点或更多节点的群集中使用。无磁盘模式下的 SBD 无法处理双节点群集节点分裂情况。

过程 11.8：配置无磁盘 SBD

1. 打开文件 `/etc/sysconfig/sbd` 并使用以下项：

```
SBD_PACEMAKER=yes
SBD_STARTMODE=always
SBD_DELAY_START=no
SBD_WATCHDOG_DEV=/dev/watchdog
SBD_WATCHDOG_TIMEOUT=5
```

此 `SBD_DEVICE` 项不需要，因为未使用共享磁盘。此参数缺失时，`sbd` 服务不会为 SBD 设备启动任何观察程序进程。

2. 在每个节点，启用 SBD 服务：

```
root # systemctl enable sbd
```

每当启动 Pacemaker 服务时，SBD 服务将与 Corosync 服务一起启动。

3. 在每个节点上重新启动群集堆栈：

```
root # crm cluster restart
```

此操作会自动触发 SBD 守护程序的启动。

4. 检查参数 `have-watchdog=true` 是否已自动设置：

```
root # crm configure show | grep have-watchdog
have-watchdog=true
```

5. 运行 `crm configure` 并在 `crm` 外壳上设置以下群集属性：

```
crm(live)configure# property stonith-enabled="true" ①
crm(live)configure# property stonith-watchdog-timeout=10 ②
```

- ① 此为默认配置，因为不支持没有 STONITH 的群集。而如果出于测试目的停用了 STONITH，请确保再次将此参数设置为 `true`。
- ② 对于无磁盘 SBD，此参数不能为零。它定义了经过多长时间之后可以假定屏蔽目标已自我屏蔽。因此，其值必须大于等于 `SBD_WATCHDOG_TIMEOUT`（`/etc/sysconfig/sbd` 中）的值。从 SUSE Linux Enterprise High Availability Extension 15 开始，如果您将 `stonith-watchdog-timeout` 设置为负值，Pacemaker 将自动计算此超时并将它设置为 `SBD_WATCHDOG_TIMEOUT` 值的两倍。

6. 使用 `show` 复查更改。

7. 使用 `commit` 提交更改，并使用 `exit` 退出 `crm` 当前配置。

11.9 测试 SBD 和屏蔽

要测试 SBD 在节点屏蔽方面是否按预期工作，请使用以下其中一种或所有方法：

手动触发节点屏蔽

要针对节点 `NODENAME` 触发屏蔽操作，请执行以下操作：

```
root # crm node fence NODENAME
```

检查经过以下时间之后，节点是否已屏蔽，以及其他节点是否将该节点视为已屏蔽：
`stonith-watchdog-timeout`。

模拟 SBD 失败

1. 识别 SBD inquisitor 的进程 ID：

```
root # systemctl status sbd
● sbd.service - Shared-storage based fencing daemon

   Loaded: loaded (/usr/lib/systemd/system/sbd.service; enabled;
   vendor preset: disabled)
   Active: active (running) since Tue 2018-04-17 15:24:51 CEST; 6 days
   ago
     Docs: man:sbd(8)
   Process: 1844 ExecStart=/usr/sbin/sbd $SBD_OPTS -p /var/run/sbd.pid
   watch (code=exited, status=0/SUCCESS)
  Main PID: 1859 (sbd)
    Tasks: 4 (limit: 4915)
   CGroup: /system.slice/sbd.service
          └─1859 sbd: inquisitor

[...]
```

2. 通过终止 SBD inquisitor 进程模拟 SBD 失败。在我们的示例中，SBD inquisitor 的进程 ID 是 `1859`：

```
root # kill -9 1859
```

节点主动自我屏蔽。经过以下时间之后，其他节点注意到该节点丢失并将它视为已自我屏蔽：`stonith-watchdog-timeout`。

通过监视操作失败触发屏蔽

对于正常配置，资源停止操作失败将触发屏蔽。要手动触发屏蔽，可以产生一个资源停止操作失败。或者，可以临时更改资源监视操作的配置，产生监视失败，如下所示：

1. 配置资源监视操作的 `on-fail=fence` 属性：

```
op monitor interval=10 on-fail=fence
```

2. 让监视操作失败（例如，如果资源与某个服务相关，则可通过终止相应的守护程序来实现）。
此失败会触发屏蔽操作。

11.10 储存保护的其他机制

除了通过 STONITH 进行节点屏蔽之外，还可使用其他方法在资源级别实现储存保护。例如，SCSI-3 和 SCSI-4 使用永久保留，而 `sfex` 提供锁定机制。这两种方法将在下面的小节中介绍。

11.10.1 配置 `sg_persist` 资源

SCSI 规范 3 和 4 定义了永久保留。其属于 SCSI 协议功能，可用于 I/O 屏蔽和故障转移。此功能在 `sg_persist` Linux 命令中实施。



注意：SCSI 磁盘兼容性

用于 `sg_persist` 的所有后备磁盘都必须与 SCSI 磁盘兼容。`sg_persist` 仅适用于 SCSI 磁盘或 iSCSI LUN 等设备。不要将它用于 IDE、SATA 或不支持 SCSI 协议的任何块设备。

继续之前，请检查您的磁盘是否支持永久保留。使用以下命令（用您的设备名称替换 `DISK`）：

```
root # sg_persist -n --in --read-reservation -d /dev/DISK
```

结果显示您的磁盘是否支持永久保留：

- 支持的磁盘：

```
PR generation=0x0, there is NO reservation held
```

- 不支持的磁盘:

```
PR in (Read reservation): command not supported  
Illegal request, Invalid opcode
```

如果您收到错误讯息（如上面所示），请用 SCSI 兼容的磁盘替换旧磁盘。否则请执行如下操作：

1. 要创建原始资源 `sg_persist`，请以 `root` 身份运行以下命令：

```
root # crm configure  
crm(live)configure# primitive sg sg_persist \  
    params devs="/dev/sdc" reservation_type=3 \  
    op monitor interval=60 timeout=60
```

2. 将 `sg_persist` 原始资源添加到主从组：

```
crm(live)configure# ms ms-sg sg \  
    meta master-max=1 notify=true
```

3. 执行一些测试。当资源处于主/从状态时，您可以在运行主实例的群集节点上的 `/dev/sdc1` 中进行装入和写入，但无法在运行从实例的群集节点上进行写入。

4. 为 Ext4 添加文件系统原始资源：

```
crm(live)configure# primitive ext4 ocf:heartbeat:Filesystem \  
    params device="/dev/sdc1" directory="/mnt/ext4" fstype=ext4
```

5. 在 `sg_persist` 主资源和文件系统资源之间添加以下顺序关系和并置：

```
crm(live)configure# order o-ms-sg-before-ext4 Mandatory: ms-sg:promote  
    ext4:start  
crm(live)configure# colocation col-ext4-with-sg-persist Mandatory: ext4 ms-  
    sg:Master
```

6. 使用 `show` 命令检查所有更改。

7. 提交更改。

有关详细信息，请参见 [sg_persist](#) 手册页。

11.10.2 使用 sfex 确保排它激活储存

此部分将介绍另一种低级别机制：[sfex](#)，可将共享储存区的访问以排它的方式锁定于一个节点。请注意，sfex 不会替代 STONITH。由于 sfex 需要共享储存，因此建议将上述 SBD 节点屏蔽机制用于储存的另一个分区。

按照设计，sfex 不能与需要并发的负载（例如 OCFS2）配合使用。其可作为传统故障转移型工作负载的一层保护。实际效果与 SCSI-2 保留类似，但更具一般性。

11.10.2.1 概述

在共享储存环境中，储存区的一个小分区专门设置为储存一个或多个锁。

在获取受保护资源之前，节点必须先获取保护锁。此顺序由 Pacemaker 强制实施。sfex 组件可确保即使 Pacemaker 遇到了节点分裂情况，也不会被多次授予锁。

这些锁必须定期刷新，这样某个节点的终止才不会永久性地阻止此锁，其他节点仍可继续操作。

11.10.2.2 设置

以下内容可帮助您了解如何创建用于 sfex 的共享分区以及如何为 CIB 中的 sfex 锁配置资源。单个 sfex 分区可存放任意数量的锁，并需要为每个锁分配 1 KB 的储存空间。默认情况下，[sfex_init](#) 将在分区上创建一个锁。

重要：要求

- sfex 的共享分区应和要保护的数据位于同一逻辑单元上。
- 共享的 sfex 分区不得使用基于主机的 RAID 或 DRBD。
- 可以使用 LVM2 逻辑卷。

过程 11.9：创建 SFEX 分区

1. 创建用于 sfex 的共享分区。注意此分区的名称，并用它替代下面的 `/dev/sfex`。
2. 使用以下命令创建 sfex 元数据：

```
root # sfex_init -n 1 /dev/sfex
```

3. 校验元数据是否正确创建：

```
root # sfex_stat -i 1 /dev/sfex ; echo $?
```

此操作应返回 `2`，因为当前未保存锁。

过程 11.10：为 SFEX 锁配置资源

1. sfex 锁通过 CIB 中的资源表示，其配置如下：

```
crm(live)configure# primitive sfex_1 ocf:heartbeat:sfex \  
# params device="/dev/sfex" index="1" collision_timeout="1" \  
    lock_timeout="70" monitor_interval="10" \  
# op monitor interval="10s" timeout="30s" on-fail="fence"
```

2. 要通过 sfex 锁保护资源，请在要保护的资源和 sfex 资源之间创建强制顺序和放置约束。如果要保护的资源 ID 是 `filesystem1`：

```
crm(live)configure# order order-sfex-1 Mandatory: sfex_1 filesystem1  
crm(live)configure# colocation col-sfex-1 Mandatory: filesystem1 sfex_1
```

3. 如果使用组语法，请将 sfex 资源添加为组内的第一个资源：

```
crm(live)configure# group LAMP sfex_1 filesystem1 apache ipaddr
```

11.11 更多信息

- [man sbd](#)
- http://www.linux-ha.org/wiki/SBD_Fencing

12 访问控制列表

crm 外壳 (crmsh) 或 Hawk2 等群集管理工具可由 `root` 用户或 `haclient` 组内的任何用户使用。默认情况下，这些用户具有完全读/写访问权。要限制访问权或指派更加细化的访问权限，可以使用访问控制列表 (ACL)。

访问控制列表由一组有序的访问规则构成。每个规则针对一部分群集配置赋予用户读取或写入访问权限，或拒绝其访问。规则通常会组合在一起产生特定角色，然后可以为用户指派与其任务匹配的角色。



注意：CIB 语法验证版本和 ACL 的差异

仅当您的 CIB 是使用 `pacemaker-2.0` 或更高 CIB 语法版本验证的情况下，此 ACL 文档才适用。有关如何查验这一点以及升级 CIB 版本的细节，请参见[注意：升级 CIB 语法版本](#)。

12.1 要求和先决条件

开始对群集使用 ACL 之前，确保满足了以下条件：

- 请使用 NIS、Active Directory 或者通过手动方式将相同用户添加到所有节点，来确保群集中所有节点上的用户一致。
- 您要使用 ACL 修改其访问权限的所有用户都必须属于 `haclient` 组。
- 所有用户都需要使用 crmsh 的绝对路径 `/usr/sbin/crm` 运行 crmsh。
- 如果非特权用户想要运行 crmsh，则需要使用 `/usr/sbin` 扩展其 `PATH` 变量。

! 重要：默认访问权限

- ACL 是可选功能。默认情况下，ACL 处于禁用状态。
- 如果未启用 ACL，则 `root` 用户以及属于 `haclient` 组的所有用户都将拥有对群集配置的完全读/写访问权。
- 即使启用并配置了 ACL，`root` 和默认 CRM 所有者 `hacluster` 也始终对群集配置拥有完全访问权。

要使用 ACL，需要具备一些关于 XPath 的知识。XPath 是在 XML 文档中选择节点所用的语言。请参见 <http://en.wikipedia.org/wiki/XPath> 或查找位于 <http://www.w3.org/TR/xpath/> 的规范。

12.2 在群集中启用 ACL

在开始配置 ACL 之前，需要先启用 ACL。要执行此操作，请在 `crmsd` 中使用以下命令：

```
root # crm configure property enable-acl=true
```

或者，按过程 12.1 “使用 Hawk 启用 ACL” 中所述使用 Hawk2。

过程 12.1：使用 HAWK 启用 ACL

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 在左侧导航栏中，选择群集配置显示全局群集选项及它们当前的值。

3. 在群集配置下面，单击空下拉框并选择 `enable-acl` 以添加该参数。系统即会添加该参数，且将其设为默认值 `No`。

4. 将其值设置为 `Yes`，然后应用更改。

12.3 ACL 的基本原理

访问控制列表由一组有序的访问规则构成。每个规则针对一部分群集配置赋予用户读取或写入访问权限，或拒绝其访问。规则通常会组合在一起产生特定角色，然后可以为用户指派与其任务匹配的角色。ACL 角色是用于描述对 CIB 访问权限的一组规则。规则包括以下组成部分：

- 访问权限，例如 读、写 或 拒绝
- 规则应用位置的规范。此规范可以是类型、ID 参照或 XPath 表达式。

通常，方便的做法是在角色中捆绑 ACL 并将特定角色指派给系统用户（ACL 目标）。创建 ACL 规则的方法有两种：

- 第 12.3.1 节 “通过 XPath 表达式设置 ACL 规则”。需要知道基础 XML 的结构才能创建 ACL 规则。
- 第 12.3.2 节 “通过缩写设置 ACL 规则”。创建速记语法和 ACL 规则以应用到匹配的对象。

12.3.1 通过 XPath 表达式设置 ACL 规则

要通过 XPath 管理 ACL 规则，需要知道基础 XML 的结构。可使用以下命令来检索结构（该命令将显示 XML 格式的群集配置，请参见例 12.1）：

```
root # crm configure show xml
```

例 12.1：XML 格式群集配置摘录

```
<num_updates="59"  
  dc-uuid="175704363"  
  crm_feature_set="3.0.9"  
  validate-with="pacemaker-2.0"  
  epoch="96"  
  admin_epoch="0"  
  cib-last-written="Fri Aug 8 13:47:28 2014"  
  have-quorum="1">  
<configuration>  
  <crm_config>  
    <cluster_property_set id="cib-bootstrap-options">
```

```

    <nvpair name="stonith-enabled" value="true" id="cib-bootstrap-options-
stonith-enabled"/>
    [...]
  </cluster_property_set>
</crm_config>
</nodes>
  <node id="175704363" uname="alice"/>
  <node id="175704619" uname="bob"/>
</nodes>
<resources> [...] </resources>
<constraints/>
<rsc_defaults> [...] </rsc_defaults>
<op_defaults> [...] </op_defaults>
<configuration>
</cib>

```

使用 XPath 语言，您可在此 XML 文档中查找节点。例如，要选择 root 节点 (`cib`)，则使用 XPath 表达式 `/cib`。要查找全局群集配置，则使用 XPath 表达式 `/cib/configuration/crm_config`。

例如，表 12.1 “Operator 角色 - 访问类型和 XPath 表达式” 显示了用于创建“操作员”角色的参数（访问类型和 XPath 表达式）。具有此角色的用户只能执行第二列中所述的任务 - 他们既不能重新配置任何资源（例如，更改参数或操作），也不能更改共置约束或顺序约束的配置。

表 12.1：OPERATOR 角色 - 访问类型和 XPATH 表达式

类型	XPath/说明
写	<pre>//crm_config// nvpair[@name='maintenance-mode']</pre> <p>打开或关闭群集维护模式。</p>
写	<pre>//op_defaults//nvpair[@name='record- pending']</pre> <p>选择是否记录待发操作。</p>
写	<pre>//nodes/node//nvpair[@name='standby']</pre>

类型	XPath/说明
	将节点设置为联机或备用模式。
写	<pre>//resources//nvpair[@name='target-role']</pre> <p>启动、停止任何资源或使资源升级或降级。</p>
写	<pre>//resources//nvpair[@name='maintenance']</pre> <p>选择是否应将资源置于维护模式。</p>
写	<pre>//constraints/rsc_location</pre> <p>将资源从一个节点迁移/移动到另一个节点。</p>
读	<pre>/cib</pre> <p>查看群集的状态。</p>

12.3.2 通过缩写设置 ACL 规则

对于不想使用 XML 结构的用户，有一种更简单的方法。

例如，请考虑以下 XPath：

```
//*[@id="rsc1"]
```

它会查找 ID 为 `rsc1` 的所有 XML 节点。

缩写语法与以下内容类似：

```
ref:"rsc1"
```

这同样适用于约束。这是详细的 XPath：

```
//constraints/rsc_location
```

缩写语法与以下内容类似：

```
type:"rsc_location"
```

可以在 `crmsh` 和 `Hawk2` 中使用缩写语法。CIB 守护程序知道如何将 ACL 规则应用到匹配的对象。

12.4 使用 Hawk 配置 ACL2

下面的过程说明如何通过定义 `monitor` 角色并将其指派给用户来配置对群集配置的只读访问权限。您也可以根据过程 12.4 “使用 `crmsh` 添加 Monitor 角色并指派用户” 中所述使用 `crmsh` 来实现此目的。

过程 12.2：使用 HAWK2 添加 MONITOR 角色

1. 登录 Hawk2:

```
https://HAWKSERVER:7630/
```

2. 在左侧导航栏中，选择角色。
3. 单击创建。
4. 输入唯一的角色 ID，例如 `monitor`。
5. 为访问权限选择 `读取`。
6. 为 Xpath 输入 Xpath 表达式 `/cib`。

ACL 角色

ACL 角色是一组描述对 CIB 的访问权限的规则。每个规则包含：

- 一个访问规则 (`读取`、`写入`或`拒绝`)
- 指定对何处应用规则的说明 (XPath 表达式、类型或 ID 参考)

创建角色

角色 ID： 定义一个唯一的 ID。

权限： 选择访问权限 (读取/写入/拒绝)

XPath： 针对要应用该访问权限的 CIB 元素输入 XPath 表达式 (例如，输入 `//constraints/rsc_location` 以将其应用于位置约束)。

类型： 输入要应用该访问权限的 CIB XML 元素的名称 (例如，输入 `rsc_location` 以将其应用于位置约束)。

参考： 输入要应用该访问权限的这类 CIB XML 元素的 ID (例如，输入 `rsc1` 以将其应用于 ID 为 `rsc1` 的所有 XML 元素)。

7. 单击创建。

如此即会创建名为 `monitor` 的新角色，为其设置 `读取` 权限，并通过使用 XPath 表达式 `/cib` 将其应用到 CIB 中的所有元素。

8. 如果需要，请通过单击加号图标并指定相应参数添加更多规则。

9. 使用向上或向下箭头按钮对各规则排序。

过程 12.3：使用 HAWK2 向目标指派角色

要向系统用户（即目标）指派我们在过程 12.2 中创建的角色，请执行以下操作继续：

1. 登录 Hawk2：

```
https://HAWKSERVER:7630/
```

2. 在左侧导航栏中，选择目标。

3. 要创建系统用户（即 ACL 目标），请单击创建，然后输入一个唯一的 `目标 ID`，例如 `tux`。确保此用户属于 `haclient` 组。

4. 要向目标指派角色，请选择一个或多个角色。

在本示例中，请选择您在过程 12.2 中创建的 `monitor` 角色。



5. 确认您的选择。

要配置资源或约束的访问权限，还可使用第 12.3.2 节“通过缩写设置 ACL 规则”中所述的缩写语法。

12.5 使用 crmsh 配置 ACL

以下过程说明如何通过定义 `monitor` 角色并将其指派给用户，来配置对群集配置的只读访问权。

过程 12.4：使用 CRMSH 添加 MONITOR 角色并指派用户

1. 以 `root` 身份登录。
2. 启动 `crmsh` 的交互模式：

```
root # crm configure
crm(live)configure#
```

3. 定义 ACL 角色：
 - a. 使用 `role` 命令定义新角色：

```
crm(live)configure# role monitor read xpath="/cib"
```

上面的命令会创建名为 `monitor` 的新角色，为其设置 `读` 权限并通过使用 XPath 表达式 `/cib` 将其应用到 CIB 中的所有元素。如有必要，可添加更多访问权限和 XPath 自变量。

- b. 根据需要添加其他角色。
4. 将角色指派给一个或多个 ACL 目标，即相应的系统用户。确保这些目标属于 `haclient` 组。

```
crm(live)configure# acl_target tux monitor
```

5. 检查更改：

```
crm(live)configure# show
```

6. 提交更改:

```
crm(live)configure# commit
```

要配置资源或约束的访问权限，还可使用第 12.3.2 节“通过缩写设置 ACL 规则”中所述的缩写语法。

13 网络设备绑定

对于许多系统，需要实施高于典型以太网设备的标准数据安全性或可用性要求的网络连接。在这些情况下，可以将多个以太网设备聚合到单个绑定设备。

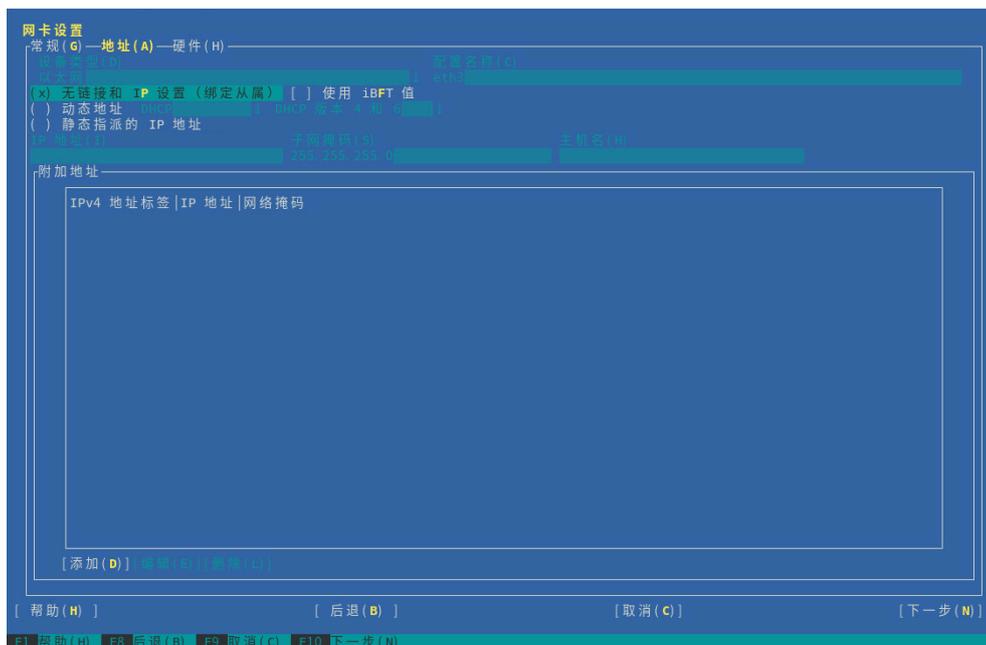
联接设备的配置通过联接模块选项来完成。其行为取决于联接设备的模式。默认情况下是 `mode=active-backup`，即如果活动从属设备发生故障，则其他从属设备将变成为活动设备。

使用 Corosync 时，绑定设备不受群集软件的管理。因此，必须在每个可能需要访问联接设备的群集节点上配置联接设备。

13.1 使用 YaST 配置联接设备

要配置联接设备，您必须有多个可以聚合到单独一个联接设备的以太网设备。按如下所示继续：

1. 以 `root` 身份启动 YaST 并选择系统 > 网络设置。
2. 在网络设置中，切换到概述选项卡以显示可用的设备。
3. 检查要聚合到联接设备的以太网设备是否有指定的 IP 地址。如果有，更改此地址：
 - a. 选择要更改的设备，然后单击编辑。
 - b. 在打开的网卡设置对话框的地址选项卡中，选择无链接和 IP 设置（绑定从属）选项。



c. 单击下一步，返回到网络设置对话框中的概述选项卡。

4. 添加新联接设备：

- a. 单击添加并将设备类型更改为绑定。单击下一步继续。
- b. 选择如何为绑定设备指派 IP 地址。有三种方法可供选择：
 - 无链接和 IP 设置（绑定从属）
 - 动态地址（使用 DHCP 或 Zeroconf）
 - 静态指派的 IP 地址

使用最适合您环境的方法。如果 Corosync 管理虚拟 IP 地址，请选择静态指派 IP 地址，并为接口指派一个 IP 地址。

- c. 切换到绑定从属选项卡。
- d. 它会显示在步骤 3.b 中已配置为绑定从属的所有以太网设备。要选择您想包含到绑定中的以太网设备，请在绑定从属和顺序下方激活相应设备前面的复选框。



e. 编辑联接驱动程序选项。可以使用以下模式：

balance-rr

提供负载均衡和容错，但会使包传输变得混乱无序。这可能会导致 TCP 重组等操作出现延迟。

active-backup

提供容错。

balance-xor

提供负载均衡和容错。

broadcast

提供容错。

802.3ad

提供动态链接集合（如果连接的交换机支持动态链接集合）。

balance-tlb

为外发的通讯量提供负载均衡。

balance-alb

为进来的和外发的通讯量提供负载均衡（如果使用的网络设备允许在使用中修改网络设备的硬件地址）。

f. 务必向联接驱动程序选项添加参数 `miimon=100`。如果不指定此参数，则不会定期检查链路，因此，绑定驱动程序可能会持续在有故障的链路上丢包。

5. 单击下一步，将 YaST 保留为确定，完成联接设备的配置。YaST 将配置写入 `/etc/sysconfig/network/ifcfg-bondDEVICENUMBER`。

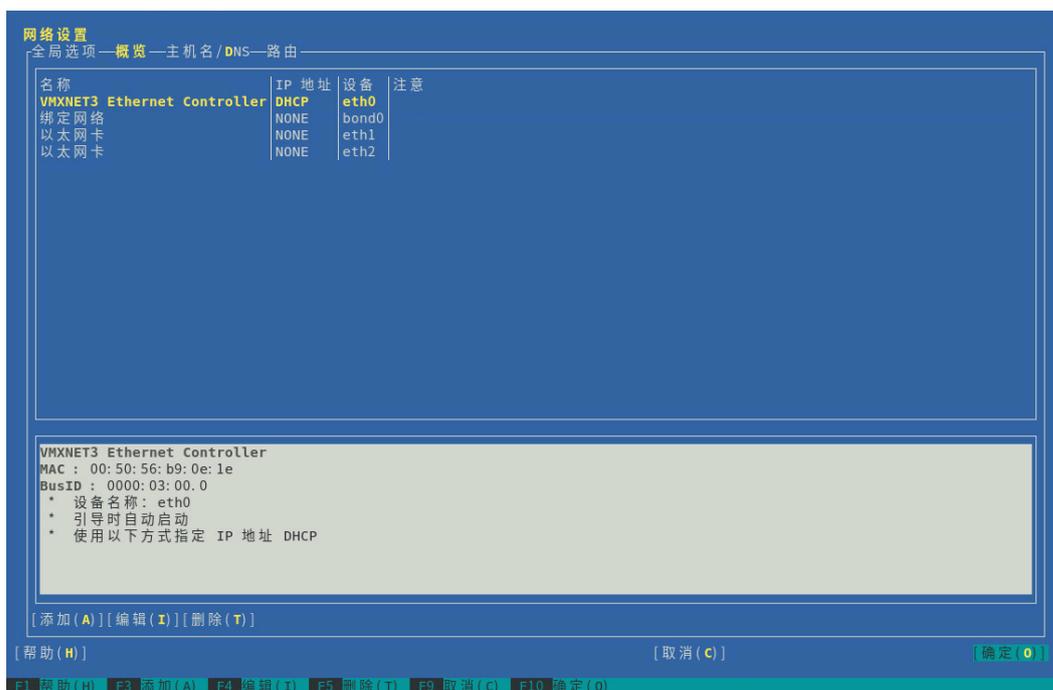
13.2 联接从属的热插拔

在某些情况下，例如相应的网络设备不断出现故障时，必须使用另一个接口来取代绑定的从属接口。解决方案是为联接从属设置热插拔，此外还需要更改 `udev` 规则，以便按总线 ID（而非 MAC 地址）匹配该设备。这样，有缺陷的硬件（同一槽内具有不同 MAC 地址的网卡）允许更换的话，您便可以更换该硬件。

过程 13.1：使用 YAST 为绑定从属配置热插拔

如果您想改用手动配置，请参见《SUSE Linux Enterprise High Availability Extension 管理指南》的“基本联网知识”一章中的“绑定从属的热插拔”一节。

1. 以 `root` 身份启动 YaST 并选择系统 > 网络设置。
2. 在网络设置中，切换到概述选项卡以显示已配置的设备。如果已配置绑定从属，则会显示在备注栏中。



3. 对于已经聚合到联接设备的每个以太网设备，请执行以下步骤：

- a. 选择要更改的设备，然后单击编辑。网卡设置对话框随即打开。
- b. 切换到常规选项卡，确保将激活设备设置为 在热插拔时。
- c. 切换到硬件选项卡。
- d. 针对 Udev 规则，单击更改并选择 BusID 选项。
- e. 单击确定和下一步，返回到网络设置对话框中的概述选项卡。如果您现在单击以太网设备项，底部窗格会显示设备的详细信息，包括总线 ID。

4. 单击确定确认您的更改，并退出网络设置。

在引导时，网络设置不会等待热插拔从属接口就绪，而是等待绑定就绪，而这至少需要有一个从属接口可用。当从系统中去除一个从属接口时（从 NIC 驱动程序拆开绑定、执行 NIC 驱动程序程序的 `rmmod` 命令或 PCI 热插拔去除为 true），内核会自动从绑定中将其去除。当向系统添加新网卡时（替换插槽中的硬件），`udev` 会应用基于总线的永久名称规则对新网卡进行重命名，并为其调用 `ifup`。`ifup` 命令会自动调用以将新网卡加入联接。

13.3 更多信息

《Linux Ethernet Bonding Driver HOWTO》（Linux 以太网绑定驱动程序操作指南）中详细介绍了所有模式和许多选项。安装包 `kernel-source` 后，您可以在 `/usr/src/linux/Documentation/networking/bonding.txt` 下找到该文件。

对于高可用性设置，本指南中所述的以下选项特别重要：`miimon` 和 `use_carrier`。

14 负载均衡

在负载均衡的情况下，服务器群集对于外部客户端而言就如同是一台大型的快速服务器。这种看上去像是单台服务器的服务器被称为虚拟服务器。它包括一个或多个用于调度进来的请求的负载均衡器，以及若干台运行实际服务的真实服务器。完成 High Availability Extension 的负载均衡设置后，您就可以构建高度可缩放且高度可用的网络服务，例如 Web、超速缓存、邮件、FTP、媒体和 VoIP 服务。

14.1 概念概述

High Availability Extension 支持两种负载均衡技术：Linux 虚拟服务器 (LVS) 和 HAProxy。两者的主要差别在于，Linux 虚拟服务器在 OSI 第 4 层（传输层）上运行，可配置内核的网络层，而 HAProxy 在第 7 层（应用层）上的用户空间中运行。因此，Linux 虚拟服务器所需的资源更少但处理的负载更多，而 HAProxy 可以检查流量，执行 SSL 终止，并根据流量内容做出调度决策。

另一方面，Linux 虚拟服务器包含两个不同的软件：IPVS（IP 虚拟服务器）和 KTCPVS（内核 TCP 虚拟服务器）。IPVS 提供第 4 层负载均衡，而 KTCPVS 提供第 7 层负载均衡。

本章概述了负载均衡与高可用性结合使用的概念，然后简要介绍了 Linux 虚拟服务器和 HAProxy。最后，提供了其他阅读材料的链接。

真实的服务器和负载均衡器可通过高速 LAN 或地理位置分散的 WAN 互相连接。负载均衡器可将请求发送到不同的服务器。它们使群集的多个并行服务看似单个 IP 地址（虚拟 IP 地址，即 VIP）上的一个虚拟服务。发送请求时可使用 IP 负载均衡技术或应用程序级的负载均衡技术。系统的可伸缩性通过在群集中透明地添加或删除节点来实现。

高可用性通过检测节点或服务故障并相应地照常重配置整个虚拟服务器系统来实现。

有多种负载均衡策略。下面介绍适用于 Linux 虚拟服务器的一些第 4 层策略：

- **循环复用：** 最简单的策略就是将每个连接轮流定向到不同的地址。例如，某个 DNS 服务器的多个条目可能与某个给定主机名对应。使用 DNS 循环复用时，该 DNS 服务器将轮流返回所有这些条目。因此，不同的客户端将看到不同的地址。
- **选择“最佳”服务器：** 可以使用“第一台做出响应的服务器”或“负载最低的服务器”方法实现平衡，不过此方案存在一些弱点。

- **平衡每台服务器的连接数：** 用户与服务器之间的负载均衡器可以在多台服务器之间划分用户数。
- **Geo 位置：** 可以将客户端定向到附近的服务器。

下面介绍适用于 HAProxy 的一些第 7 层策略：

- **URI：** 检查 HTTP 内容并将流量发送到最适合此特定 URL 的服务器。
- **URL 参数, RDP Cookie：** 检查会话参数的 HTTP 内容（可能在 post 参数中，或者 RDP（远程桌面协议）会话 Cookie 中），并将流量发送到为此会话提供服务的服务器。

尽管存在一些重叠的情况，但在 LVS/[ipvsadm](#) 不足以满足要求的情况下，可以使用 HAProxy，反之亦然：

- **SSL 终止：** 前端负载均衡器可以处理 SSL 层。因此，云节点不需要访问 SSL 密钥，而可以利用负载均衡器中的 SSL 加速器。
- **应用级别：** HAProxy 在应用级别运行，因此，负载均衡决策受内容流的影响。这样，便可以基于 Cookie 和其他此类过滤器实现持久性。

另一方面，HAProxy 不能完全取代 LVS/[ipvsadm](#)：

- LVS 支持“直接路由”，在此模式下，负载均衡器只作用于入站流，而出站流量将直接路由到客户端。在非对称环境中，这有可能会大幅提高吞吐量。
- LVS 支持状态连接表复制（通过 [connttrackd](#)）。因此，负载均衡器能够实现对客户端和服务器透明的故障转移。

14.2 使用 Linux 虚拟服务器配置负载均衡

以下部分概述了 LVS 主要组件和概念。然后，将介绍如何在 High Availability Extension 上设置 Linux 虚拟服务器。

14.2.1 定向器

LVS 的主要组件是 `ip_vs`（也称 IPVS）内核代码。它是默认内核的一部分，在 Linux 内核中实施传输层负载平衡（第 4 层交换）。运行包含 IPVS 代码的 Linux 内核的节点称为定向器。控制器上运行的 IPVS 代码是 LVS 的基本功能。

当客户端连接到定向器时，进来的请求在所有群集节点间是负载平衡的：定向器使用一组能使 LVS 正常工作的修改过的路由规则，将包转发到真实服务器。例如，连接不会在定向器上发起或终止，它也不会发送确认。定向器相当于将包从最终用户转发到真实服务器（运行用于处理请求的应用程序的主机）的专用路由器。

14.2.2 用户空间控制器和守护程序

`ldirectord` 守护程序是一个用户空间守护程序，用于在 LVS 负载平衡虚拟服务器群集中管理 Linux 虚拟服务器并监视真实的服务器。配置文件（见下文）指定虚拟服务及其关联的真实服务器，并告知 `ldirectord` 如何将此服务器配置为 LVS 重定向器。守护程序初始化时，将为群集创建虚拟服务。

`ldirectord` 守护程序通过定期请求已知的 URL 并检查响应来监视真实服务器的运行状况。如果真实服务器发生故障，将从负载平衡器的可用服务器列表中删除它。如果服务监视程序检测到发生故障的服务器已恢复并重新运行，则它会将此服务器重新添加到可用服务器列表中。如果出现所有真实服务器关闭的情况，则可以指定一台将 Web 服务重定向到的回退服务器。备用服务器通常是本地主机，它会显示一个应急页面，说明 Web 服务暂时不可用。

`ldirectord` 使用 `ipvsadm` 工具（包 `ipvsadm`）来操作 Linux 内核中的虚拟服务器表。

14.2.3 包的转发

定向器可采用三种不同方法将包从客户端发送到真实服务器：

网络地址转换 (NAT)

进来的请求到达虚拟 IP，然后通过将目标 IP 地址和端口更改为所选真实服务器的 IP 地址和端口将进来的请求转发到真实服务器。真实服务器向负载平衡器发送响应，负载平衡器再更改目标 IP 地址并将响应发回客户端。因此，最终用户就能从预期的源收到回复了。由于所有通讯都要流经负载平衡器，它通常会成为群集的瓶颈。

IP 隧道通讯进程 (IP-IP 封装)

IP 隧道通讯进程允许将发送到某个 IP 地址的包重定向到可能处于其他网络上的另一个地址。LVS 通过 IP 隧道 (重定向到其他 IP 地址) 向真实服务器发送请求, 然后真实服务器使用自己的路由选择表直接回复到客户端。群集成员可以处于不同的子网中。

直接路由选择

来自最终用户的包将直接转发到真实服务器。IP 包未经修改, 因此必须配置真实服务器以接受虚拟服务器 IP 地址的通讯。来自真实服务器的响应会直接发送到客户端。真实服务器和负载均衡器需处于同一物理网络段中。

14.2.4 调度算法

确定将哪台真实服务器用于客户端请求的新连接, 是使用不同算法来实现的。这些算法以模块的形式提供, 可进行调整以适应特定需要。有关可用模块的概述, 请参见 [ipvsadm\(8\)](#) 手册页。从客户端接收到连接请求时, 控制器会根据日程表将一台真实的服务器指派给此客户端。调度程序是 IPVS 内核代码中的一部分, 它决定哪台真实的服务器将获取下一个新连接。

有关 Linux 虚拟服务器调度算法的更详细说明, 请访问 <http://kb.linuxvirtualserver.org/wiki/IPVS>。此外, 还可以在 [ipvsadm](#) 手册页中搜索 `--scheduler`。

可以在 <http://www.haproxy.org/download/1.6/doc/configuration.txt> 上找到 HAProxy 的相关负载均衡策略。

14.2.5 使用 YaST 设置 IP 负载均衡

可使用 YaST IP 负载均衡模块配置基于内核的 IP 负载均衡。它是 [ldirectord](#) 的前端。

要访问“IP 负载均衡”对话框, 请以 `root` 用户身份启动 YaST 并选择 High Availability > IP 负载均衡。或者, 以 `root` 用户身份使用 `yast2 iplb` 从命令行启动 YaST 群集模块。

默认安装不包括配置文件 `/etc/ha.d/ldirectord.cf`。此文件由 YaST 模块创建。YaST 模块中的可用选项卡对应于 `/etc/ha.d/ldirectord.cf` 配置文件的结构, 此配置文件用于定义全局选项和虚拟服务的选项。

有关配置示例以及负载均衡器和真实服务器之间产生的进程, 请参见例 14.1 “简单的 [ldirectord](#) 配置”。



注意：全局参数和虚拟服务器参数

如果在虚拟服务器部分和全局部分都指定了某个参数，那么在虚拟服务器部分中定义的值将覆盖在全局部分中定义的值。

过程 14.1：配置全局参数

以下过程描述了如何配置最重要的全局参数。有关个别参数（以及此处未提及的参数）的更多细节，请单击[帮助](#)或参见 [ldirectord](#) 手册页。

1. 通过检查间隔定义 [ldirectord](#) 连接到每台真实服务器以检查它们是否仍处于联机状态的间隔。
2. 通过检查超时设置真实服务器应该在多长时间内响应上次检查。
3. 通过失败计数定义在检查被视为失败前 [ldirectord](#) 可尝试向真实服务器发送多少次请求。
4. 通过协商超时定义协商检查经过多少秒后应视为超时。
5. 在回退中，输入当所有真实服务器都停机时，要将 Web 服务重定向到的 Web 服务器的主机名或 IP 地址。
6. 如果希望系统在任何真实服务器的连接状态发生改变时均发送警报，请在电子邮件警报中输入有效的电子邮件地址。
7. 通过电子邮件警报频率定义经过多少秒后，如果任何真实服务器仍无法访问，应重复发出电子邮件警报。
8. 在电子邮件警报状态中指定应发送电子邮件警报的服务器状态。如果要定义多个状态，请使用逗号分隔的列表。
9. 通过自动重新装载定义 [ldirectord](#) 是否应连续监视配置文件有无修改。如果设置为 [yes](#)，则将在发生更改时自动重新装载配置。
10. 通过 Quiescent 开关定义是否应从内核的 LVS 表中去除发生故障的真实服务器。如果设置为是，则不会删除发生故障的服务器。而是将其权重设置为 [0](#)，表示不接受新连接。已建立的连接将继续存在，直到超时为止。
11. 如果要使用备用路径进行日志记录，请在日志文件中指定日志文件的路径。默认情况下，[ldirectord](#) 会将其日志文件写入 [/var/log/ldirectord.log](#)。

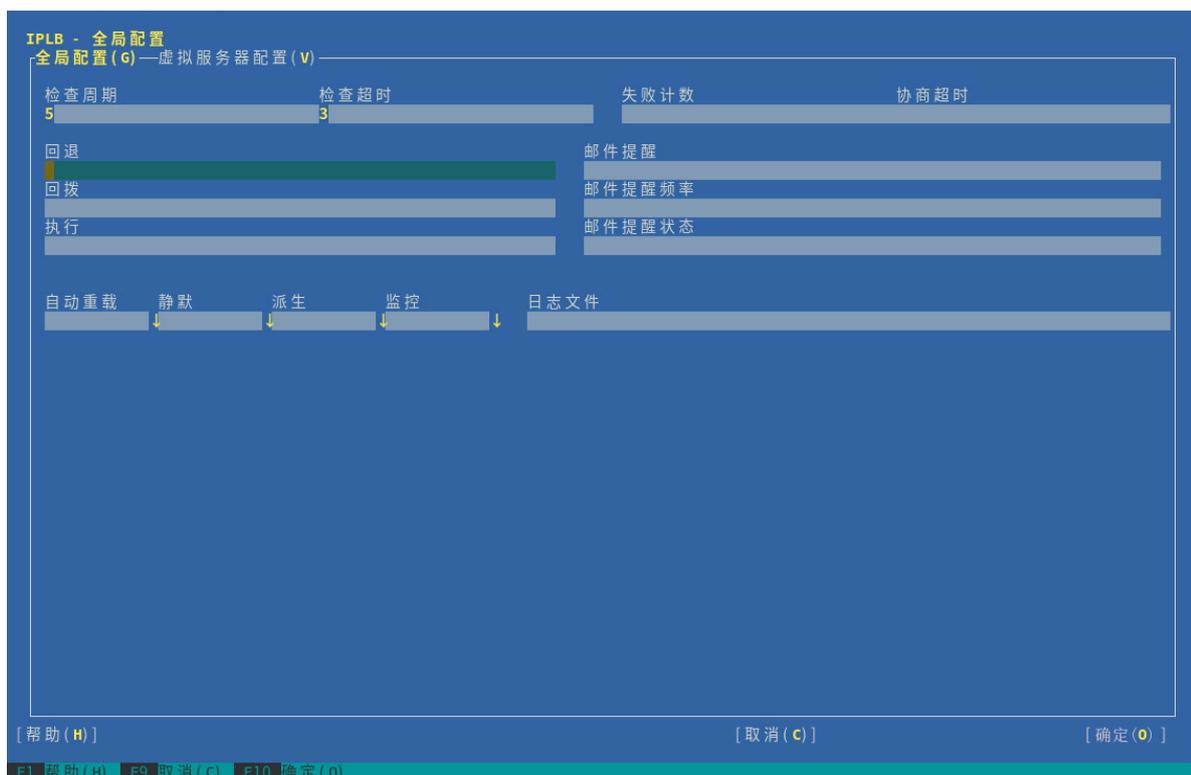


图 14.1：YAST IP 负载均衡 - 全局参数

过程 14.2：配置虚拟服务

可通过为每种虚拟服务定义若干参数来配置一个或多个虚拟服务。以下过程描述了如何配置虚拟服务最重要的参数。有关个别参数（以及此处未提及的参数）的更多细节，请单击帮助或参见 [ldirectord](#) 手册页。

1. 在 YaST IP 负载均衡模块中，切换到虚拟服务器配置选项卡。
2. 添加新虚拟服务器或编辑现有虚拟服务器。一个新对话框将显示可用选项。
3. 在虚拟服务器中，输入负载均衡器和真实服务器可作为 LVS 访问的共享虚拟 IP 地址（IPv4 或 IPv6）和端口。还可以指定主机名和服务来代替 IP 地址和端口号。或者，也可以使用防火墙标记。防火墙标记是一种将任意 `VIP:port` 服务的集合聚合到一个虚拟服务中的方法。
4. 要指定真实服务器，需要输入服务器的 IP 地址（IPv4、IPv6 或主机名）、端口（或服务名称）以及转发方法。转发方法必须是 `gate`、`ipip` 或 `masq` 中的一种，请参见第 14.2.3 节“包的转发”。

单击添加按钮，为每台真实服务器输入需要的自变量。

5. 在检查类型中选择用于测试真实服务器是否仍然活动的检查类型。例如，要发送请求并检查响应是否包含预期的字符串，请选择 Negotiate。
6. 如果已将检查类型设置为 Negotiate，则还需定义要监视的服务类型。从服务下拉框中进行选择。
7. 在请求中输入检查间隔期间每台真实服务器上所请求对象的 URI。
8. 如果要检查来自真实服务器的响应是否包含特定字符串（如 “I'm alive” 消息），请定义需要匹配的正则表达式。将正则表达式输入到接收中。如果来自真实服务器的响应包含此表达式，则认为真实服务器处于活动状态。
9. 根据您在 [步骤 6](#) 中选定的服务类型，您还必须为身份验证指定更多参数。切换到授权类型选项卡，输入登录、口令、数据库或机密等详细信息。有关更多信息，请参见 YaST 帮助文本或 [ldirectord](#) 手册页。
10. 切换至其他选项卡。
11. 选择用于负载均衡的调度程序。有关可用调度程序的信息，请参见 [ipvsadm\(8\)](#) 手册页。
12. 选择要使用的协议。如果将虚拟服务指定为 IP 地址和端口，则它必须是 tcp 或 udp。如果将虚拟服务指定为防火墙标记，则协议必须是 fwm。
13. 如果需要，可定义其他参数。单击确定确认配置。YaST 会将此配置写入 /etc/ha.d/ldirectord.cf。

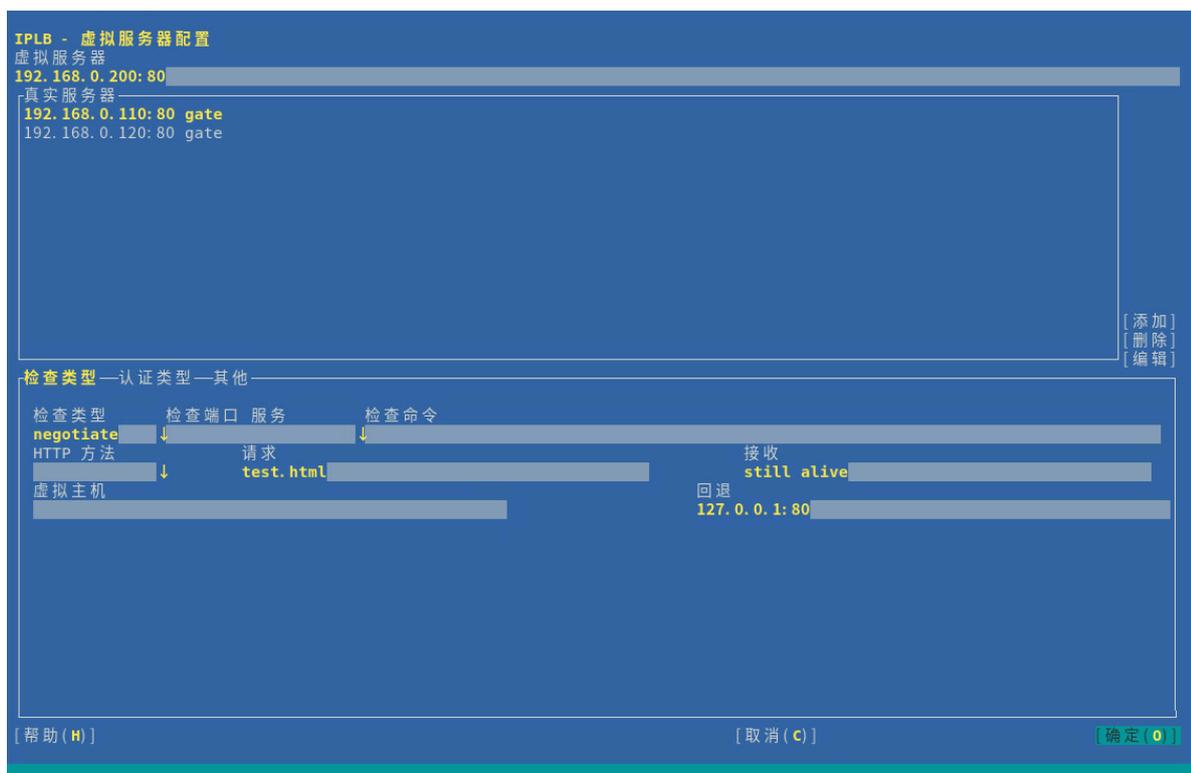


图 14.2：YAST IP 负载均衡 - 虚拟服务

例 14.1：简单的 LDIRECTORD 配置

图 14.1 “YaST IP 负载均衡 - 全局参数”和图 14.2 “YaST IP 负载均衡 - 虚拟服务”中所示的值将生成在 `/etc/ha.d/ldirectord.cf` 中定义的以下配置：

```

autoreload = yes ①
  checkinterval = 5 ②
  checktimeout = 3 ③
  quiescent = yes ④
  virtual = 192.168.0.200:80 ⑤
  checktype = negotiate ⑥
  fallback = 127.0.0.1:80 ⑦
  protocol = tcp ⑧
  real = 192.168.0.110:80 gate ⑨
  real = 192.168.0.120:80 gate ⑨
  receive = "still alive" ⑩
  request = "test.html" ⑪
  scheduler = wlc ⑫
  service = http ⑬

```

- ① 定义 `ldirectord` 应连续检查配置文件有无修改。
- ② `ldirectord` 连接到每台真实服务器以检查它们是否仍处于联机状态的间隔。
- ③ 真实服务器必须在上次检查后多长时间内作出响应。
- ④ 定义不要从内核的 LVS 表中删除发生故障的真实服务器，而是将其权重设置为 `0`。
- ⑤ LVS 的虚拟 IP 地址 (VIP)。可通过端口 `80` 访问 LVS。
- ⑥ 用于测试真实服务器是否仍处于活动状态的检查类型。
- ⑦ 此服务的所有真实服务器都宕机时，要将 Web 服务重定向到的服务器。
- ⑧ 要使用的协议。
- ⑨ 定义了两台真实服务器，均可通过端口 `80` 访问。包的转发方法是 `gate`，表示使用直接路由选择。
- ⑩ 需要在真实服务器的响应字符串中匹配的正则表达式。
- ⑪ 检查间隔期间每台真实服务器上所请求对象的 URI。
- ⑫ 用于负载均衡的所选调度程序。
- ⑬ 要监视的服务类型。

此配置将导致以下进程流：`ldirectord` 每 5 秒连接一次每台真实服务器

(②)，并按 ⑨ 和 ⑪ 中指定的方式请求 `192.168.0.110:80/test.html` 或 `192.168.0.120:80/test.html`。如果上次检查后 3 秒内 (③) 未收到来自真实服务器的预期的 `still alive` 字符串 (⑩)，它会将此真实服务器从可用池中删除。但是，由于 `quiescent=yes` 设置 (④)，真实服务器不会从 LVS 表中去除。其权重将设置为 `0`，这样就不会接受到此真实服务器的新连接请求。已建立的连接将继续存在，直到超时为止。

14.2.6 其他设置

除了使用 YaST 配置 `ldirectord` 外，还需要确保满足以下条件，才能完成 LVS 设置：

- 正确设置真实服务器以提供所需服务。
- 负载均衡服务器（或服务器）必须能够使用 IP 转发将通讯路由到真实服务器。真实服务器的网络配置取决于选择的包转发方法。

- 为避免负载均衡服务器（或服务器）成为整个系统的单个故障点，需要设置负载均衡器的一个或多个备份。在群集配置中配置 `ldirectord` 的原始资源，以便在发生硬件故障时，`ldirectord` 可以故障转移到其他服务器。
- 由于负载均衡器的备份也需要 `ldirectord` 配置文件才能完成其任务，因此请确保 `/etc/ha.d/ldirectord.cf` 在要用作负载均衡器备份的所有服务器上都可用。可以按第 4.5 节“将配置传送到所有节点”中所述使用 `Csync2` 同步配置文件。

14.3 使用 HAProxy 配置负载均衡

以下章节概述了 HAProxy 以及如何针对高可用性进行设置。负载均衡器会将所有请求分发到其后端服务器。它采用主动/被动配置，也就是说，当主节点发生故障时，从节点就会变成主节点。在这种情况下，用户察觉不到任何服务中断的迹象。

在本节中，我们将使用以下设置：

- IP 地址为 `192.168.1.99` 的负载均衡器。
- 一个虚拟浮动 IP 地址 `192.168.1.99`。
- 我们的服务器（通常用于 Web 内容）`www.example1.com`（IP: `192.168.1.200`）和 `www.example2.com`（IP: `192.168.1.201`）

要配置 HAProxy，请使用以下过程：

1. 安装 `haproxy` 包中提供。
2. 创建包含以下内容的 `/etc/haproxy/haproxy.cfg` 文件：

```
global ①
  maxconn 256
  daemon

defaults ②
  log      global
  mode     http
  option   httplog
  option   dontlognull
```

```

retries 3
option redispatch
maxconn 2000
timeout connect 5000 ③
timeout client 50s ④
timeout server 50000 ⑤

frontend LB
bind 192.168.1.99:80 ⑥
reqadd X-Forwarded-Proto:\ http
default_backend LB

backend LB
mode http
stats enable
stats hide-version
stats uri /stats
stats realm Haproxy\ Statistics
stats auth haproxy:password ⑦
balance roundrobin ⑧
option httpclose
option forwardfor
cookie LB insert
option httpchk GET /robots.txt HTTP/1.0
server web1-srv 192.168.1.200:80 cookie web1-srv check
server web2-srv 192.168.1.201:80 cookie web2-srv check

```

- ① 该部分包含进程范围的选项和特定于操作系统的选项。

maxconn

每个进程的最大并发连接数。

daemon

建议的模式，HAProxy 将在后台运行。

- ② 该部分为其声明后的所有其他部分设置默认参数。一些重要的行：

redispatch

启用或禁用在连接失败时重新分发会话。

log

启用事件和流量日志记录。

mode http

以 HTTP 模式运行（针对 HAProxy 建议采用的模式）。在此模式下，将会先分析请求，然后再执行与任何服务器的连接。不符合 RFC 要求的请求将被拒绝。

option forwardfor

将 HTTP X-Forwarded-For 报头添加到请求中。如果您想要保留客户端的 IP 地址，则需要使用此选项。

- ③ 与服务器建立连接的尝试成功之前可等待的最长时间。
- ④ 客户端可保持非活动状态的最长时间。
- ⑤ 服务器端可保持非活动状态的最长时间。
- ⑥ 该部分将前端部分和后端部分合并在一起。

balance leastconn

定义负载均衡算法，请参见 <http://cbonte.github.io/haproxy-dconv/configuration-1.5.html#4-balance>。

stats enable ,

stats auth

启用统计报告（通过 stats enable）。auth 选项记录针对特定帐户的身份验证的统计信息。

- ⑦ HAProxy 统计数字报告页面的身份凭证。
- ⑧ 负载均衡将在轮询过程中工作。

3. 测试配置文件：

```
root # haproxy -f /etc/haproxy/haproxy.cfg -c
```

4. 将以下行添加到 Csync2 的配置文件 /etc/csync2/csync2.cfg 中，以确保包括 HAProxy 配置文件：

```
include /etc/haproxy/haproxy.cfg
```

5. 同步该文件:

```
root # csync2 -f /etc/haproxy/haproxy.cfg
root # csync2 -xv
```

注意

Csync2 配置部分假设 HA 节点是使用 [ha-cluster-bootstrap](#) 配置的。有关细节, 请参见《安装和设置快速入门》。

6. 确保在两个负载均衡器 (alice 和 bob) 上禁用 HAProxy, 因为它由 Pacemaker 启动:

```
root # systemctl disable haproxy
```

7. 配置新的 CIB:

```
root # crm configure
crm(live)# cib new haproxy-config
crm(haproxy-config)# primitive haproxy systemd:haproxy \
  op start timeout=120 interval=0 \
  op stop timeout=120 interval=0 \
  op monitor timeout=100 interval=5s \
  meta target-role=Started
crm(haproxy-config)# primitive vip IPAddr2 \
  params ip=192.168.1.99 nic=eth0 cidr_netmask=23 broadcast=192.168.1.255
  \
  op monitor interval=5s timeout=120 on-fail=restart
crm(haproxy-config)# group g-haproxy vip haproxy
```

8. 校验新 CIB 并更正任何错误:

```
crm(haproxy-config)# verify
```

9. 提交新的 CIB:

```
crm(haproxy-config)# cib use live
crm(live)# cib commit haproxy-config
```

14.4 更多信息

- <http://www.haproxy.org> 
- 项目主页 <http://www.linuxvirtualserver.org/> 。
- 有关 ldirectord 的更多信息，请参见其综合性手册页。
- LVS 知识库: http://kb.linuxvirtualserver.org/wiki/Main_Page 

15 Geo 群集（多站点群集）

除本地群集和城域群集外，SUSE® Linux Enterprise High Availability Extension 15 SP2 还支持地理位置分散的群集（Geo 群集，有时也称为多站点群集）。这意味着，每个本地群集可以有多个地域分散的站点。这些群集之间的故障转移由更高级的实体、所谓的 投票间 进行协调。有关如何使用和设置 Geo 群集的详细信息，请参见《Geo 群集快速入门》文章和《Geo 群集指南》。

16 执行维护任务

要在群集节点上执行维护任务，可能需要停止该节点上运行的资源、移动这些资源，或者关闭或重引导该节点。此外，可能还需要暂时接管群集中资源的控制权，甚至需要在资源仍在运行时停止群集服务。

本章介绍如何在不产生负面影响的情况下手动关闭群集节点。此外，本章将会概述群集堆栈提供的用于执行维护任务的不同选项。

16.1 关闭群集节点所造成的影响

关闭或重引导某个群集节点（或停止节点上的 Pacemaker 服务）时，将触发以下过程：

- 该节点上运行的资源将会停止，或被移出该节点。
- 如果停止资源的操作失败或超时，STONITH 机制将屏蔽该节点并将其关闭。

过程 16.1：手动重引导群集节点

如果您的目的是先按顺序将服务移出节点，然后再关闭或重引导该节点，请执行以下操作：

1. 在要重引导或关闭的节点上，以 `root` 或同等的身份登录。
2. 将节点置于 `standby` 模式：

```
root # crm -w node standby
```

如此即可将服务迁移出节点，而不会受限于 Pacemaker 的关闭超时。

3. 使用以下命令检查群集状态：

```
root # crm status
```

此命令显示相关节点处于 `standby` 模式：

```
[...]
```

```
Node bob: standby
[...]
```

4. 停止该节点上的 Pacemaker 服务：

```
root # crm cluster stop
```

5. 重引导该节点。

要再次检查节点是否已加入群集，请执行以下操作：

1. 以 `root` 或同等身份登录到该节点。
2. 检查 Pacemaker 服务是否已启动：

```
root # crm cluster status
```

3. 如果未启动，请将其启动：

```
root # crm cluster start
```

4. 使用以下命令检查群集状态：

```
root # crm status
```

此命令应该会显示节点已重新联机。

16.2 用于维护任务的不同选项

Pacemaker 提供了各种选项用于执行系统维护：

将群集置于维护模式

使用全局群集属性 `maintenance-mode` 可以一次性将所有资源置于维护状态。群集将停止监视这些资源，因此不知道它们的状态。

将节点置于维护模式

此选项可以一次性将特定节点上运行的所有资源置于维护状态。群集将停止监视这些资源，因此不知道它们的状态。

将节点置于待机模式

处于待机模式的节点不再能够运行资源。该节点上运行的所有资源将被移出或停止（如果没有其他节点可用于运行资源）。另外，该节点上的所有监视操作将会停止（设置了 `role="Stopped"` 的操作除外）。

如果您需要停止群集中的某个节点，同时继续提供另一个节点上运行的服务，则可以使用此选项。

将资源置于维护模式

为某个资源启用此模式后，将不会针对该资源触发监视操作。

如果您需要手动调整此资源所管理的服务，并且不希望群集在此期间对该资源运行任何监视操作，则可以使用此选项。

将资源置于不受管理模式

使用 `is-managed` 元属性可以暂时“释放”某个资源，使其不受群集堆栈的管理。这意味着，您可以手动调整此资源管理的服务（例如，调整任何组件）。不过，群集将继续监视该资源，并继续报告任何错误。

如果您希望群集同时停止监视该资源，请改为使用按资源维护模式（请参见[将资源置于维护模式](#)）。

16.3 准备和完成维护工作



警告：数据丢失风险

如果您需要执行测试或维护工作，请执行下面的一般步骤。

如果不执行，有可能会产生不利的负面影响，例如，资源不按顺序启动、CIB 在群集节点之间不同步，甚至丢失数据。

1. 在开始之前，请选择第 16.2 节中所述的适合您情况的选项。
2. 请使用 Hawk2 或 crmsh 应用此选项。
3. 执行维护任务或测试。
4. 完成后，请将资源、节点或群集恢复“正常”工作状态。

16.4 将群集置于维护模式

要在 `crm` 外壳中将群集置于维护模式，请使用以下命令：

```
root # crm configure property maintenance-mode=true
```

要在完成维护工作后将群集恢复正常模式，请使用以下命令：

```
root # crm configure property maintenance-mode=false
```

过程 16.2：使用 HAWK2 将群集置于维护模式

1. 按第 7.2 节“登录”中所述，启动 Web 浏览器并登录到群集。
2. 在左侧导航栏中，选择群集配置。
3. 在 CRM 配置组中，从空下拉框中选择 `maintenance-mode` 属性，然后单击加号图标添加该属性。
4. 要设置 `maintenance-mode`，请选中 `maintenance-mode` 旁边的复选框，并确认您所做的更改。
5. 完成整个群集的维护任务之后，取消选中 `maintenance-mode` 属性旁边的复选框。此后，High Availability Extension 将再次接管群集管理工作。

16.5 将节点置于维护模式

要在 `crm` 外壳中将节点置于维护模式，请使用以下命令：

```
root # crm node maintenance NODENAME
```

要在完成维护工作后将节点恢复正常模式，请使用以下命令：

```
root # crm node ready NODENAME
```

过程 16.3：使用 HAWK2 将节点置于维护模式

1. 按第 7.2 节“登录”中所述，启动 Web 浏览器并登录到群集。

2. 在左侧导航栏中，选择群集状态。
3. 在其中某个节点视图中，单击节点旁边的扳手图标，然后选择维护。
4. 完成维护任务后，单击节点旁边的扳手图标，然后选择就绪。

16.6 将节点置于待机模式

要在 crm 外壳中将节点置于待机模式，请使用以下命令：

```
root # crm node standby NODENAME
```

要在完成维护工作后将节点恢复联机状态，请使用以下命令：

```
root # crm node online NODENAME
```

过程 16.4：使用 HAWK2 将节点置于待机模式

1. 按第 7.2 节“登录”中所述，启动 Web 浏览器并登录到群集。
2. 在左侧导航栏中，选择群集状态。
3. 在其中某个节点的视图中，单击节点旁边的扳手图标，然后选择待机。
4. 完成节点的维护任务。
5. 要停用待机模式，请单击节点旁边的扳手图标，然后选择就绪。

16.7 将资源置于维护模式

要在 crm 外壳中将资源置于维护模式，请使用以下命令：

```
root # crm resource maintenance RESOURCE_ID true
```

要在完成维护工作后将资源恢复正常模式，请使用以下命令：

```
root # crm resource maintenance RESOURCE_ID false
```

过程 16.5：使用 HAWK2 将资源置于维护模式

1. 按第 7.2 节“登录”中所述，启动 Web 浏览器并登录到群集。
2. 在左侧导航栏中，选择资源。
3. 选择要置于维护模式或不受管理模式的资源，单击该资源旁边的扳手图标，然后选择编辑资源。
4. 打开元属性类别。
5. 从空下拉列表中，选择 maintenance 属性，然后单击加号图标添加该属性。
6. 选中 maintenance 旁边的复选框，以将 maintenance 属性设置为 yes。
7. 确认更改。
8. 完成该资源的维护任务之后，取消选中该资源的 maintenance 属性旁边的复选框。此后，资源将再次由 High Availability Extension 软件管理。

16.8 将资源置于不受管理模式

要在 crm 外壳中将资源置于不受管理模式，请使用以下命令：

```
root # crm resource unmanage RESOURCE_ID
```

要在完成维护工作后再次将资源置于受管模式，请使用以下命令：

```
root # crm resource manage RESOURCE_ID
```

过程 16.6：使用 HAWK2 将资源置于不受管理模式

1. 按第 7.2 节“登录”中所述，启动 Web 浏览器并登录到群集。
2. 在左侧导航栏中选择状态，然后转到资源列表。
3. 在操作列中，单击要修改的资源旁边的向下箭头图标，然后选择编辑。资源配置屏幕即会打开。
4. 在元属性下面，从空下拉框中选择 is-managed 项。

5. 将其值设置为 No，然后单击应用。
6. 完成维护任务后，将 `is-managed` 设置为 Yes（默认值）并应用更改。
此后，资源将再次由 High Availability Extension 软件管理。

16.9 在维护模式下重引导群集节点



注意：含义

如果群集或某个节点处于维护模式，您可以根据需要停止或重启动群集资源 — High Availability Extension 不会尝试重启动它们。如果您停止节点上的 Pacemaker 服务，所有守护程序和进程（最初作为 Pacemaker 管理的群集资源启动）将继续运行。

如果您在群集或某个节点处于维护模式的情况下尝试启动该节点上的 Pacemaker 服务，Pacemaker 将针对每个资源启动一次性的监视操作（“探测”），以确定哪些资源当前正在该节点上运行。但是，它只会确定资源的状态，而不执行其他操作。

如果您要在群集或某个节点处于 维护模式 时关闭该节点，请执行以下操作：

1. 在要重引导或关闭的节点上，以 root 或同等的身份登录。
2. 如果您使用 DLM 资源（或依赖于 DLM 的其他资源），请确保在停止 Pacemaker 服务之前显式停止这些资源：

```
crm(live)resource# stop RESOURCE_ID
```

这是因为，停止 Pacemaker 也会停止 DLM 对其成员资格和讯息交换服务有依赖的 Corosync 服务。如果 Corosync 停止，DLM 资源将假设一种节点分裂情况并触发屏蔽操作。

3. 停止该节点上的 Pacemaker 服务：

```
root # crm cluster stop
```

4. 关闭或重引导节点。

III 储存和数据复制

- 17 分布式锁管理器 (DLM) 239
- 18 OCFS2 241
- 19 GFS2 250
- 20 DRBD 255
- 21 群集逻辑卷管理器 (群集 LVM) 273
- 22 群集多设备 (群集 MD) 288
- 23 Samba 群集 292
- 24 使用 Rear (Relax-and-Recover) 实现灾难恢复 302

17 分布式锁管理器 (DLM)

内核中的分布式锁管理器 (DLM) 是 OCFS2、GFS2、群集 MD 和群集 LVM (lvmlockd) 用于在每个相关层提供主动-主动储存的基础组件。

17.1 用于 DLM 通讯的协议

为了避免单一故障点，非常有必要对高可用性群集配置冗余通讯路径。对于 DLM 通讯也是如此。如果出于任何原因无法使用网络绑定（聚合控制协议，LACP），则我们强烈建议在 Corosync 中定义冗余通讯通道（另一个环）。有关详细信息，请参见[过程 4.3 “定义冗余通讯通道”](#)。

然后，DLM 将根据 `/etc/corosync/corosync.conf` 中的配置确定是要使用 TCP 还是 SCTP 协议来进行通讯：

- 如果 `rrp_mode` 设置为 `none`（表示禁用冗余环配置），DLM 会自动使用 TCP。但是，如果未定义冗余通讯通道，在 TCP 链路断开的情况下，DLM 通讯将会失败。
- 如果 `rrp_mode` 设置为 `passive`（典型的设置），并且在 `/etc/corosync/corosync.conf` 中正确配置了另一个通讯环，DLM 将自动使用 SCTP。在这种情况下，DLM 讯息交换将获得 SCTP 提供的冗余功能。

17.2 配置 DLM 群集资源

DLM 使用 Pacemaker 提供的并在用户空间中运行的群集成员资格服务。因此，DLM 需要配置为群集中每个节点上都存在的克隆资源。

注意：多种解决方案的 DLM 资源

由于 OCFS2、GFS2、群集 MD 和群集 LVM (lvmlockd) 全部使用 DLM，因此为 DLM 配置一个资源便已足够。由于 DLM 资源在群集中的所有节点上运行，因此它配置为克隆资源。

如果您的设置包含 OCFS2 和群集 LVM，则只需为 OCFS2 和群集 LVM 配置一个 DLM 资源就够了。

过程 17.1：配置 DLM 的基础组

这种配置由包含多个原始资源的基础组和一个基础克隆构成。之后，基本组和基本克隆便可用于各种方案（例如，用于 OCFS2 和群集 LVM）。您只需视需要使用相应的基元资源扩展基本组。由于基本组具有内部共置和顺序约束，您无需单独指定多个组、克隆及其依赖项，方便了总体设置。

在群集中的一个节点上执行以下步骤：

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 运行 `crm configure`。
3. 输入以下命令以创建 DLM 的基元资源：

```
crm(live)configure# primitive dlm ocf:pacemaker:controld \  
op monitor interval="60" timeout="60"
```

4. 为 DLM 资源和其他储存相关的资源创建一个基础组：

```
crm(live)configure# group g-storage dlm
```

5. 克隆 `g-storage` 组，使它在所有节点上运行：

```
crm(live)configure# clone cl-storage g-storage \  
meta interleave=true target-role=Started
```

6. 使用 `show` 复查更改。
7. 如果全部都正确，请使用 `commit` 提交更改，并使用 `exit` 退出 `crm` 当前配置。



注意：禁用 STONITH 时出错

未启用 STONITH 的群集不受支持。如果您出于测试或查错目的将全局群集选项 `stonith-enabled` 设置为 `false`，则 DLM 资源以及依赖于它的所有服务（例如群集 LVM、GFS2 和 OCFS2）将无法启动。

18 OCFS2

Oracle Cluster File System 2 (OCFS2) 是一个自 Linux 2.6 内核以来完全集成的通用日记文件系统。OCFS2 可将应用程序二进制文件、数据文件和数据库储存到共享储存设备。群集中的所有节点对文件系统都有并行的读和写权限。用户空间控制守护程序（通过克隆资源管理）提供与 HA 堆栈的集成，尤其是与 Corosync 和分布式锁管理器 (DLM) 的集成。

18.1 功能和优点

OCFS2 可用于以下储存解决方案，例如：

- 一般应用程序和工作负荷。
- 群集中的 Xen 映像存储。Xen 虚拟机和虚拟服务器可存储于由群集服务器安装的 OCFS2 卷中。它提供了 Xen 虚拟机在各服务器之间的快速便捷的可移植性。
- LAMP (Linux、Apache、MySQL 和 PHP | Perl | Python) 堆栈。

作为高性能、非对称的并行群集文件系统，OCFS2 支持以下功能：

- 应用程序文件对群集中的所有节点均可用。用户只需在群集中的 OCFS2 上安装它一次。
- 所有节点都可以通过标准文件系统接口直接并行读写至储存区，从而方便地管理运行于群集上的应用程序。
- 文件访问通过 DLM 协调。DLM 控制在多数情况下都运行良好，但如果应用程序的设计与 DLM 争夺对文件访问的协调，则此设计可能会限制可伸缩性。
- 所有后端储存上都可以使用储存备份功能。可以方便地创建共享应用程序文件的图形，它能够帮助提供有效的故障恢复。

OCFS2 还提供以下功能：

- 元数据缓存。
- 元数据日记。

- 跨节点的文件数据一致性。
- 支持多达 4 KB 的多个块大小、多达 1 MB 的群集大小，最大卷大小为 4 PB (Petabyte)。
- 支持最多 32 个群集节点。
- 对于数据库文件的异步和直接 I/O 支持，提高了数据库性能。



注意：OCFS2 的支持

仅当与 SUSE Linux Enterprise High Availability Extension 提供的 pcmk (Pacemaker) 堆栈搭配使用时，SUSE 才支持 OCFS2。与 o2cb 堆栈结合使用时，SUSE 不提供对 OCFS2 的支持。

18.2 OCFS2 包和管理实用程序

OCFS2 内核模块 (`ocfs2`) 自动安装到 SUSE® Linux Enterprise Server 15 SP2 中的 High Availability Extension 上。要使用 OCFS2，请确保在群集中的每个节点上安装您内核适用的以下包：`ocfs2-tools` 及匹配的 `ocfs2-kmp-*` 包。

此 `ocfs2-tools` 包提供以下实用程序，用于管理 OFS2 卷。有关语法信息，请参见其手册页。

表 18.1：OCFS2 实用程序

OCFS2 实用程序	描述
<code>debugfs.ocfs2</code>	检查 OCFS 文件系统的状态以进行调试。
<code>fsck.ocfs2</code>	检查文件系统的错误并进行选择性的修改。
<code>mkfs.ocfs2</code>	在某个设备上创建 OCFS2 文件系统，通常是共享物理或逻辑磁盘上的某个分区。
<code>mounted.ocfs2</code>	检测并列出行群集系统上所有的 OCFS2 卷。检测并列出已经安装了 OCFS2 设备的系统上的所有节点或列出所有的 OCFS2 设备。

OCFS2 实用程序	描述
tunefs.ocfs2	更改 OCFS2 文件系统参数，包括卷标、节点槽号、所有节点槽的日志大小和卷大小。

18.3 配置 OCFS2 服务和 STONITH 资源

创建 OCFS2 卷之前，必须先将 DLM 和 STONITH 资源配置为群集中的服务。

以下过程使用 `crm` 外壳配置群集资源。或者，您也可以按第 18.6 节“使用 Hawk2 配置 OCFS2 资源”中所述，使用 Hawk2 来配置资源。

过程 18.1：配置 STONITH 资源



注意：所需的 STONITH 设备

您需要配置屏蔽设备。没有一个适当的 STONITH 机制（如 `external/sbd`），则配置会失败。

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 如过程 11.3 “初始化 SBD 设备”中所述，创建 SBD 分区。
3. 运行 `crm configure`。
4. 将 `external/sbd` 配置为屏蔽设备，并将 `/dev/sdb2` 作为存储检测信号和屏蔽的共享储存上的专用分区：

```
crm(live)configure# primitive sbd_stonith stonith:external/sbd \
  params pcmk_delay_max=30 meta target-role="Started"
```

5. 使用 `show` 复查更改。
6. 如果全部都正确，请使用 `commit` 提交更改，并使用 `exit` 退出 `crm` 当前配置。

有关为 DLM 配置资源组的细节，请参见过程 17.1 “配置 DLM 的基础组”。

18.4 创建 OCFS2 卷

根据第 18.3 节“配置 OCFS2 服务和 STONITH 资源”中所述配置 DLM 群集资源后，请将系统配置为使用 OCFS2，并创建 OCFS2 卷。

注意：适用于应用程序和数据文件的 OCFS2 卷

我们建议您通常将应用程序文件和数据文件储存在不同的 OCFS2 卷上。如果应用程序卷和数据卷具有不同的装入要求，则必须将它们储存在不同的卷上。

开始之前要准备计划用于 OCFS2 卷的块设备。将这些设备留作可用空间。

然后，按过程 18.2“创建并格式化 OCFS2 卷”中所述使用 `mkfs.ocfs2` 创建和格式化 OCFS2 卷。此命令最重要的参数列于表 18.2“重要的 OCFS2 参数”中。有关此命令的更多信息和命令语法，请参见 `mkfs.ocfs2` 手册页。

表 18.2：重要的 OCFS2 参数

OCFS2 参数	描述和建议
卷标 (<code>-L</code>)	卷的描述性名称能够在不同节点上安装卷时唯一标识它。使用 <code>tunefs.ocfs2</code> 实用程序根据需要修改该卷标。
群集大小 (<code>-C</code>)	群集大小是分配给文件以保存数据的最小空间单元。有关可用选项和推荐的信息，请参见 <code>mkfs.ocfs2</code> 手册页。
节点槽数 (<code>-N</code>)	可以同时安装卷的最大节点数。OCFS2 将为每个节点创建单独的系统文件（如日记）。访问卷的节点可以是小端体系结构（如 AMD64/Intel 64）和大端体系结构（如 S/390x）的组合。 节点特定的文件称为本地文件。节点槽号附加到该本地文件。例如： <code>journal:0000</code> 属于指派到槽号 <code>0</code> 的所有节点。

OCFS2 参数	描述和建议
	<p>根据预期有多少个节点并行装入卷，在创建卷时设置每个卷的最大节点槽数。使用 tunefs.ocfs2 实用程序根据需要增加节点槽数。请注意，此值不能减小。</p> <p>如果未指定 -N 参数，系统会根据文件系统的大小确定槽数。</p>
块大小 (-b)	<p>文件系统可寻址的最小空间单元创建卷时请指定块大小。有关可用选项和推荐的信息，请参见 mkfs.ocfs2 手册页。</p>
打开/关闭特定功能 (--fs-features)	<p>可以提供功能标志的逗号分隔列表，mkfs.ocfs2 会尝试根据此列表创建具有那些功能的文件系统。要打开某功能，请将其加入列表。要关闭某功能，请在其名称前加 no。</p> <p>有关所有可用标志的概述，请参见 mkfs.ocfs2 手册页。</p>
预定义功能 (--fs-feature-level)	<p>允许您从一组预定义文件系统功能中进行选择。有关可用选项的信息，请参见 mkfs.ocfs2 手册页。</p>

如果使用 **mkfs.ocfs2** 创建和格式化卷时未指定任何功能，则默认情况下将启用以下功能：**backup-super**、**sparse**、**inline-data**、**unwritten**、**metaecc**、**indexed-dirs** 和 **xattr**。

过程 18.2：创建并格式化 OCFS2 卷

只在群集节点之一上执行以下步骤。

1. 以 **root** 用户身份打开终端窗口并登录。
2. 使用命令 **crm status** 检查群集是否联机。

3. 使用 `mkfs.ocfs2` 实用程序创建并格式化卷。有关此命令语法的信息，请参见 [mkfs.ocfs2 手册页](#)。

例如，要在 `/dev/sdb1` 上创建最多支持 32 个群集节点的新 OCFS2 文件系统，请输入以下命令：

```
root # mkfs.ocfs2 -N 32 /dev/sdb1
```

18.5 装入 OCFS2 卷

可以手动装入 OCFS2 卷，也可以按[过程 18.4 “使用群集资源管理器装入 OCFS2 卷”](#)中所述使用群集管理器将其装入。

过程 18.3：手动装入 OCFS2 卷

1. 以 `root` 用户身份打开终端窗口并登录。
2. 使用命令 `crm status` 检查群集是否联机。
3. 使用 `mount` 命令从命令行装入卷。



警告：手动装入的 OCFS2 设备

如果为方便测试而手动装入了 OCFS2 文件系统，在开始将其作为群集资源使用前，请务必将其卸载恢复原状。

过程 18.4：使用群集资源管理器装入 OCFS2 卷

要使用 High Availability 软件装入 OCFS2 卷，请在群集中配置 `ocfs2` 文件系统资源。以下过程使用 `crm` 外壳配置群集资源。或者，您也可以按[第 18.6 节 “使用 Hawk2 配置 OCFS2 资源”](#)中所述，使用 Hawk2 来配置资源。

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 运行 `crm configure`。
3. 配置 Pacemaker 以在群集中的每个节点上装入 OCFS2 文件系统：

```
crm(live)configure# primitive ocfs2-1 ocf:heartbeat:Filesystem \
  params device="/dev/sdb1" directory="/mnt/shared" \
  fstype="ocfs2" options="acl" \
  op monitor interval="20" timeout="40" \
  op start timeout="60" op stop timeout="60" \
  meta target-role="Started"
```

4. 将 `ocfs2-1` 原始资源添加到您在过程 17.1 “配置 DLM 的基础组”中创建的 `g-storage` 组。

```
crm(live)configure# modgroup g-storage add ocfs2-1
```

`add` 子命令默认会追加新的组成员。由于基本组具有内部共置和顺序约束，Pacemaker 将仅在也已运行 `dlm` 资源的节点上启动 `ocfs2-1` 资源。

5. 使用 `show` 复查更改。
6. 如果全部都正确，请使用 `commit` 提交更改，并使用 `exit` 退出 `crm` 当前配置。

18.6 使用 Hawk2 配置 OCFS2 资源

除了使用 `crm` 外壳为 OCFS2 手动配置 DLM 和文件系统资源以外，您还可以使用 Hawk2 设置向导中的 OCFS2 模板。

! 重要：手动配置与使用 Hawk2 的差别

设置向导中的 OCFS2 模板不包括 STONITH 资源的配置。如果使用该向导，仍需在共享储存上创建 SBD 分区，并根据过程 18.1 “配置 STONITH 资源”中所述配置一个 STONITH 资源。

使用 Hawk2 设置向导中的 OCFS2 模板还会导致资源配置与过程 17.1 “配置 DLM 的基础组”和过程 18.4 “使用群集资源管理器装入 OCFS2 卷”中所述的手动配置略有不同。

过程 18.5：使用 HAWK2 的向导配置 OCFS2 资源

1. 登录 Hawk2:

<https://HAWKSERVER:7630/>

2. 在左侧导航栏中，选择向导。
3. 展开文件系统类别，然后选择 OCFS2 文件系统。
4. 按照屏幕指导执行操作。如果需要有关某个选项的信息，在 Hawk2 中单击它即可显示简短帮助文本。完成最后的配置步骤后，校验您所输入的值。
向导会显示将应用于 CIB 的配置代码片段以及任何其他更改（如果需要）。



5. 检查建议的更改。如果一切都符合您的预期，请应用更改。
如果操作成功，屏幕上会显示一条讯息。

18.7 在 OCFS2 文件系统上使用定额

要在 OCFS2 文件系统上使用定额，请分别使用相应的定额功能或装入选项创建和装入文件系统：ursquota（用于单独用户的定额）或 grpquota（用于组的定额）。这些功能也可以稍后使用 tunefs.ocfs2 在未装入的文件系统上启用。

文件系统启用了相应的定额功能后，它会跟踪其元数据，查看每个用户（或组）使用的空间和文件数。由于 OCFS2 将定额信息视为文件系统内部元数据，因此您无需运行 quotacheck (8) 程序。所有功能都内置到 fsck.ocfs2 和文件系统驱动程序中。

要实施强加于每个用户或组的限制，请如同在任何其他文件系统上一样运行 quotaon (8)。

由于性能原因，每个群集节点都会在本地图行定额会计，并每 10 秒将此信息与通用中央储存同步一次。此间隔可使用 `tunefs.ocfs2`、选项 `usrquota-sync-interval` 和 `grpquota-sync-interval` 进行调整。因此，定额信息可能不会始终都准确，因而在几个群集节点上并行操作时，用户或组可以稍微超出其定额限制。

18.8 更多信息

有关 OCFS2 的更多信息，请参见以下链接：

<https://ocfs2.wiki.kernel.org/> 

OCFS2 项目主页。

<http://oss.oracle.com/projects/ocfs2/> 

Oracle 上的原 OCFS2 项目主页。

<http://oss.oracle.com/projects/ocfs2/documentation> 

项目的原文档主页。

19 GFS2

全局文件系统 2 或称 GFS2 是适用于 Linux 计算机群集的共享磁盘文件系统。GFS2 允许所有节点直接同时访问同一个共享块储存。GFS2 不提供断开操作模式，也没有客户端角色或服务器角色。GFS2 群集中的所有节点以对等体的形式运行。GFS2 最多支持 32 个群集节点。在群集中使用 GFS2 需要通过硬件来访问共享储存，并需要通过一个锁管理器来控制对储存的访问。

如果性能是其中一个主要要求，SUSE 建议为群集环境使用 OCFS2 而不要使用 GFS2。我们的测试表明，与采用此设置的 GFS2 相比，OCFS2 的表现更好。

19.1 GFS2 包和管理实用程序

要使用 GFS2，请确保在群集的每个节点上安装您内核适用的 `gfs2-utils` 包和匹配的 `gfs2-kmp-*` 包。

此 `gfs2-utils` 包提供以下实用程序，用于管理 GFS2 卷。有关语法信息，请参见其手册页。

表 19.1：GFS2 实用程序

GFS2 实用程序	描述
<code>fsck.gfs2</code>	检查文件系统的错误并进行选择性的修改。
<code>gfs2_jadd</code>	将其他日记添加到 GFS2 文件系统。
<code>gfs2_grow</code>	产生 GFS2 文件系统。
<code>mkfs.gfs2</code>	在设备上创建 GFS2 文件系统，这通常是一个共享设备或分区。
<code>tunegfs2</code>	允许查看和处理 GFS2 文件系统参数，例如 <code>UUID</code> 、 <code>label</code> 、 <code>lockproto</code> 和 <code>locktable</code> 。

19.2 配置 GFS2 服务和 STONITH 资源

在创建 GFS2 卷之前，必须先配置 DLM 和 STONITH 资源。

过程 19.1：配置 STONITH 资源



注意：所需的 STONITH 设备

您需要配置屏蔽设备。没有一个适当的 STONITH 机制（如 `external/sbd`），则配置会失败。

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 如过程 11.3 “初始化 SBD 设备”中所述，创建 SBD 分区。
3. 运行 `crm configure`。
4. 将 `external/sbd` 配置为屏蔽设备，并将 `/dev/sdb2` 作为存储检测信号和屏蔽的共享储存上的专用分区：

```
crm(live)configure# primitive sbd_stonith stonith:external/sbd \  
  params pcmk_delay_max=30 meta target-role="Started"
```

5. 使用 `show` 复查更改。
6. 如果全部都正确，请使用 `commit` 提交更改，并使用 `exit` 退出 `crm` 当前配置。

有关为 DLM 配置资源组的细节，请参见过程 17.1 “配置 DLM 的基础组”。

19.3 创建 GFS2 卷

按第 19.2 节 “配置 GFS2 服务和 STONITH 资源”中所述将 DLM 配置为群集资源后，将系统配置为使用 GFS2 并创建 GFS2 卷。



注意：用于存储应用程序和数据文件的 GFS2 卷

我们建议您通常应将应用程序文件和数据文件储存在不同的 GFS2 卷上。如果应用程序卷和数据卷具有不同的装入要求，则必须将它们储存在不同的卷上。

开始之前要准备计划用于 GFS2 卷的块设备。将这些设备留作可用空间。

然后，按[过程 19.2 “创建并格式化 GFS2 卷”](#)中所述使用 `mkfs.gfs2` 创建并格式化 GFS2 卷。此命令最重要的参数列于[表 19.2 “重要的 GFS2 参数”](#)中。有关更多信息和命令语法，请参见 [mkfs.gfs2 手册页](#)。

表 19.2：重要的 GFS2 参数

GFS2 参数	描述和建议
锁定协议名称 (<code>-p</code>)	要使用的锁定协议的名称。可接受的锁定协议包括 <code>lock_dlm</code> （用于共享储存）；如果您将 GFS2 用作本地文件系统（只有 1 个节点），则可以指定 <code>lock_nolock</code> 协议。如果未指定此选项，假定采用 <code>lock_dlm</code> 协议。
锁定表名称 (<code>-t</code>)	与您正在使用的锁定模块对应的锁定表字段。该字段的格式为 <code>clustername:fsname</code> 。 <code>clustername</code> 必须与群集配置文件 <code>/etc/corosync/corosync.conf</code> 中的条目匹配。只允许此群集的成员使用此文件系统。 <code>fsname</code> 是唯一的文件系统名称（1 到 16 个字符），用于将此 GFS2 文件系统与创建的其他文件系统区分开来。
日记数量 (<code>-j</code>)	要为 <code>gfs2_mkfs</code> 创建的日记数量。要装入该文件系统的每台计算机至少需要一个日记。如果未指定此选项，将创建一个日记。

过程 19.2：创建并格式化 GFS2 卷

只在群集节点之一上执行以下步骤。

1. 以 `root` 用户身份打开终端窗口并登录。
2. 使用命令 `crm status` 检查群集是否联机。
3. 使用 `mkfs.gfs2` 实用程序创建并格式化卷。有关此命令的语法的信息，请参见 [mkfs.gfs2 手册页](#)。

例如，要在 `/dev/sdb1` 上创建最多支持 32 个群集节点的新的 GFS2 文件系统，请使用以下命令：

```
root # mkfs.gfs2 -t hacluster:mygfs2 -p lock_dlm -j 32 /dev/sdb1
```

`hacluster` 名称与文件 `/etc/corosync/corosync.conf` 中的条目 `cluster_name` 相关（这是默认设置）。

19.4 装入 GFS2 卷

可以手动装入 GFS2 卷，也可以按[过程 19.4 “使用群集管理器装入 GFS2 卷”](#)中所述使用群集管理器将其装入。

过程 19.3：手动装入 GFS2 卷

1. 以 `root` 用户身份打开终端窗口并登录。
2. 使用命令 `crm status` 检查群集是否联机。
3. 使用 `mount` 命令从命令行装入卷。



警告：手动装入的 GFS2 设备

如果因方便测试而手动装入了 GFS2 文件系统，在开始将其作为群集资源使用前，请务必再将其卸载。

过程 19.4：使用群集管理器装入 GFS2 卷

要使用高可用性软件装入 GFS2 卷，请在群集中配置 OCF 文件系统资源。以下过程使用 `crm` 外壳配置群集资源。或者，您可以使用 Hawk2 配置资源。

1. 启动壳层，并以 `root` 用户身份或同等身份登录。
2. 运行 `crm configure`。
3. 配置 Pacemaker 以在群集中的每个节点上装入 GFS2 文件系统：

```
crm(live)configure# primitive gfs2-1 ocf:heartbeat:Filesystem \
  params device="/dev/sdb1" directory="/mnt/shared" fstype="gfs2" \
  op monitor interval="20" timeout="40" \
  op start timeout="60" op stop timeout="60" \
  meta target-role="Stopped"
```

4. 创建一个基本组，该组包含您在过程 17.1 “配置 DLM 的基础组”中创建的 dlm 基元以及 gfs2-1 基元。克隆组：

```
crm(live)configure# group g-storage dlm gfs2-1
    clone cl-storage g-storage \
    meta interleave="true"
```

由于基本组具有内部共置和顺序约束，Pacemaker 将仅在同样已运行 dlm 资源的节点上启动 gfs2-1 资源。

5. 使用 show 复查更改。
6. 如果全部都正确，请使用 commit 提交更改，并使用 exit 退出 crm 当前配置。

20 DRBD

通过分布式复制块设备 (DRBD*)，您可以为位于 IP 网络上两个不同站点的两个块设备创建镜像。和 Corosync 一起使用时，DRBD 支持分布式高可用性 Linux 群集。本章说明如何安装和设置 DRBD。

20.1 概念概述

DRBD 以确保数据的两个副本保持相同的方式将主设备上的数据复制到次设备上。将其视为联网的 RAID 1。它实时对数据进行镜像，以便连续复制。应用程序不需要知道实际上它们的数据储存在不同的磁盘上。

DRBD 是 Linux 内核模块，位于下端的 I/O 调度程序与上端的文件系统之间，请参见图 20.1 “DRBD 在 Linux 中的位置”。要与 DRBD 通讯，用户需使用高级别命令 `drbdadm`。为了提供最大的灵活性，DRBD 附带了低级别工具 `drbdsetup`。

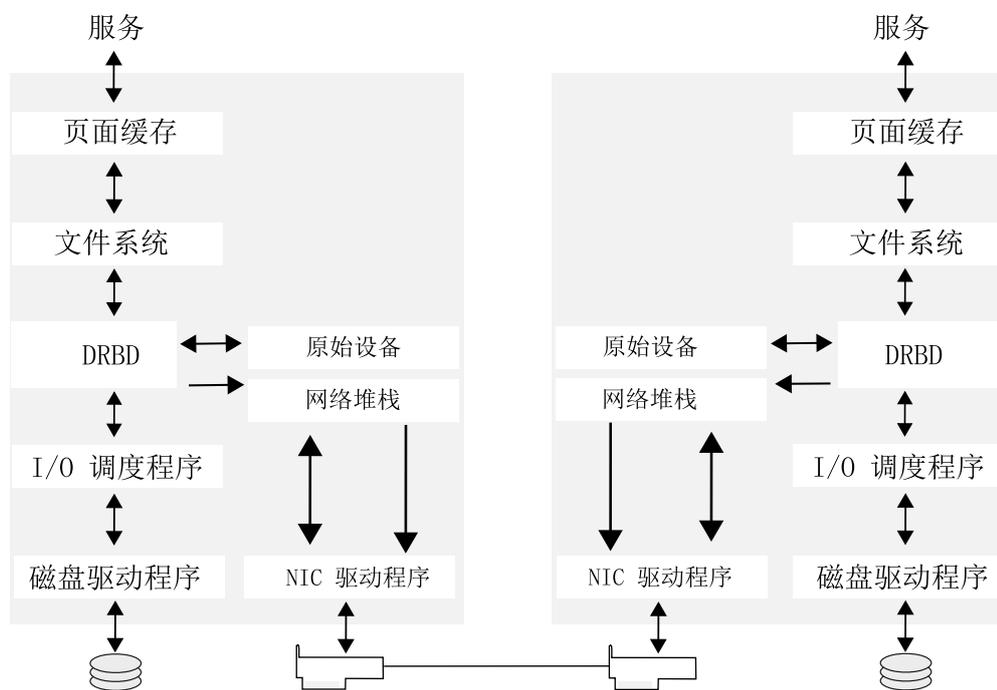


图 20.1：DRBD 在 LINUX 中的位置

! 重要：未加密数据

镜像之间的数据通讯是不加密的。为实现安全数据交换，您应为连接部署虚拟专用网 (VPN) 解决方案。

DRBD 允许使用 Linux 支持的任何块设备，通常包括：

- 硬盘分区或完整硬盘
- 软件 RAID
- 逻辑卷管理器 (LVM)
- 企业卷管理系统 (EVMS)

默认情况下，DRBD 使用 TCP 端口 7788 及更高端口进行 DRBD 节点间的通讯。请确保您的防火墙不会阻止所用端口上的通讯。

在 DRBD 设备上创建文件系统之前，必须先设置 DRBD 设备。与用户数据相关的所有操作都应通过 /dev/drbdN 设备单独执行，而不是在原始设备上执行，因为 DRBD 会将原始设备的最后一个部分用于元数据。使用原始设备会导致数据不一致。

借助 udev 集成，您还可以获取 /dev/drbd/by-res/RESOURCES 格式的符号链接，这种链接更易于使用，而且还能防范在记错设备次要编号的情况下出现安全问题。

例如，如果原始设备大小为 1024 MB，则 DRBD 设备仅有 1023 MB 可用于数据存储，而大约保留 70 KB 的隐藏容量用于存储元数据。通过原始磁盘访问剩余 KB 的任何尝试都会失败，因为这些 KB 不可用于储存用户数据。

20.2 安装 DRBD 服务

按第 I 部分“[安装、设置和升级](#)”中所述在联网群集中的两台 SUSE Linux Enterprise Server 计算机上安装 High Availability Extension。安装 High Availability Extension 时还安装 DRBD 程序文件。

如果不需要完整群集堆栈而只需使用 DRBD，请安装包 drbd、drbd-kmp-FLAVOR、drbd-utils 和 yast2-drbd。

要简化 `drbdadm` 的处理，可使用 Bash 补全支持。如果要在当前外壳会话中启用它，请插入以下命令：

```
root # source /etc/bash_completion.d/drbdadm.sh
```

要将其永久用于 `root`，请创建或扩展文件 `/root/.bashrc` 并插入上一行。

20.3 设置 DRBD 服务



注意：所需的调整

以下过程使用服务器名称 `alice` 和 `bob`，以及群集资源名称 `r0`。它将 `alice` 设置为主节点，并使用 `/dev/sda1` 储存数据。确保修改这些说明，以使用您自己的节点和文件名。

以下几节假定您有两个节点 `alice` 和 `bob`，它们应使用 TCP 端口 `7788`。确保此端口在防火墙中处于打开状态。

1. 准备系统：

- a. 确保 Linux 节点中的块设备已就绪且已分区（如果需要）。
- b. 如果磁盘包含您已不再需要的文件系统，请使用以下命令销毁文件系统结构：

```
root # dd if=/dev/zero of=YOUR_DEVICE count=16 bs=1M
```

如果有多个文件系统需要销毁，请在您希望包含到 DRBD 设置中的所有设备上重复此步骤。

- c. 如果群集已在使用 DRBD，请将群集置于维护模式：

```
root # crm configure property maintenance-mode=true
```

当群集已使用 DRBD 时，如果您跳过此步骤，当前配置中的语法错误会导致服务关闭。

或者，也可以使用 `drbdadm -c 文件` 来测试配置文件。

2. 选择相应的方法来配置 DRBD:

- 第 20.3.1 节 “手动配置 DRBD”
- 第 20.3.2 节 “使用 YaST 配置 DRBD”

3. 如果您已配置 Csync2（这应该是默认设置），则 DRBD 配置文件已包含在需要同步的文件列表中。要同步文件，请使用以下命令：

```
root # csync2 -xv /etc/drbd.d/
```

如果您不具有 Csync2（或不想使用它），请将 DRBD 配置文件手动复制到其他节点：

```
root # scp /etc/drbd.conf bob:/etc/  
root # scp /etc/drbd.d/* bob:/etc/drbd.d/
```

4. 执行初始同步（请参见第 20.3.3 节 “初始化和格式化 DRBD 资源”）。

5. 重置群集的维护模式标志：

```
root # crm configure property maintenance-mode=false
```

20.3.1 手动配置 DRBD



注意：“自动升级”功能支持受限

DRBD9 的“自动升级”功能可以使用克隆和文件系统资源，而非主/从连接。如果使用此功能时正在装入某个文件系统，DRBD 会自动变为主要模式。

目前，自动升级功能仅具有受限支持。使用 DRBD 9 时，SUSE 支持的用例与使用 DRBD 8 时相同。除此以外的用例（例如所包含节点超过两个的设置）不受支持。

要手动设置 DRBD，请按如下操作：

过程 20.1：手动配置 DRBD

从 DRBD 版本 8.3 开始，以前的配置文件将拆分成几个独立的文件，位于目录 /etc/drbd.d/ 下。

1. 打开文件 `/etc/drbd.d/global_common.conf`。它已包含一些全局预定义值。转到 `startup` 部分并插入以下行：

```
startup {
    # wfc-timeout degr-wfc-timeout outdated-wfc-timeout
    # wait-after-sb;
    wfc-timeout 100;
    degr-wfc-timeout 120;
}
```

这些选项用于在引导时减少超时，有关更多细节，请参见 <https://docs.linbit.com/docs/users-guide-9.0/#ch-configure>。

2. 创建文件 `/etc/drbd.d/r0.res`。根据具体情况更改以下几行并保存：

```
resource r0 { ❶
    device /dev/drbd0; ❷
    disk /dev/sda1; ❸
    meta-disk internal; ❹
    on alice { ❺
        address 192.168.1.10:7788; ❻
        node-id 0; ❼
    }
    on bob { ❺
        address 192.168.1.11:7788; ❻
        node-id 1; ❼
    }
    disk {
        resync-rate 10M; ❽
    }
    connection-mesh { ❾
        hosts alice bob;
    }
}
```

- ❶ 方便与需要资源的服务建立某种关联的 DRBD 资源名称。例如 `nfs`、`http`、`mysql_0`、`postgres_wal` 等。本文中更一般的名称 `r0`。
- ❷ DRBD 的设备名及其次要编号。

在以上示例中，为 DRBD 使用了次要编号 0。udev 集成脚本将提供符号链接 `/dev/drbd/by-res/nfs/0`。也可以在配置中忽略设备节点名称，改为使用以下行：
`drbd0 minor 0`（`/dev/` 是可选的）或 `/dev/drbd0`

- ③ 在节点间复制的原始设备。请注意，在本例中，两个节点上的设备相同。如果需要不同的设备，请将 `disk` 参数移到 `on` 主机中。
 - ④ 元磁盘参数通常包含值 `internal`，但可以明确指定一个设备来保存元数据。有关详细信息，请参见 <https://docs.linbit.com/docs/users-guide-9.0/#s-metadata>。
 - ⑤ `on` 部分指定了此配置语句要应用到的主机。
 - ⑥ 各个节点的 IP 地址和端口号。每个资源都需要单独的端口，通常从 `7788` 开始。DRBD 资源的两个端口必须相同。
 - ⑦ 配置两个以上的节点时，需要节点 ID。该 ID 是用于区分不同节点的唯一非负整数。
 - ⑧ 同步率。将其设置为磁盘和网络带宽中较小者的三分之一。它仅限制重新同步，而不限制复制。
 - ⑨ 定义网格的所有节点。`hosts` 参数包含共享相同 DRBD 设置的所有主机名。
3. 检查配置文件的语法。如果以下命令返回错误，请校验文件：

```
root # drbdadm dump all
```

4. 继续第 20.3.3 节“初始化和格式化 DRBD 资源”。

20.3.2 使用 YaST 配置 DRBD

可以使用 YaST 来启动 DRBD 的初始设置。创建 DRBD 设置后，可以手动调整生成的文件。

但是，一旦更改了配置文件，请不要再使用 YaST DRBD 模块。DRBD 模块仅支持有限的一组基本配置。如果您再次使用该模块，它很有可能不会显示您所做的更改。

要使用 YaST 设置 DRBD，请执行以下操作：

过程 20.2：使用 YAST 配置 DRBD

1. 启动 YaST 并选择配置模块 High Availability > DRBD。如果您已有 DRBD 配置，YaST 会向您发出警告。YaST 会更改您的配置，并将旧的 DRBD 配置文件另存为 `*.YaSTsave` 文件。
2. 将启动配置 > 引导中的引导标志保留原样（默认情况下为 `off`）；请不要更改此项设置，因为 Pacemaker 会管理此服务。
3. 如果正在运行防火墙，请启用打开防火墙中的端口。
4. 转到资源配置项。按添加创建新资源（请参见图 20.2 “资源配置”）。

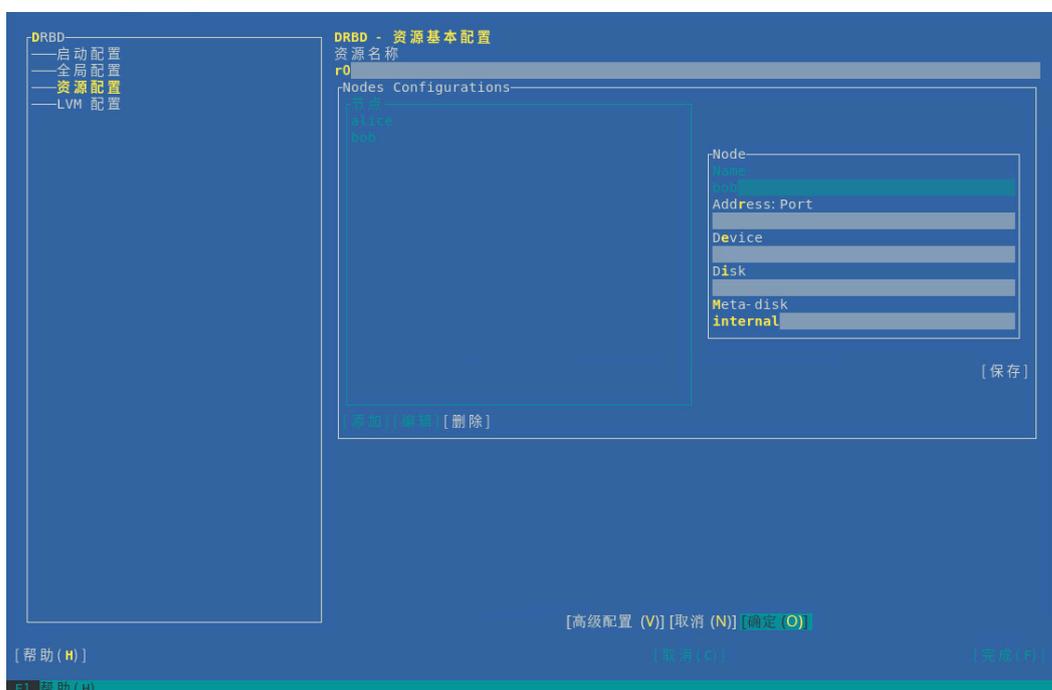


图 20.2：资源配置

需要设置以下参数：

资源名称

DRBD 资源的名称（必填）

名称

相关节点的主机名

地址:端口

相应节点的 IP 地址和端口号 (默认值 7788)

设备

用于访问复制数据的块设备路径。如果设备包含次要编号, 则关联的块设备通常命名为 /dev/drbdX, 其中的 X 是设备次要编号。如果设备不包含次要编号, 请务必在设备名称后面添加 minor 0。

磁盘

在两个节点之间复制的原始设备。如果您使用 LVM, 请插入 LVM 设备名称。

元磁盘

可将元磁盘的值设为 internal, 或明确指定一个由索引扩展的设备来保存 DRBD 所需的元数据。

一个实际设备也可用于多个 DRBD 资源。例如, 如果第一个资源的元磁盘为 /dev/sda6[0], 则可以将 /dev/sda6[1] 用于第二个资源。但是, 必须为此磁盘上的每个资源保留至少 128 MB 空间。固定的元数据大小会限制可复制的最大数据大小。

所有这些选项在 /usr/share/doc/packages/drbd/drbd.conf 文件的示例和 **drbd.conf(5)** 的手册页中均有说明。

5. 单击保存。
6. 单击添加输入第二个 DRBD 资源, 然后单击保存完成。
7. 单击确定和完成关闭资源配置。
8. 如果您对 DRBD 使用 LVM, 则需要在 LVM 配置文件中更改一些选项 (请参见 LVM 配置项)。YaST DRBD 模块可自动完成此项更改。
LVM 过滤器中将会拒绝 DRBD 资源的 localhost 磁盘名称和默认过滤器。只能在 /dev/drbd 中扫描 LVM 设备。
例如, 如果将 /dev/sda1 用作 DRBD 磁盘, 系统会插入设备名称作为 LVM 过滤器中的第一项。要手动更改过滤器, 请单击自动修改 LVM 设备过滤器复选框。
9. 单击完成以保存更改。
10. 继续第 20.3.3 节 “初始化和格式化 DRBD 资源”。

20.3.3 初始化和格式化 DRBD 资源

准备好系统并配置好 DRBD 后，请执行磁盘的首次初始化：

1. 在 alice 和 bob 这两个节点上，初始化元数据储存：

```
root # drbdadm create-md r0
root # drbdadm up r0
```

2. 要想缩短 DRBD 资源的初始重新同步时间，请检查以下项：

- 如果所有节点上的 DRBD 设备都具有相同数据（例如，通过使用第 20.3 节“设置 DRBD 服务”中所述的 `dd` 命令销毁文件系统结构），则请在这两个节点上使用以下命令跳过初始重新同步步骤：

```
root # drbdadm new-current-uuid --clear-bitmap r0/0
```

状态将为 `Secondary/Secondary UpToDate/UpToDate`

- 否则，请继续下一步。

3. 在主节点 alice 上，启动重新同步过程：

```
root # drbdadm primary --force r0
```

4. 使用以下命令检查状态：

```
root # drbdadm status r0
r0 role:Primary
  disk:UpToDate
  bob role:Secondary
  peer-disk:UpToDate
```

5. 在 DRBD 设备上创建文件系统，例如：

```
root # mkfs.ext3 /dev/drbd0
```

6. 装入文件系统并使用它：

```
root # mount /dev/drbd0 /mnt/
```

20.4 从 DRBD 8 迁移到 DRBD 9

从 DRBD 8（随附于 SUSE Linux Enterprise High Availability Extension 12 SP1 中）到 DRBD 9（随附于 SUSE Linux Enterprise High Availability Extension 12 SP2 中），元数据格式已发生变化。DRBD 9 不会将之前的元数据文件自动转换到新格式。

迁移到 12 SP2 之后，在启动 DRBD 之前，请先将 DRBD 元数据手动转换为版本 9 格式。请使用 `drbdadm create-md` 执行此操作。不需要更改任何配置。



注意：支持受限

使用 DRBD 9 时，SUSE 支持的用例与使用 DRBD 8 时相同。除此以外的用例（例如所包含节点超过两个的设置）不受支持。

DRBD 9 将会回退到与版本 8 兼容的状态。如果有三个或更多节点，您需要重新创建元数据才能使用 DRBD 版本 9 特定选项。

如果具有堆叠式 DRBD 资源，另请参见第 20.5 节“创建堆叠式 DRBD 设备”了解详细信息。

要保留数据并允许在无需重新创建新资源的情况下添加新节点，请执行以下操作：

1. 将一个节点设置为待机模式。
2. 更新所有节点上的所有 DRBD 包，请参见第 20.2 节“安装 DRBD 服务”。
3. 将新节点信息添加到资源配置中：
 - `node-id`（针对每个 `on` 部分）。
 - `connection-mesh` 部分在 `hosts` 参数中包含所有主机名。

请参见过程 20.1“手动配置 DRBD”中的示例配置。

4. 在使用 `internal` 作为 `meta-disk` 键时扩大 DRBD 磁盘的空间。使用支持扩大空间的设备，例如 LVM。或者，更换为用于元数据的外部磁盘，并使用 `meta-disk` 设备；。
5. 根据新配置重新创建元数据：

```
root # drbdadm create-md RESOURCE
```

6. 取消待机模式。

20.5 创建堆叠式 DRBD 设备

堆叠式 DRBD 设备包含两个其他设备，其中至少有一个设备也是 DRBD 资源。也就是说，DRBD 在一个现有 DRBD 资源的基础上又添加了一个节点（请参见图 20.3 “资源堆叠”）。此类复制设置可用于备份和灾难恢复用途。

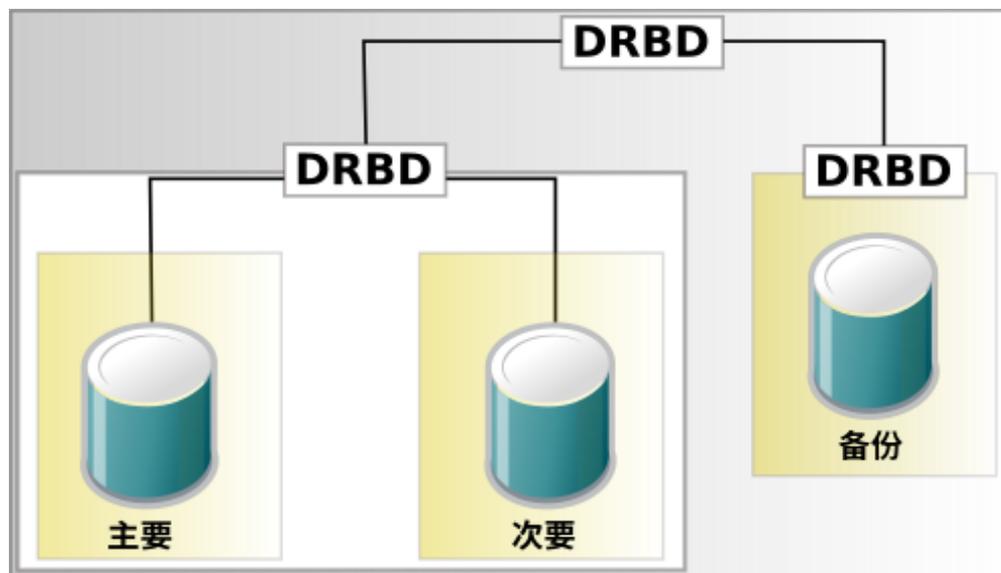


图 20.3：资源堆叠

三向复制运用了异步（DRBD 协议 A）和同步复制（DRBD 协议 C）。异步部分用于堆叠的资源，同步部分用于备用资源。

您的生产环境使用堆叠设备。例如，如果您有一个 DRBD 设备 `/dev/drbd0` 和一个堆叠在其上的设备 `/dev/drbd10`，则将会在 `/dev/drbd10` 上创建文件系统（请参见例 20.1 “三节点堆叠 DRBD 资源的配置”了解更多细节）。

例 20.1：三节点堆叠 DRBD 资源的配置

```
# /etc/drbd.d/r0.res
resource r0 {
    protocol C;
    device    /dev/drbd0;
    disk      /dev/sda1;
    meta-disk internal;

    on amsterdam-alice {
```

```

    address    192.168.1.1:7900;
}

on amsterdam-bob {
    address    192.168.1.2:7900;
}
}

resource r0-U {
    protocol A;
    device     /dev/drbd10;

    stacked-on-top-of r0 {
        address    192.168.2.1:7910;
    }

    on berlin-charlie {
        disk       /dev/sda10;
        address    192.168.2.2:7910; # Public IP of the backup node
        meta-disk  internal;
    }
}
}

```

20.6 使用资源级屏蔽

当 DRBD 复制链路中断时，Pacemaker 会尝试将 DRBD 资源升级到另一个节点。为防止 Pacemaker 使用过时的数据启动服务，请按例 20.2 “使用群集信息库 (CIB) 启用资源级屏蔽的 DRBD 配置” 中所示，在 DRBD 配置文件中启用资源级屏蔽。

例 20.2：使用群集信息库 (CIB) 启用资源级屏蔽的 DRBD 配置

```

resource RESOURCE {
    net {
        fencing resource-only;
        # ...
    }
    handlers {

```

```
fence-peer "/usr/lib/drbd/crm-fence-peer.9.sh";
after-resync-target "/usr/lib/drbd/crm-unfence-peer.9.sh";
# ...
}
...
}
```

如果 DRBD 复制链路断开，DRBD 将执行以下操作：

1. DRBD 调用 `crm-fence-peer.9.sh` 脚本。
2. 该脚本会联系群集管理器。
3. 该脚本会确定与此 DRBD 资源关联的 Pacemaker 资源。
4. 该脚本会确保 DRBD 资源不再升级到其他任何节点。DRBD 资源将保留在当前活动的节点上。
5. 如果复制链路再次连通并且 DRBD 完成了其同步过程，则去除该约束。现在，群集管理器可以任意升级资源。

20.7 测试 DRBD 服务

如果安装和配置过程和预期一样，则您就准备好运行 DRBD 功能的基本测试了。此测试还有助于了解该软件的工作原理。

1. 在 alice 上测试 DRBD 服务。
 - a. 打开终端控制台，然后以 `root` 用户身份登录。
 - b. 在 alice 上创建一个安装点，如 `/srv/r0`：

```
root # mkdir -p /srv/r0
```

- c. 装入 `drbd` 设备：

```
root # mount -o rw /dev/drbd0 /srv/r0
```

- d. 从主节点创建文件：

```
root # touch /srv/r0/from_alice
```

- e. 卸载 alice 上的磁盘：

```
root # umount /srv/r0
```

- f. 通过在 alice 上键入以下命令，降级 alice 上的 DRBD 服务：

```
root # drbdadm secondary r0
```

2. 在 bob 上测试 DRBD 服务。

- a. 在 bob 上打开终端控制台，然后以 `root` 身份登录。

- b. 在 bob 上，将 DRBD 服务升级为主服务：

```
root # drbdadm primary r0
```

- c. 在 bob 上，检查 bob 是否为主节点：

```
root # drbdadm status r0
```

- d. 在 bob 上，创建一个安装点，如 `/srv/r0`：

```
root # mkdir /srv/r0
```

- e. 在 bob 上，装入 DRBD 设备：

```
root # mount -o rw /dev/drbd0 /srv/r0
```

- f. 校验在 alice 上创建的文件是否存在：

```
root # ls /srv/r0/from_alice
```

`/srv/r0/from_alice` 文件应该会列出。

3. 如果该服务在两个节点上都运行正常，则 DRBD 安装即已完成。

4. 再次将 alice 设置为主节点。

- a. 通过在 bob 上键入以下命令，卸下 bob 上的磁盘：

```
root # umount /srv/r0
```

- b. 通过在 bob 上键入以下命令，降级 bob 上的 DRBD 服务：

```
root # drbdadm secondary r0
```

- c. 在 alice 上，将 DRBD 服务升级为主服务：

```
root # drbdadm primary r0
```

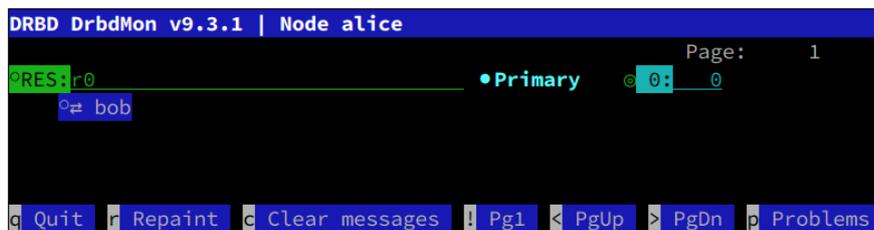
- d. 在 alice 上，检查 alice 是否为主节点：

```
root # drbdadm status r0
```

5. 要使服务在服务器出问题自动启动并进行故障转移，可以使用 Pacemaker/Corosync 将 DRBD 设置为高可用性服务。有关针对 SUSE Linux Enterprise 15 SP2 进行安装和配置的信息，请参见第 II 部分“配置和管理”。

20.8 监视 DRBD 设备

DRBD 随附了可提供实时监视的实用程序 `drbdmon`。该实用程序会显示所有已配置的资源及其问题。



```
DRBD DrbdMon v9.3.1 | Node alice
Page: 1
PRES: r0 ● Primary @ 0: 0
bob ○ Secondary @ 0: 0
q Quit r Repaint c Clear messages Pg1 PgUp PgDn p Problems
```

图 20.4：drbdmon 显示了一个正常的连接

如果出现了问题，`drbdadm` 会显示错误讯息：

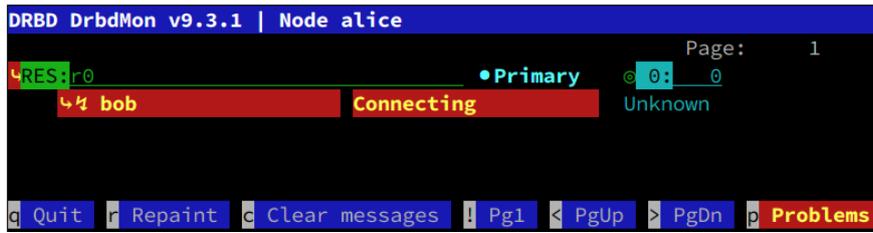


图 20.5：drbdmon 显示了一个错误的连接

20.9 调整 DRBD

可使用几种方式调整 DRBD：

1. 对元数据使用外部磁盘。这可能会有所帮助，不过会降低维护便捷性。
2. 通过 `sysctl` 更改接收和发送缓冲区设置，以优化网络连接。
3. 在 DRBD 配置中更改 `max-buffers` 和/或 `max-epoch-size`。
4. 根据 IO 模式增大 `al-extents` 值。
5. 如果您有一个配备了 BBU（电池备份单元）的硬件 RAID 控制器，设置 `no-disk-flushes`、`no-disk-barrier` 和/或 `no-md-flushes` 可能会有所帮助。
6. 根据工作负载启用读平衡。有关详细信息，请参见 <https://www.linbit.com/en/read-balancing/>。

20.10 DRBD 查错

DRBD 设置涉及很多组件，因此导致问题发生的原因多种多样。以下各部分包括多个常用方案和多种建议解决方案。

20.10.1 配置

如果初始 DRBD 设置不符合预期，说明配置中可能有错误。

获取关于配置的信息：

1. 打开终端控制台，然后以 `root` 用户身份登录。
2. 通过运行 `drbdadm`（带 `-d` 选项），测试配置文件。输入以下命令：

```
root # drbdadm -d adjust r0
```

在 `adjust` 选项的干运行中，`drbdadm` 将 DRBD 资源的实际配置与您的 DRBD 配置文件进行比较，但它不会执行这些调用。检查输出以确保您了解任何错误的根源。

3. 如果 `/etc/drbd.d/*` 和 `drbd.conf` 文件中存在错误，请先更正，然后再继续。
4. 如果分区和设置正确，请在不使用 `-d` 的情况下再次运行 `drbdadm`。

```
root # drbdadm adjust r0
```

这会将配置文件应用到 DRBD 资源。

20.10.2 主机名

对于 DRBD，主机名区分大小写（`Node0` 和 `node0` 是不同的主机），并将与内核中储存的主机名进行比较（参见 `uname -n` 输出）。

如果有多个网络设备，且想要使用专用网络设备，可能不会将主机名解析为所用的 IP 地址。在这种情况下，可使用参数 `disable-ip-verification`。

20.10.3 TCP 端口 7788

如果系统无法连接到对等体，说明本地防火墙可能有问题。默认情况下，DRBD 使用 TCP 端口 `7788` 访问另一个节点。确保在两个节点上该端口均可访问。

20.10.4 DRBD 设备在重引导后损坏

如果 DRBD 不知道哪个实际设备保管了最新数据，它就会变为节点分裂状态。在这种情况下，DRBD 子系统将分别成为次系统，并且互不相连。在这种情况下，可以在日志记录数据中找到以下讯息：

```
Split-Brain detected, dropping connection!
```

要解决此问题，请在要丢弃其数据的节点上输入以下命令：

```
root # drbdadm secondary r0
```

如果状态为 `WFconnection`，则请先断开连接：

```
root # drbdadm disconnect r0
```

在具有最新数据的节点上输入以下命令：

```
root # drbdadm connect --discard-my-data r0
```

通过使用对等体的数据重写一个节点的数据，以此确保两个节点上的视图保持一致，该问题可得到解决。

20.11 更多信息

以下开放源代码资源可用于 DRBD：

- 项目主页：<http://www.drbd.org>。
- 请参见《使用 DRBD 和 Pacemaker 的高度可用 NFS 储存》文章。
- Linux Pacemaker 群集堆栈项目的 http://clusterlabs.org/wiki/DRBD_HowTo_1.0。
- 发行套件中提供了 DRBD 的以下手册页：[drbd\(8\)](#)、[drbdmeta\(8\)](#)、[drbdsetup\(8\)](#)、[drbdadm\(8\)](#)、[drbd.conf\(5\)](#)。
- [/usr/share/doc/packages/drbd-utils/drbd.conf.example](#) 中提供了被注释的 DRBD 示例配置。
- 此外，为了方便地在整个群集中进行储存管理，请参见 <https://www.linbit.com/en/drbd-manager/> 上有关 DRBD-Manager 的最新声明。

21 群集逻辑卷管理器（群集 LVM）

当管理群集上的共享储存区时，所有节点必须收到有关对储存子系统所做更改的通知。Logical Volume Manager 2 (LVM2) 广泛用于管理本地储存，已扩展为支持对整个群集中的卷组进行透明管理。在多个主机之间共享的卷组可使用与本地储存相同的命令进行管理。

21.1 概念概述

系统通过不同的工具来协调群集 LVM：

分布式锁管理器 (DLM)

通过群集范围的锁定协调对多个主机之间共享资源的访问。

逻辑卷管理器 (LVM2)

LVM2 提供磁盘空间的虚拟池，允许将一个逻辑卷灵活分布到多个磁盘。

群集逻辑卷管理器（群集 LVM）

群集 LVM 一词表示群集环境中使用 LVM2。这需要进行一些配置调整，以保护共享储存上的 LVM2 元数据。自 SUSE Linux Enterprise 15 起，群集扩展使用 `lvmlockd`，取代了众所周知的 `clvmd`。有关 `lvmlockd` 的详细信息，请参见 `lvmlockd` 命令的手册页(`man 8 lvmlockd`)。

卷组和逻辑卷

卷组 (VG) 和逻辑卷 (LV) 都属于 LVM2 的基本概念。卷组是多个物理磁盘的储存池。逻辑卷属于卷组，可视为一种弹性卷，您可以在其上创建文件系统。在群集环境中，存在共享 VG 的概念，共享 VG 由共享储存组成，可被多个主机同时使用。

21.2 群集式 LVM 的配置

确保满足以下要求：

- 有共享储存设备可用，例如，该共享储存设备可通过光纤通道、FCoE、SCSI、iSCSI SAN 或 DRBD* 提供。
- 确保已安装以下包：`lvm2` 和 `lvm2-lockd`。
- 自 SUSE Linux Enterprise 15 起，我们使用 `lvmlockd` 作为 LVM2 群集扩展，而不再使用 `clvmd`。确保 `clvmd` 守护程序未运行，否则 `lvmlockd` 将无法启动。

21.2.1 创建群集资源

在一个节点上执行以下基本步骤，以在群集中配置共享 VG：

- 创建 DLM 资源
- 创建 `lvmlockd` 资源
- 创建共享 VG 和 LV
- 创建 LVM 激活资源

过程 21.1：创建 DLM 资源

1. 以 `root` 用户身份启动外壳并登录。
2. 检查群集资源的当前配置：

```
root # crm configure show
```

3. 如果已经配置 DLM 资源（及相应的基本组和基本克隆），则继续过程 21.2 “创建 `lvmlockd` 资源”。

否则，如过程 17.1 “配置 DLM 的基础组”中所述配置 DLM 资源和相应的基本组和基本克隆。

过程 21.2：创建 LVMLOCKD 资源

1. 以 `root` 用户身份启动外壳并登录。
2. 运行以下命令以查看此资源的使用情况：

```
root # crm configure ra info lvmlockd
```

3. 按如下所示配置 `lvmlockd` 资源:

```
root # crm configure primitive lvmlockd ocf:heartbeat:lvmlockd \  
  op start timeout="90" \  
  op stop timeout="100" \  
  op monitor interval="30" timeout="90"
```

4. 为了确保在每个节点上启动 `lvmlockd` 资源, 请将原始资源添加到您在过程 21.1 “创建 DLM 资源” 中为储存创建的基本组:

```
root # crm configure modgroup g-storage add lvmlockd
```

5. 查看所做的更改:

```
root # crm configure show
```

6. 检查资源是否运行正常:

```
root # crm status full
```

过程 21.3 : 创建共享 VG 和 LV

1. 以 `root` 用户身份启动外壳并登录。
2. 假设您已有两个共享磁盘, 并使用它们创建共享 VG:

```
root # vgcreate --shared vg1 /dev/sda /dev/sdb
```

3. 创建 LV, 但一开始不激活它:

```
root # lvcreate -an -L10G -n lv1 vg1
```

过程 21.4 : 创建 LVM 激活资源

1. 以 `root` 用户身份启动外壳并登录。
2. 运行以下命令以查看此资源的使用情况:

```
root # crm configure ra info LVM-activate
```

此资源负责管理 VG 的激活。在共享 VG 中，有两种不同的 LV 激活模式：排它模式和共享模式。排它模式是默认模式，通常应在 `ext4` 等本地文件系统使用 LV 时使用。共享模式仅应用于 OCFS2 等群集文件系统。

3. 配置资源以管理 VG 的激活。根据您的方案，选择下列其中一个选项：

- 对于本地文件系统使用，请使用排它激活模式：

```
root # crm configure primitive vg1 ocf:heartbeat:LVM-activate \  
  params vname=vg1 vg_access_mode=lvmlockd \  
  op start timeout=90s interval=0 \  
  op stop timeout=90s interval=0 \  
  op monitor interval=30s timeout=90s
```

- 对于 OCFS2，请使用共享激活模式，并将其添加到克隆的 `g-storage` 组：

```
root # crm configure primitive vg1 ocf:heartbeat:LVM-activate \  
  params vname=vg1 vg_access_mode=lvmlockd activation_mode=shared \  
  op start timeout=90s interval=0 \  
  op stop timeout=90s interval=0 \  
  op monitor interval=30s timeout=90s  
root # crm configure modgroup g-storage add vg1
```

4. 检查资源是否运行正常：

```
root # crm status full
```

21.2.2 方案：在 SAN 上将群集 LVM 与 iSCSI 搭配使用

以下方案使用两个 SAN 盒，将其 iSCSI 目标导出到多个客户端。大致想法如图 21.1 “使用群集 LVM 设置共享磁盘” 所示。

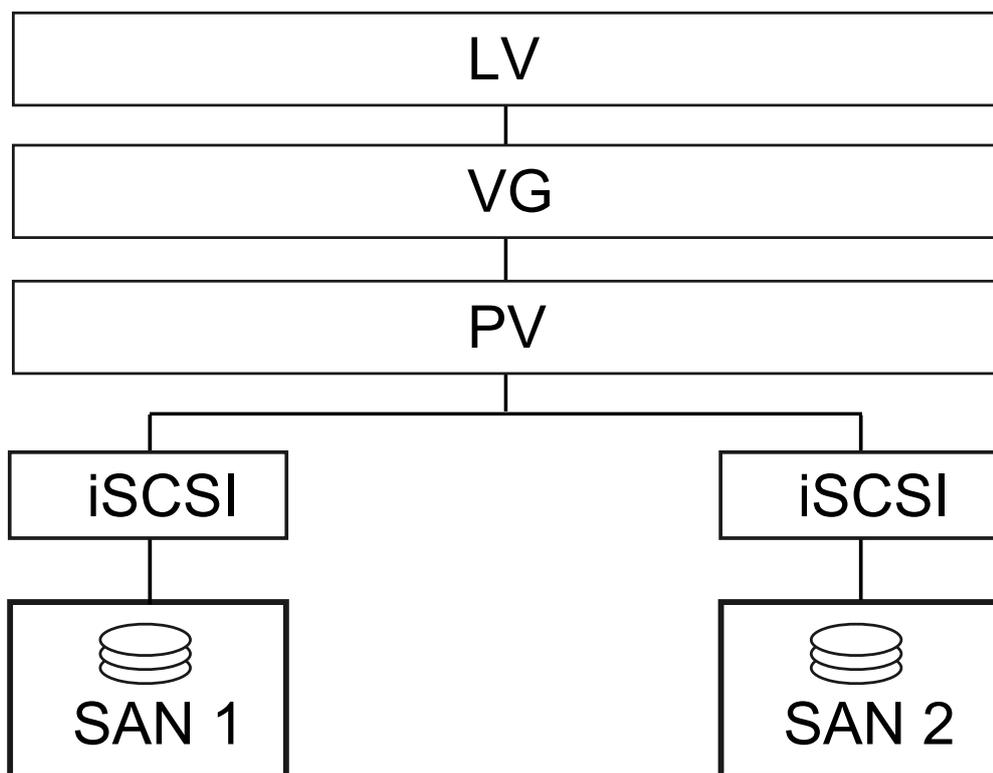


图 21.1：使用群集 LVM 设置共享磁盘



警告：数据丢失

以下过程将损坏磁盘上的所有数据！

首先只配置一个 SAN 盒。每个 SAN Box 都需要导出自己的 iSCSI 目标。按如下所示继续：

过程 21.5：配置 iSCSI 目标 (SAN)

1. 运行 YaST，然后单击 网络服务 > iSCSI LIO 目标 启动 iSCSI 服务器模块。
2. 如果要在计算机引导时启动 iSCSI 目标，请选择引导时，否则请选择手动。
3. 如果正在运行防火墙，请启用打开防火墙中的端口。
4. 切换到全局选项卡。如果需要身份验证，请启用传入及/或传出身份验证。在本例中，我们选择无身份验证。
5. 添加新的 iSCSI 目标：
 - a. 切换到目标选项卡。

- b. 单击添加。
- c. 输入目标名称。名称需要采用如下所示的格式：

```
iqn.DATE.DOMAIN
```

有关格式的更多信息，请参见 3.2.6.3.1. Type "iqn." (iSCSI Qualified Name) (3.2.6.3.1. “iqn.” (iSCSI 限定名称) 类型) 一节，网址：<http://www.ietf.org/rfc/rfc3720.txt>。

- d. 如果需要描述性更强的名称，可以进行更改，但要确保不同目标之间的标识符是唯一的。
 - e. 单击添加。
 - f. 在路径中输入设备名，并使用 Scsiid。
 - g. 单击下一步两次。
6. 出现警告框时单击是进行确认。
 7. 打开配置文件 `/etc/iscsi/iscsid.conf`，将参数 `node.startup` 更改为 `automatic`。

现在按如下方式设置 iSCSI 发起程序：

过程 21.6：配置 iSCSI 发起程序

1. 运行 YaST，然后单击 网络服务 > iSCSI 发起程序。
2. 如果要在计算机引导时启动 iSCSI 发起程序，请选择引导时，否则请将其设置为手动。
3. 切换到发现选项卡并单击发现按钮。
4. 添加 iSCSI 目标的 IP 地址和端口（请参见过程 21.5 “配置 iSCSI 目标 (SAN)”）。通常，可以保留端口并使用其默认值。
5. 如果使用身份验证，请插入进来的和出去的用户名和口令，否则请激活无身份验证。
6. 选择下一步。找到的连接随即显示在列表中。
7. 按完成继续。

8. 打开外壳，并以 `root` 用户身份登录。

9. 测试 iSCSI 发起程序是否已成功启动：

```
root # iscsiadm -m discovery -t st -p 192.168.3.100
192.168.3.100:3260,1 iqn.2010-03.de.jupiter:san1
```

10. 建立会话：

```
root # iscsiadm -m node -l -p 192.168.3.100 -T iqn.2010-03.de.jupiter:san1
Logging in to [iface: default, target: iqn.2010-03.de.jupiter:san1, portal:
192.168.3.100,3260]
Login to [iface: default, target: iqn.2010-03.de.jupiter:san1, portal:
192.168.3.100,3260]: successful
```

使用 `lsscsi` 查看设备名：

```
...
[4:0:0:2]    disk    IET      ...    0      /dev/sdd
[5:0:0:1]    disk    IET      ...    0      /dev/sde
```

查找第三列中有 `IET` 的项。在本例中，设备为 `/dev/sdd` 和 `/dev/sde`。

过程 21.7：创建共享卷组

1. 打开已按过程 21.6 “配置 iSCSI 发起程序” 运行 iSCSI 发起程序的一个节点上的 `root` 外壳。

2. 在磁盘 `/dev/sdd` 和 `/dev/sde` 上创建共享卷组：

```
root # vgcreate --shared testvg /dev/sdd /dev/sde
```

3. 根据需要创建逻辑卷：

```
root # lvcreate --name lv1 --size 500M testvg
```

4. 使用 `vgdisplay` 检查卷组：

```
--- Volume group ---
VG Name                testvg
System ID
```

```

Format                lvm2
Metadata Areas        2
Metadata Sequence No  1
VG Access              read/write
VG Status              resizable
Clustered             yes
Shared                 no
MAX LV                0
Cur LV                0
Open LV                0
Max PV                0
Cur PV                2
Act PV                2
VG Size                1016,00 MB
PE Size                4,00 MB
Total PE              254
Alloc PE / Size       0 / 0
Free PE / Size        254 / 1016,00 MB
VG UUID                UCyWw8-2jqV-enuT-KH4d-NXQI-JhH3-J24anD

```

创建卷并启动资源后，`/dev/testvg` 下会显示新的设备名称，例如 `/dev/testvg/lv1`。这表示 LV 已激活，可以使用。

21.2.3 方案：将群集 LVM 与 DRBD 搭配使用

如果数据中心位于城市、国家/地区或大洲的不同区域，则可使用以下方案。

过程 21.8：创建使用 DRBD 的群集感知卷组

1. 创建主/主 DRBD 资源：

- a. 首先，按过程 20.1 “手动配置 DRBD” 中所述将 DRBD 设备设置为主/从模式。确保两个节点上的磁盘状态均为 `up-to-date`。使用 `drbdadm status` 确认是否如此。
- b. 将以下选项添加到配置文件（通常类似 `/etc/drbd.d/r0.res`）：

```
resource r0 {
```

```
net {
    allow-two-primaries;
}
...
}
```

- c. 将更改的配置文件复制到另一个节点，例如：

```
root # scp /etc/drbd.d/r0.res venus:/etc/drbd.d/
```

- d. 在两个节点上运行以下命令：

```
root # drbdadm disconnect r0
root # drbdadm connect r0
root # drbdadm primary r0
```

- e. 检查节点的状态：

```
root # drbdadm status r0
```

2. 将 `lvmlockd` 资源作为克隆包含在 Pacemaker 配置中，并使它依赖于 DLM 克隆资源。有关详细指示信息，请参见[过程 21.1 “创建 DLM 资源”](#)。继续之前，请确认这些资源已在群集上成功启动。可以使用 `crm status` 或 Web 界面检查正在运行的服务。
3. 使用命令 `pvcreate` 准备 LVM 的物理卷。例如，在设备 `/dev/drbd_r0` 上，命令应类似于：

```
root # pvcreate /dev/drbd_r0
```

4. 创建共享卷组：

```
root # vgcreate --shared testvg /dev/drbd_r0
```

5. 根据需要创建逻辑卷。您有时可能想要更改逻辑卷的大小。例如，使用以下命令创建 4 GB 的逻辑卷：

```
root # lvcreate --name lv1 -L 4G testvg
```

6. 现在 VG 内的逻辑卷可作为文件系统装入或原始用法提供。确保使用逻辑卷的服务具备适当的依赖项，以便在激活 VG 后对它们进行共置和排序。

完成这些配置步骤后，即可像在任何独立工作站中一样进行 LVM2 配置。

21.3 显式配置合格的 LVM2 设备

如果看似有若干设备共享同一个物理卷签名（多路径设备或 DRBD 就有可能发生这种情况），建议显式配置 LVM2 扫描 PV 的设备。

例如，如果命令 **vgcreate** 使用物理设备而非镜像块设备，DRBD 会产生混乱。进而导致 DRBD 出现节点分裂情况。

要停用 LVM2 的单个设备，请执行以下操作：

1. 编辑文件 `/etc/lvm/lvm.conf` 并搜索以 `filter` 开头的行。
2. 其中的模式作为正则表达式来处理。前面的“a”表示接受扫描的设备模式，前面的“r”表示拒绝遵守该设备模式的设备。
3. 要删除名为 `/dev/sdb1` 的设备，请向过滤规则添加以下表达式：

```
"r|^/dev/sdb1$|"
```

完整的过滤行将显示如下：

```
filter = [ "r|^/dev/sdb1$|", "r|/dev/.*/by-path/.*/", "r|/dev/.*/by-id/.*/", "a/.*/" ]
```

接受 DRBD 和 MPIO 设备但拒绝其他所有设备的过滤行如下所示：

```
filter = [ "a|/dev/drbd.*|", "a|/dev/.*/by-id/dm-uuid-mpath-.*/", "r/.*/" ]
```

4. 编写配置文件并将它复制到所有群集节点。

21.4 从镜像 LV 联机迁移到群集 MD

从 SUSE Linux Enterprise High Availability Extension 15 开始，群集 LVM 中的 `cmirror` 已遭弃用。我们强烈建议将群集中的镜像逻辑卷迁移到群集 MD。群集 MD 表示“群集多设备”，是适用于群集的基于软件的 RAID 储存解决方案。

21.4.1 迁移之前的示例设置

假设您采用以下示例设置：

- 您有一个双节点群集，它由节点 `alice` 和 `bob` 组成。
- 名为 `test-lv` 的镜像逻辑卷是基于名为 `cluster-vg2` 的卷组创建的。
- 卷组 `cluster-vg2` 由磁盘 `/dev/vdb` 和 `/dev/vdc` 组成。

```
root # lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
vda                                  253:0    0   40G  0 disk
├─vda1                               253:1    0    4G  0 part [SWAP]
└─vda2                               253:2    0   36G  0 part /
vdb                                  253:16   0   20G  0 disk
├─cluster--vg2-test--lv_mlog_mimage_0 254:0    0    4M  0 lvm
│ └─cluster--vg2-test--lv_mlog        254:2    0    4M  0 lvm
│   └─cluster--vg2-test--lv          254:5    0   12G  0 lvm
└─cluster--vg2-test--lv_mimage_0     254:3    0   12G  0 lvm
    └─cluster--vg2-test--lv          254:5    0   12G  0 lvm
vdc                                  253:32   0   20G  0 disk
├─cluster--vg2-test--lv_mlog_mimage_1 254:1    0    4M  0 lvm
│ └─cluster--vg2-test--lv_mlog        254:2    0    4M  0 lvm
│   └─cluster--vg2-test--lv          254:5    0   12G  0 lvm
└─cluster--vg2-test--lv_mimage_1     254:4    0   12G  0 lvm
    └─cluster--vg2-test--lv          254:5    0   12G  0 lvm
```

! 重要：避免迁移失败

在启动迁移过程之前，请检查逻辑卷和物理卷的容量与利用率。如果逻辑卷使用了 100% 的物理卷容量，则迁移可能会失败，并在目标卷上显示 可用空间不足 错误。如何防止这种迁移失败取决于镜像日志所用的选项：

- **镜像日志本身是否已镜像（mirrored 选项），并且已在镜像根所在的同一个设备上分配？**（例如，如果您根据 [这些版本的《管理指南》](https://documentation.suse.com/sle-ha/12-SP5/html/SLE-HA-all/cha-ha-clvm.html#sec-ha-clvm-config-cmirrord) (<https://documentation.suse.com/sle-ha/12-SP5/html/SLE-HA-all/cha-ha-clvm.html#sec-ha-clvm-config-cmirrord>) 中所述，为 SUSE Linux Enterprise High Availability Extension 11 或 12 上的 cmirrord 设置创建了逻辑卷，则可能符合这种情况）。
默认情况下，mdadm 会在设备开头与数组数据开头之间保留一定的空间量。在迁移期间，您可以检查未使用的填充空间，并使用 data-offset 选项减小此空间，如 [步骤 1.d](#) 和下文所述。
data-offset 必须在设备上保留足够的空间，使群集 MD 能够将其元数据写入设备。另一方面，偏移量必须足够小，使设备的剩余容量可以容纳所迁移卷的所有物理卷区域。由于卷可能已跨越整个设备但不包括镜像日志，因此，偏移量必须小于镜像日志的大小。
我们建议将 data-offset 设置为 128 KB。如果未指定偏移量的值，其默认值为 1 KB（1024 字节）。
- **镜像日志是已写入不同的设备（disk 选项）还是保留在内存中（core 选项）？**
在开始迁移之前，请增大物理卷的大小，或减小逻辑卷的大小（以便为物理卷释放更多的空间）。

21.4.2 将镜像 LV 迁移到群集 MD

以下过程基于 [第 21.4.1 节“迁移之前的示例设置”](#)。请根据设置调整指令，并相应地替换 LV、VG、磁盘和群集 MD 设备的名称。

迁移过程完全不会造成停机。在迁移过程中仍可装入文件系统。

1. 在节点 alice 上执行以下步骤：

- a. 将镜像逻辑卷 `test-lv` 转换为线性逻辑卷：

```
root # lvconvert -m0 cluster-vg2/test-lv /dev/vdc
```

- b. 从卷组 `cluster-vg2` 中去除物理卷 `/dev/vdc`：

```
root # vgreduce cluster-vg2 /dev/vdc
```

- c. 从 LVM 中去除以下物理卷：

```
root # pvremove /dev/vdc
```

如果现在就运行 `lsblk`，您将会看到：

NAME		MAJ:MIN	RM	SIZE	RO	TYPE
MOUNTPOINT						
vda		253:0	0	40G	0	disk
├vda1		253:1	0	4G	0	part [SWAP]
└vda2		253:2	0	36G	0	part /
vdb		253:16	0	20G	0	disk
└cluster--vg2-test--lv		254:5	0	12G	0	lvm
vdc		253:32	0	20G	0	disk

- d. 使用磁盘 `/dev/vdc` 创建群集 MD 设备 `/dev/md0`：

```
root # mdadm --create /dev/md0 --bitmap=clustered \  
--metadata=1.2 --raid-devices=1 --force --level=mirror \  
/dev/vdc --data-offset=128
```

有关为何要使用 `data-offset` 选项的细节，请参见[重要：避免迁移失败](#)。

2. 在节点 `bob` 上组装以下 MD 设备：

```
root # mdadm --assemble md0 /dev/vdc
```

如果您的群集由两个以上的节点组成，请在该群集中的所有剩余节点上执行此步骤。

3. 返回到节点 `alice`：

- a. 将 MD 设备 `/dev/md0` 初始化为与 LVM 搭配使用的物理卷：

```
root # pvcreate /dev/md0
```

- b. 将 MD 设备 `/dev/md0` 添加到卷组 `cluster-vg2`：

```
root # vgextend cluster-vg2 /dev/md0
```

- c. 将磁盘 `/dev/vdb` 中的数据移到 `/dev/md0` 设备：

```
root # pvmove /dev/vdb /dev/md0
```

- d. 从卷组 `cluster-vg2` 中去除物理卷 `/dev/vdb`：

```
root # vgreduce cluster-vg2 /dev/vdb
```

- e. 从设备中去除标签，使 LVM 不再将该设备识别为物理卷：

```
root # pvremove /dev/vdb
```

- f. 将 `/dev/vdb` 添加到 MD 设备 `/dev/md0`：

```
root # mdadm --grow /dev/md0 --raid-devices=2 --add /dev/vdb
```

21.4.3 迁移之后的示例设置

如果现在就运行 `lsblk`，您将会看到：

```
NAME                                MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
vda                                  253:0    0   40G  0 disk
├─vda1                               253:1    0    4G  0 part  [SWAP]
└─vda2                               253:2    0   36G  0 part  /
vdb                                  253:16   0   20G  0 disk
└─md0                                9:0      0   20G  0 raid1
   └─cluster--vg2-test--lv          254:5    0   12G  0 lvm
vdc                                  253:32   0   20G  0 disk
└─md0                                9:0      0   20G  0 raid1
```

```
└─cluster--vg2-test--lv 254:5    0    12G  0 lvm
```

21.5 更多信息

有关 `lvmlockd` 的详细信息，请参见 `lvmlockd` 命令的手册页(`man 8 lvmlockd`)。

可从 <http://www.clusterlabs.org/wiki/Help:Contents> 处的 pacemaker 邮件列表中获取完整信息。

22 群集多设备（群集 MD）

群集多设备（群集 MD）是一项基于软件的群集 RAID 储存解决方案。目前，群集 MD 为群集提供了 RAID1 镜像冗余。在 SUSE Linux Enterprise High Availability Extension 15 SP2 中，随附了 RAID10 技术预览版。如果您要尝试 RAID10，请在相关 `mdadm` 命令中用 `10` 替换 `mirror`。本章介绍如何创建和使用群集 MD。

22.1 概念概述

群集 MD 提供在群集环境中使用 RAID1 的支持。每个节点都会访问群集 MD 使用的磁盘或设备。如果群集 MD 的一个设备发生故障，则在运行时可以由另一个设备代替它，并且系统会对其进行重新同步以提供相同数量的冗余。群集 MD 需要使用 Corosync 和分布式锁管理器 (DLM) 进行协调和讯息交换。

群集 MD 设备不会像其他常规 MD 设备一样在引导时自动启动。为确保 DLM 资源已启动，需要使用资源代理来启动群集设备。

22.2 创建群集 MD RAID 设备

要求

- 一个安装了 Pacemaker 的正在运行的群集。
- DLM 的资源代理（有关如何配置 DLM 的信息，请参见[过程 17.1 “配置 DLM 的基础组”](#)）。
- 至少两个共享磁盘设备。您可以使用另外一个设备作为备用设备，该备用设备会在设备发生故障时自动进行故障转移。
- 一个已安装的包 `cluster-md-kmp-default`。

1. 请确保 DLM 资源在群集的每个节点上都正常运行，并使用以下命令检查资源状态：

```
root # crm_resource -r dlm -W
```

2. 创建群集 MD 设备：

- 如果您没有现有的常规 RAID 设备，请使用以下命令在运行 DLM 资源的节点上创建群集 MD 设备：

```
root # mdadm --create /dev/md0 --bitmap=clustered \  
--metadata=1.2 --raid-devices=2 --level=mirror /dev/sda /dev/sdb
```

由于群集 MD 只能与 1.2 版的元数据配合使用，因此建议使用 `--metadata` 选项来指定版本。有关其他有用选项，请参见 `mdadm` 手册页。在 `/proc/mdstat` 中监控重新同步进度。

- 如果您有现有的常规 RAID，请先清除现有位图，然后再创建群集位图：

```
root # mdadm --grow /dev/mdX --bitmap=none  
root # mdadm --grow /dev/mdX --bitmap=clustered
```

- （可选）要创建带有用于自动故障转移的备用设备的群集 MD 设备，请在群集节点上运行以下命令：

```
root # mdadm --create /dev/md0 --bitmap=clustered --raid-devices=2 \  
--level=mirror --spare-devices=1 /dev/sda /dev/sdb /dev/sdc --  
metadata=1.2
```

3. 获取 UUID 以及相关的 MD 路径：

```
root # mdadm --detail --scan
```

该 UUID 必须与超级块中储存的 UUID 匹配。有关 UUID 的细节，请参见 `mdadm.conf` 手册页。

4. 打开 `/etc/mdadm.conf` 并添加 MD 设备名称以及与其关联的设备。使用上一步中获得的 UUID：

```
DEVICE /dev/sda /dev/sdb  
ARRAY /dev/md0 UUID=1d70f103:49740ef1:af2afce5:fcf6a489
```

5. 打开 Csync2 的配置文件 `/etc/csync2/csync2.cfg`，添加 `/etc/mdadm.conf`：

```
group ha_group
```

```
{
  # ... list of files pruned ...
  include /etc/mdadm.conf
}
```

22.3 配置资源代理

按如下所示配置 CRM 资源：

1. 创建 `Raid1` 原始资源：

```
crm(live)configure# primitive raider Raid1 \  
  params raidconf="/etc/mdadm.conf" raiddev=/dev/md0 \  
  force_clones=true \  
  op monitor timeout=20s interval=10 \  
  op start timeout=20s interval=0 \  
  op stop timeout=20s interval=0
```

2. 将 `raider` 资源添加到您已为 DLM 创建的储存基础组：

```
crm(live)configure# modgroup g-storage add raider
```

add 子命令默认会追加新的组成员。

如果尚未克隆 `g-storage` 组，请执行该操作，以使其在所有节点上运行：

```
crm(live)configure# clone cl-storage g-storage \  
  meta interleave=true target-role=Started
```

3. 使用 `show` 查看更改。

4. 如果所有配置正确无误，请使用 `commit` 提交您的更改。

22.4 添加设备

要将某个设备添加到现有的活动群集 MD 设备，请先使用命令 `cat /proc/mdstat` 确保该设备在每个节点上均“可见”。如果设备不可见，则命令将会失败。

在一个群集节点上使用以下命令：

```
root # mdadm --manage /dev/md0 --add /dev/sdc
```

所添加的新设备的行为取决于群集 MD 设备的状态：

- 如果只有一个镜像设备处于活动状态，则新设备将会成为镜像设备中的第二个设备，并且会启动恢复进程。
- 如果群集 MD 设备的两个设备都处于活动状态，则新添加的设备将会成为备用设备。

22.5 重新添加暂时发生故障的设备

故障常常是短暂发生的，并且仅限于一个节点。如果在 I/O 操作期间有任何节点发生了故障，则会针对整个群集将设备标记为失败。

例如，其中一个节点发生电缆故障，可能就会导致发生这种情况。更正此问题后，您可以重新添加设备。与添加新设备会同步整个设备不同，这样只会同步过时的部件。

要重新添加设备，请在一个群集节点上运行以下命令：

```
root # mdadm --manage /dev/md0 --re-add /dev/sdb
```

22.6 去除设备

在运行时去除设备以进行替换之前，请执行以下操作：

1. 检查 `/proc/mdstat` 以确保设备已失败。查看设备前面有无 `(F)`。
2. 在一个群集节点上运行以下命令，以使设备失败：

```
root # mdadm --manage /dev/md0 --fail /dev/sda
```

3. 在一个群集节点上使用以下命令去除失败的设备：

```
root # mdadm --manage /dev/md0 --remove /dev/sda
```

23 Samba 群集

群集 Samba 服务器提供异构网络的高可用性解决方案。本章说明了一些背景信息以及如何设置群集 Samba 服务器。

23.1 概念概述

Samba 使用 Trivial Database (TDB) 已经许多年了。它允许多个应用程序同时写入。为确保所有写操作都成功执行而不会彼此冲突，TDB 使用内部锁定机制。

Cluster Trivial Database (CTDB) 是现有的 TDB 的小扩展。项目对 CTDB 的描述是：“Samba 和其他项目用于储存临时数据的 TDB 数据库的群集实现”。

每个群集节点都运行本地 CTDB 守护程序。Samba 与其本地 CTDB 守护程序通讯，而非直接写入其 TDB。守护程序通过网络交换元数据，但实际的读写操作是在快速储存的本地副本上进行的。CTDB 的概念如[图 23.1 “CTDB 群集的结构”](#)中所示。



注意：CTDB 仅用于 Samba

CTDB 资源代理的当前实现将 CTDB 配置为只管理 Samba。任何其他功能，包括 IP 故障转移，都应使用 Pacemaker 进行配置。

CTDB 仅支持完全同类的群集。例如，群集中的所有节点都需要具有相同的体系结构。不能混用 x86 与 AMD64。



图 23.1：CTDB 群集的结构

群集 Samba 服务器必须共享某些数据：

- 将 Unix 用户和组 ID 与 Windows 用户和组关联的映射表。
- 用户数据库必须在所有节点间同步。
- Windows 域中的成员服务器的连接信息必须在所有节点上都可用。
- 元数据（如活动 SMB 会话、共享连接和各种锁）需在所有节点上都可用。

目标是：具有 N+1 个节点的群集 Samba 服务器比只有 N 个节点的快。一个节点不会比非群集 Samba 服务器慢。

23.2 基本配置

注意：已更改的配置文件

CTDB 资源代理会自动更改 `/etc/sysconfig/ctdb`。使用 `crm ra info CTDB` 列出可为 CTDB 资源指定的所有参数。

要设置群集 Samba 服务器，请按如下操作：

过程 23.1：设置基本的群集 SAMBA 服务器

1. 准备群集：

- a. 在继续下一步之前，请确保已安装以下包：`ctdb`、`tdb-tools` 和 `samba`（`smb` 和 `nmb` 资源需要这些包）。
- b. 根据本指南的第 II 部分“配置和管理”中所述配置您的群集（Pacemaker、OCFS2）。
- c. 配置共享文件系统（如 OCFS2）并将其装入，例如装入到 `/srv/clusterfs` 上。有关详细信息，请参见第 18 章“OCFS2”。
- d. 要打开 POSIX ACL，请启用它：

- 对新的 OCFS2 文件系统，使用：

```
root # mkfs.ocfs2 --fs-features=xattr ...
```

- 对现有 OCFS2 文件系统，使用：

```
root # tuneufs.ocfs2 --fs-feature=xattr DEVICE
```

确保在文件系统资源中指定了 `acl` 选项。按如下方式使用 `crm` 外壳：

```
crm(live)configure# primitive ocfs2-3 ocf:heartbeat:Filesystem  
params options="acl" ...
```

- e. 确保服务 `ctdb`、`smb` 和 `nmb` 已禁用：

```
root # systemctl disable ctdb  
root # systemctl disable smb  
root # systemctl disable nmb
```

- f. 在所有节点上打开防火墙的端口 `4379`。这是为了使 CTDB 能够与其他集群节点通讯。

2. 在共享文件系统上为 CTDB 锁定创建一个目录：

```
root # mkdir -p /srv/clusterfs/samba/
```

3. 在 `/etc/ctdb/nodes` 中插入包含集群中每个节点的所有私有 IP 地址的所有节点：

```
192.168.1.10  
192.168.1.11
```

4. 配置 Samba。在 `/etc/samba/smb.conf` 的 `[global]` 部分中添加以下行。使用所选的主机名取代“CTDB-SERVER”（群集中的所有节点将显示为一个此名称的大节点，以方便操作）：

```
[global]  
# ...  
# settings applicable for all CTDB deployments  
netbios name = CTDB-SERVER  
clustering = yes  
idmap config * : backend = tdb2  
passdb backend = tdbsam  
ctdbd socket = /var/lib/ctdb/ctdb.socket  
# settings necessary for CTDB on OCFS2  
fileid:algorithm = fsid  
vfs objects = fileid  
# ...
```

5. 使用 `csync2` 将配置文件复制到您的所有节点：

```
root # csync2 -xv
```

有关详细信息，请参见[过程 4.6 “使用 Csync2 同步配置文件”](#)。

6. 将 CTDB 资源添加到群集：

```
root # crm configure  
crm(live)configure# primitive ctdb ocf:heartbeat:CTDB params \  
  ctdb_manages_winbind="false" \  
  ctdb_manages_samba="false" \  
  ctdb_recovery_lock="/srv/clusterfs/samba/ctdb.lock" \  
  ctdb_socket="/var/lib/ctdb/ctdb.socket" \  
  \
```

```

    op monitor interval="10" timeout="20" \
    op start interval="0" timeout="90" \
    op stop interval="0" timeout="100"
crm(live)configure# primitive nmb systemd:nmb \
    op start timeout="60" interval="0" \
    op stop timeout="60" interval="0" \
    op monitor interval="60" timeout="60"
crm(live)configure# primitive smb systemd:smb \
    op start timeout="60" interval="0" \
    op stop timeout="60" interval="0" \
    op monitor interval="60" timeout="60"
crm(live)configure# group g-ctdb ctdb nmb smb
crm(live)configure# clone cl-ctdb g-ctdb meta interleave="true"
crm(live)configure# colocation col-ctdb-with-clusterfs Mandatory: cl-ctdb
    cl-clusterfs
crm(live)configure# order o-clusterfs-then-ctdb Mandatory: cl-clusterfs cl-
    ctdb
crm(live)configure# commit

```

7. 添加集群 IP 地址:

```

crm(live)configure# primitive ip ocf:heartbeat:IPaddr2 params
    ip=192.168.2.222 \
    unique_clone_address="true" \
    op monitor interval="60" \
    meta resource-stickiness="0"
crm(live)configure# clone cl-ip ip \
    meta interleave="true" clone-node-max="2" globally-unique="true"
crm(live)configure# colocation col-ip-with-ctdb 0: cl-ip cl-ctdb
crm(live)configure# order o-ip-then-ctdb 0: cl-ip cl-ctdb
crm(live)configure# commit

```

如果 `unique_clone_address` 设置为 `true`，`IPaddr2` 资源代理将向指定的地址添加一个克隆 ID，从而导致出现三个不同的 IP 地址。这些地址通常是不需要的，但有助于实现负载均衡。有关此主题的更多信息，请参见第 14.2 节“使用 Linux 虚拟服务器配置负载均衡”。

8. 提交更改:

```
crm(live)configure# commit
```

9. 检查结果:

```
root # crm status
Clone Set: cl-storage [dlm]
  Started: [ factory-1 ]
  Stopped: [ factory-0 ]
Clone Set: cl-clusterfs [clusterfs]
  Started: [ factory-1 ]
  Stopped: [ factory-0 ]
Clone Set: cl-ctdb [g-ctdb]
  Started: [ factory-1 ]
  Started: [ factory-0 ]
Clone Set: cl-ip [ip] (unique)
  ip:0      (ocf:heartbeat:IPaddr2):      Started factory-0
  ip:1      (ocf:heartbeat:IPaddr2):      Started factory-1
```

10. 从客户端计算机进行测试。在 Linux 客户端上运行以下命令，以检查能否从系统复制文件以及将文件复制到系统:

```
root # smbclient //192.168.2.222/myshare
```

23.3 加入 Active Directory 域

Active Directory (AD) 是 Windows Server 系统的一项目录服务。

下列说明概述了如何将 CTDB 群集加入到 Active Directory 域:

1. 按照过程 23.1 “设置基本的群集 Samba 服务器” 中所述创建 CTDB 资源。
2. 安装 `samba-winbind` 包中提供。
3. 禁用 `winbind` 服务:

```
root # systemctl disable winbind
```

4. 定义 `winbind` 群集资源:

```

root # crm configure
crm(live)configure# primitive winbind systemd:winbind \
  op start timeout="60" interval="0" \
  op stop timeout="60" interval="0" \
  op monitor interval="60" timeout="60"
crm(live)configure# commit

```

5. 编辑 `g-ctdb` 组，并在 `nmb` 与 `smb` 资源之间插入 `winbind`：

```

crm(live)configure# edit g-ctdb

```

保存更改，然后使用 `:w` (`vim`) 关闭编辑器。

6. 有关如何设置 Active Directory 域的说明，请参见 Windows Server 文档。在此示例中，使用以下参数：

AD 和 DNS 服务器	win2k3.2k3test.example.com
AD 域	2k3test.example.com
群集 AD 成员 NETBIOS 名称	CTDB-SERVER

7. 过程 23.2 “加入 Active Directory”

最后，将群集加入 Active Directory 服务器：

过程 23.2：加入 ACTIVE DIRECTORY

1. 请确保 Csync2 的配置中包含下列文件，才可在所有群集主机上进行安装：

```

/etc/samba/smb.conf
/etc/security/pam_winbind.conf
/etc/krb5.conf
/etc/nsswitch.conf
/etc/security/pam_mount.conf.xml
/etc/pam.d/common-session

```

您也可以为此任务使用 YaST 的配置 Csync2 模块，请参见第 4.5 节 “将配置传送到所有节点”。

2. 运行 YaST 并从网络服务项中打开 Windows 域成员资格模块。
3. 输入域或工作组设置然后单击确定完成设置。

23.4 调试和测试群集 Samba

要调试群集 Samba 服务器，可使用以下作用于不同级别的工具：

ctdb_diagnostics

运行此工具可诊断群集 Samba 服务器。详细的调试消息应有助于您跟踪任何可能存在的问题。

ctdb_diagnostics 命令可搜索以下文件，这些文件必须在所有节点上都可用：

```
/etc/krb5.conf
/etc/hosts
/etc/ctdb/nodes
/etc/sysconfig/ctdb
/etc/resolv.conf
/etc/nsswitch.conf
/etc/sysctl.conf
/etc/samba/smb.conf
/etc/fstab
/etc/multipath.conf
/etc/pam.d/system-auth
/etc/sysconfig/nfs
/etc/exports
/etc/vsftpd/vsftpd.conf
```

如果文件 /etc/ctdb/public_addresses 和 /etc/ctdb/static-routes 存在，它们也会被检查。

ping_pong

检查文件系统是否支持 CTDB 使用 ping_pong。它会对群集文件系统执行一致性和性能之类的特定测试（请参见 http://wiki.samba.org/index.php/Ping_pong），从而给出群集在高负载下将会表现如何的一些指示。

send_arp 工具和 SendArp 资源代理

SendArp 资源代理位于 `/usr/lib/heartbeat/send_arp`（或 `/usr/lib64/heartbeat/send_arp`）。**send_arp** 工具发出免费的 ARP（Address Resolution Protocol，地址解析协议）包，可用于更新其他计算机的 ARP 表。它可以帮助确定故障转移过程之后的通讯问题。如果节点对 Samba 显示了群集 IP 地址，但您却无法连接到节点或 ping 到它，请使用 **send_arp** 命令测试节点是否只需要 ARP 表更新。有关更多信息，请参见 http://wiki.wireshark.org/Gratuitous_ARP。

要测试群集文件系统的某些方面，请如下继续操作：

过程 23.3：测试群集文件系统的一致性和性能

1. 在一个节点上启动命令 **ping_pong**，将占位符 `N` 替换为节点数 + 1。文件 `ABSPATH/data.txt` 在共享存储区中提供，因此在所有节点（`ABSPATH` 表示绝对路径）上都可以访问到：

```
ping_pong ABSPATH/data.txt N
```

应该会得到很高的锁定率，因为只运行一个节点。如果程序不打印锁定率，请替换群集文件系统。

2. 使用相同的参数在另一个节点上启动 **ping_pong** 的第二个副本。
应该会看到锁定率急剧下降。如果以下任意情况适用于群集文件系统，请替换它：
 - **ping_pong** 不打印每秒锁定率，
 - 两个实例中的锁定率并非几乎相等，
 - 启动第二个实例后锁定率未下降。
3. 启动 **ping_pong** 的第三个副本。添加另一个节点，注意锁定率的变化。
4. 逐个终止 **ping_pong** 命令。应该观察到锁定率上升，直到回到单一节点的情况。如果没有看到预期行为，请参见第 18 章“OCFS2”中的详细信息。

23.5 更多信息

- <http://www.linux-ha.org/doc/man-pages/re-ra-CTDB.html> ↗
- http://wiki.samba.org/index.php/CTDB_Setup ↗
- <http://ctdb.samba.org> ↗
- http://wiki.samba.org/index.php/Samba_%26_Clustering ↗

24 使用 Rear (Relax-and-Recover) 实现灾难恢复

Relax-and-Recover (“Rear”，本章中缩写为 Rear) 是供系统管理员使用的灾难恢复框架。它是一个 Bash 脚本集合，您需要根据要在发生灾难时加以保护的特定生产环境调整这些脚本。

不存在能够现成地解决问题的灾难恢复解决方案。因此，在发生任何灾难之前做好预防措施至关重要。

24.1 概念概述

以下几节介绍了一般性的灾难恢复概念，以及使用 Rear 成功实现恢复所需执行的基本步骤。另外，还提供了一些有关 Rear 要求、要注意的一些限制、各种方案和备份工具的指导。

24.1.1 创建灾难恢复计划

在最坏的情况发生之前采取措施：分析 IT 基础结构是否存在任何重大风险，评估您的预算，并创建灾难恢复计划。如果您尚未创建灾难恢复计划，请先了解有关以下每个步骤的一些信息：

- **风险分析：** 对基础设施进行可靠的风险分析。列出所有可能的威胁并评估它们的严重性。确定这些威胁的相似程度并划分优先级。建议使用简单的分类：可能性和影响。
- **预算计划：** 分析结果是一个概述，指出哪些风险可以忍受，哪些风险对业务非常关键。请自问一下，怎样才能最大限度地降低风险，以及降低风险所需的费用。根据公司的规模，在灾难恢复方面的花费占总体 IT 预算的 2% 到 15%。
- **灾难恢复计划开发：** 制作核对清单、测试过程、建立并指派优先级以及列出 IT 基础设施库存。定义当基础设施中的一些服务失败时，如何处理问题。
- **测试：** 定义详细的计划后，测试该计划。每年至少测试一次。使用与主要 IT 基础设施相同的测试硬件。

24.1.2 灾难恢复意味着什么？

如果生产环境中的某个系统已毁坏（可能出于任何原因 - 例如，硬件损坏、配置不当或软件问题），您需要重建该系统。可以在相同的硬件或者兼容的替代硬件上重建。重建系统并不只是意味着从备份中恢复文件，还包括准备系统的储存（与分区、文件系统和安装点相关），以及重新安装引导加载程序。

24.1.3 灾难恢复如何与 Rear 配合工作？

在系统正常运行期间，创建文件的备份并在恢复媒体上创建恢复系统。该恢复系统包含一个恢复安装程序。

如果系统已损坏，您可以更换受损的硬件（如果需要），从恢复媒体引导恢复系统，然后启动恢复安装程序。恢复安装程序会重建系统：首先，它会准备储存（分区、文件系统、安装点），然后从备份中恢复文件。最后，它会重新安装引导加载程序。

24.1.4 Rear 要求

要使用 Rear，至少需要两个相同的系统：运行生产环境的计算机，以及相同的测试计算机。举例来说，这里所说的“相同”是指您可以将一块网卡替换为使用相同内核驱动程序的另一块网卡。



警告：需要相同的驱动程序

如果某个硬件组件使用的驱动程序不同于生产环境中所用的驱动程序，则 Rear 不会将该组件视为相同。

24.1.5 Rear 版本更新

SUSE Linux Enterprise High Availability Extension 15 SP2 在以下包中随附了 Rear 版本 2.3: [rear23a](#).



注意：在更改日志中查找重要信息

有关 Bug 修复、不兼容性及其他问题的任何信息都可在包的更改日志中找到。如果需要重新验证灾难恢复过程，建议您另外也要审阅 Rear 的较新包版本。

您需要了解 Rear 的以下问题：

- 要能够在 UEFI 系统上实现灾难恢复，您至少需要 Rear 1.18.a 版和包 [ebiso](#)。只有此版本支持新的助手工具 [/usr/bin/ebiso](#)。此助手工具用于创建 UEFI 可引导 Rear 系统 ISO 映像。
- 如果您使用一个 Rear 版本实现的灾难恢复过程已通过测试并且功能完好，请不要更新 Rear。请保留该 Rear 包，并且不要更改您的灾难恢复方法！
- Rear 的版本更新是以独立包的形式提供的，这些包在设计上有意彼此冲突，目的是防止所安装的版本意外地被另一个版本替换。

在以下情况下，您需要全面重新验证现有的灾难恢复过程：

- 针对每个 Rear 版本更新。
- 手动更新 Rear 时。
- 针对 Rear 使用的每个软件。
- 更新底层系统组件（例如 [parted](#)、[btrfs](#) 及类似组件）时。

24.1.6 针对 Btrfs 的限制

如果您使用 Btrfs，请注意以下限制。

您的系统包括子卷，但不包括快照子卷

至少需要 Rear 版本 1.17.2.a。此版本支持重创建“正常的”Btrfs 子卷结构（不包括快照子卷）。

您的系统包括快照子卷



警告

无法照常使用基于文件的备份软件备份和恢复 Btrfs 快照子卷。

尽管 Btrfs 文件系统上的最新快照子卷几乎不占用任何磁盘空间（因为 Btrfs 具有写入时复制功能），但在使用基于文件的备份软件时，这些文件将作为完整文件进行备份。在备份中，这些文件的大小是其原始文件大小的两倍。因此，也就无法将快照恢复到它们以前在原始系统上的状态。

您的 SLE 系统需要匹配的 Rear 配置

例如，SLE12 GA、SLE12 SP1 和 SLE12 SP2 中的设置具有数个不同的不兼容 Btrfs 默认结构。因此，使用匹配的 Rear 配置文件至关重要。请参见示例文件 `/usr/share/rear/conf/examples/SLE12*-btrfs-example.conf`。

24.1.7 方案和备份工具

Rear 能够创建可从本地媒体（例如硬盘、闪存盘、DVD/CD-R）或通过 PXE 引导的灾难恢复系统（包括一个系统特定的恢复安装程序）。可以根据例 24.1 所述，将备份数据储存在网络文件系统中，如 NFS。

Rear 不会替换文件备份，而是对它进行补充。默认情况下，Rear 支持常规 `tar` 命令和若干第三方备份工具（例如 Tivoli Storage Manager、QNetix Galaxy、Symantec NetBackup、EMC NetWorker 或 HP DataProtector）。有关将 Rear 与用作备份工具的 EMC NetWorker 配合使用的示例配置，请参见例 24.2。

24.1.8 基本步骤

要在发生灾难时使用 Rear 成功进行恢复，需要执行以下基本步骤：

设置 Rear 和您的备份解决方案

这会涉及到一些任务，例如，编辑 Rear 配置文件、调整 Bash 脚本，以及配置您要使用的备份解决方案。

创建恢复安装系统

在要保护的系统正常运行期间，使用 `rear mkbackup` 命令创建文件备份，并生成包含特定于系统的 Rear 恢复安装程序的恢复系统。

测试恢复过程

每当使用 Rear 创建灾难恢复媒体后，都要全面测试灾难恢复过程。所用测试计算机上的硬件必须与生产环境中的硬件相同。有关细节，请参见第 24.1.4 节“Rear 要求”。

从灾难中恢复

灾难发生后，更换任何受损的硬件（如果需要）。然后，引导 Rear 恢复系统，并使用 `rear recover` 命令启动恢复安装程序。

24.2 设置 Rear 和您的备份解决方案

要设置 Rear，您至少需要编辑 Rear 配置文件 `/etc/rear/local.conf`，此外可以根据需要编辑属于 Rear 框架一部分的 Bash 脚本。

具体而言，您需要定义 Rear 应该执行的以下任务：

- **当您的系统是通过 UEFI 引导时：** 如果您的系统是通过 UEFI 引导加载程序引导的，请安装包 `ebiso` 并将下行添加到 `/etc/rear/local.conf` 中：

```
ISO_MKISOFS_BIN=/usr/bin/ebiso
```

- **如何备份文件以及如何创建和储存灾难恢复系统：** 这需要在 `/etc/rear/local.conf` 中配置。
- **需要重建的确切对象（分区、文件系统、安装点，等等）：** 这可以在 `/etc/rear/local.conf` 中定义（例如，要排除哪些对象）。要重建非标准系统，您可能需要增强 Bash 脚本。
- **恢复过程的工作方式：** 要更改 Rear 生成恢复安装程序的方式，或者要调整 Rear 恢复安装程序执行的操作，您需要编辑 Bash 脚本。

要配置 Rear，请将您的选项添加到 `/etc/rear/local.conf` 配置文件中（以前的配置文件 `/etc/rear/sites.conf` 已从包中去除。但是，如果您有来自以前设置环境中的此文件，Rear 会继续使用该文件）。

所有 Rear 配置变量及其默认值都在 `/usr/share/rear/conf/default.conf` 中设置。 `examples` 子目录中提供了一些用户配置（例如， `/etc/rear/local.conf` 中设置的项目）的示例文件（ `*example.conf` ）。请在 Rear 手册页中了解详细信息。

您应该使用一个匹配的示例配置文件作为模板，然后根据需要对其进行调整，以此创建您的特定配置文件。从数个示例配置文件中复制各选项，然后将其粘贴到与特定系统匹配的特定 `/etc/rear/local.conf` 文件中。请不要使用原始的示例配置文件，因为这些文件提供了可能用于特定设置的变量的概述。

例 24.1：使用 NFS 服务器储存文件备份

Rear 可在多种情境下使用。以下示例使用 NFS 服务器作为文件备份的储存。

1. 根据 《SUSE Linux Enterprise Server 15 SP2 管理指南》 (<https://documentation.suse.com/sles/15-SP2/html/SLES-all/cha-nfs.html>) 中所述，使用 YaST 设置 NFS 服务器。
2. 在 `/etc/exports` 文件中定义 NFS 服务器的配置。确保 NFS 服务器上的目录（要储存备份数据的位置）具有适当的装入选项。例如：

```
/srv/nfs *([...],rw,no_root_squash,[...])
```

将 `/srv/nfs` 替换为 NFS 服务器上备份数据的路径，并调整装入选项。您可能需要 `no_root_squash`，因为 `rear mkbackup` 命令以 `root` 身份运行。

3. 调整各个 `BACKUP` 参数（在配置文件 `/etc/rear/local.conf` 中），以便 Rear 将文件备份储存在相应的 NFS 服务器上。已安装系统中的 `/usr/share/rear/conf/examples/SLE*-example.conf` 下提供了示例。

例 24.2：使用 EMC NETWORKER 等第三方备份工具

要使用第三方备份工具代替 `tar`，您需要在 Rear 配置文件中进行相应的设置。

以下是 EMC NetWorker 的示例配置。将此配置片段添加到 `/etc/rear/local.conf` 中，并根据您的设置进行调整：

```
BACKUP=NSR
OUTPUT=ISO
BACKUP_URL=nfs://host.example.com/path/to/rear/backup
OUTPUT_URL=nfs://host.example.com/path/to/rear/backup
```

```
NSRSERVER=backupserver.example.com
RETENTION_TIME="Month"
```

24.3 创建恢复安装系统

根据第 24.2 节所述配置 Rear 后，使用以下命令创建恢复安装系统（包括 Rear 恢复安装程序和文件备份）：

```
rear -d -D mkbackup
```

该命令将执行以下步骤：

1. 分析目标系统并收集信息，尤其是有关磁盘布局（分区、文件系统、安装点）和引导加载程序的信息。
2. 使用第一步收集的信息创建一个可引导恢复系统。生成的 Rear 恢复安装程序特定于在发生灾难时要保护的系统。使用该安装程序只能重建这个特定的系统。
3. 调用配置的备份工具来备份系统和用户文件。

24.4 测试恢复过程

创建恢复系统之后，在具有相同硬件的测试计算机上测试恢复过程。另请参见第 24.1.4 节“Rear 要求”。确保测试计算机已正确设置，并可替代主计算机。



警告：在相同硬件上执行全面测试

必须在计算机上全面测试灾难恢复过程。请定期测试恢复过程，确保一切按预期运行。

过程 24.1：在测试计算机上执行灾难恢复

1. 将您在第 24.3 节中创建的恢复系统刻录到 DVD 或 CD 中，以创建恢复媒体。或者，您可以通过 PXE 使用网络引导。
2. 从恢复媒体引导测试计算机。

3. 从菜单中选择恢复。
4. 以 `root` 用户身份登录（无需密码）。
5. 输入以下命令启动恢复安装程序：

```
rear -d -D recover
```

有关在此过程中 Rear 所执行的步骤的细节，请参见[恢复过程](#)。

6. 恢复过程完成后，检查是否已成功重创建系统，并且该系统可在生产环境中替代原始系统运作。

24.5 从灾难中恢复

如果灾难已发生，请根据需要更换任何受损的硬件。然后按照[过程 24.1](#)所述，使用已修复的计算机（或使用已经过测试可替代原始系统运作的相同计算机）继续操作。

`rear recover` 命令将执行以下步骤：

恢复过程

1. 恢复磁盘布局（分区、文件系统和安装点）。
2. 从备份中恢复系统和用户文件。
3. 恢复引导加载程序。

24.6 更多信息

- http://en.opensuse.org/SDB:Disaster_Recovery 
- [rear](#) 手册页
- </usr/share/doc/packages/rear/>

IV 附录

- A 查错 311
- B 命名约定 321
- C 群集管理工具（命令行） 322
- D 在没有 root 访问权的情况下运行群集报告 324

A 查错

用户可能会遇到奇怪而不易理解的问题，特别是刚开始尝试使用 High Availability 时。不过，有一些实用程序可用来仔细地观察 High Availability 的内部进程。本章将推荐各种解决方案。

A.1 安装及初始步骤

对安装包或使群集联机的过程中遇到的问题查错。

是否安装了 HA 包？

配置和管理群集所需的包位于 High Availability Extension 提供的 High Availability 安装模式中。

按《安装和设置快速入门》中所述，检查 High Availability Extension 是否已作为 SUSE Linux Enterprise Server 15 SP2 的扩展安装在每个群集节点上，以及每台计算机上是否安装了 High Availability 模式。

所有群集节点的初始配置是否相同？

如第 4 章“使用 YaST 群集模块”中所述，为了相互通讯，属于同一个群集的所有节点需要使用相同的 bindnetaddr、mcastaddr 和 mcastport。

检查 /etc/corosync/corosync.conf 中配置的通讯通道和选项是否对所有群集节点都相同。

如果使用加密通讯，请检查 /etc/corosync/authkey 文件是否在所有群集节点上都可用。

除 nodeid 以外的所有 corosync.conf 设置都必须相同；所有节点上的 authkey 文件都必须相同。

防火墙是否允许通过 mcastport 进行通讯？

如果用于群集节点之间通讯的 mcastport 由防火墙阻止，这些节点将无法相互可见。在分别按第 4 章“使用 YaST 群集模块”或《安装和设置快速入门》文章中所述使用 YaST 或引导脚本执行初始设置时，防火墙设置通常会自动调整。

要确保 mcastport 不被防火墙阻止，请检查每个节点上的防火墙设置。

每个群集节点上是否都启动了 Pacemaker 和 Corosync?

通常，启动 Pacemaker 也会启动 Corosync 服务。要检查这两个服务是否在运行，请使用以下命令：

```
root # crm cluster status
```

如果它们未运行，请执行以下命令将其启动：

```
root # crm cluster start
```

A.2 日志记录

可以在哪里找到日志文件？

Pacemaker 会将其日志文件写入 `/var/log/pacemaker` 目录。主要的 Pacemaker 日志文件是 `/var/log/pacemaker/pacemaker.log`。如果您找不到日志文件，请检查 `/etc/sysconfig/pacemaker` 中的日志记录设置（Pacemaker 自己的配置文件）。如果该配置文件中配置了 `PCMK_logfile`，Pacemaker 将使用此参数定义的路径。

如果您需要获得一份显示所有相关日志文件的群集级报告，请参见[如何创建包含所有群集节点分析报告？](#)以了解更多信息。

我启用了监视，但为什么日志文件中没有监视操作的任何跟踪信息？

除非发生错误，否则 `pacemaker-execd` 守护程序不会记录重复的监视操作。记录所有重现的操作会产生太多噪音。因此，只会每小时记录一次重现的监视操作。

我只收到了一条失败消息。有可能收到更多信息吗？

在命令中添加 `--verbose` 参数。如果多次执行该操作，调试输出会变得相当详细。请参见日志记录数据 (`sudo journalctl -n`) 以获得有用的提示。

如何获取所有节点和资源的概述？

使用 `crm_mon` 命令。下面显示了资源操作的历史记录（选项 `-o`）和处于不活动状态的资源（`-r`）：

```
root # crm_mon -o -r
```

状态改变时，显示内容会刷新（要取消，请按 `Ctrl-C`）。示例可能显示如下：

例 A.1：已停止的资源

```
Last updated: Fri Aug 15 10:42:08 2014
Last change: Fri Aug 15 10:32:19 2014
Stack: corosync
Current DC: bob (175704619) - partition with quorum
Version: 1.1.12-ad083a8
2 Nodes configured
3 Resources configured

Online: [ alice bob ]

Full list of resources:

my_ipaddress      (ocf:heartbeat:Dummy): Started bob
my_filesystem     (ocf:heartbeat:Dummy): Stopped
my_webserver      (ocf:heartbeat:Dummy): Stopped

Operations:
* Node bob:
  my_ipaddress: migration-threshold=3
    + (14) start: rc=0 (ok)
    + (15) monitor: interval=10000ms rc=0 (ok)
* Node alice:
```

<http://www.clusterlabs.org/pacemaker/doc/> 上《Pacemaker Explained》

(Pacemaker 说明) PDF 文件中的“[How are OCF Return Codes Interpreted?](#)”（如何解释 OCF 返回代码？）部分中介绍了三种不同的恢复类型。

如何查看日志？

要详细查看群集中的当前活动，请使用以下命令：

```
root # crm history log [NODE]
```

将 `NODE` 替换为您要检查的节点，或将它保留空白。有关更多信息，请参见第 A.5 节“历史记录”。

A.3 资源

如何清理我的资源？

使用以下命令：

```
root # crm resource list
crm resource cleanup rscid [node]
```

如果遗漏此节点，则资源将在所有节点上清除。更多信息可以在第 8.4.4 节“清理资源”中找到。

如何列出当前已知的资源？

使用命令 `crm resource list` 可以显示当前资源。

我配置了一个资源，但是它总是失败。为什么？

要检查 OCF 脚本，请使用 `ocf-tester`，例如：

```
ocf-tester -n ip1 -o ip=YOUR_IP_ADDRESS \
  /usr/lib/ocf/resource.d/heartbeat/IPaddr
```

对更多参数，请多次使用 `-o`。通过运行 `crm ra info AGENT` 可获取必需参数和可选参数的列表，例如：

```
root # crm ra info ocf:heartbeat:IPaddr
```

运行 `ocf-tester` 之前，请确保资源不受群集管理。

资源为什么不故障转移，为什么没有错误？

已终止的节点可能会被视为不干净。这样就必须屏蔽它。如果 STONITH 资源不可运行或不运行，则剩余的节点将等待屏蔽发生。屏蔽超时值通常比较大，因此可能需要很长一段时间才能看到问题的明显迹象（如有）。

另一种可能的解释是仅仅是不允许资源在此节点上运行。这可能是由于未“清理”过去发生的失败所致。或者可能是先前的管理操作（即，添加了分数为负值的位置约束）所致。例如，此类位置约束是通过 `crm resource migrate` 命令插入的。

我为什么从不知道资源将在何处运行？

如果资源没有位置约束，则其放置取决于（几乎）随机节点选择。建议您始终明确表示资源的首选节点。这并不意味着您需要指定所有资源的位置自选设置。一个自选设置就能满足一组相关（共置）资源的需要。节点自选设置类似如下：

```
location rsc-prefers-alice rsc 100: alice
```

A.4 STONITH 和屏蔽

我的 STONITH 资源为什么不启动？

启动（或启用）操作包括检查设备的状态。如果设备未就绪，STONITH 资源将无法启动。同时系统将要求 STONITH 插件生成主机列表。如果此列表为空，则运行无法关闭任何节点的 STONITH 资源将毫无意义。运行 STONITH 的主机的名称是从列表中滤除的，因为节点不能自我关闭。

如果要使用单主机管理设备（如无人值守设备），请确保不允许 STONITH 资源在应当屏蔽的节点上运行。使用无限负位置节点自选设置（约束）。群集会将 STONITH 资源移到其他可以启动的位置，但不会未通知您就移动。

为什么尽管我有 STONITH 资源，却没有发生屏蔽？

每个 STONITH 资源都必须提供主机列表。您可以手动将此列表插入 STONITH 资源配置，也可以从设备自身（例如，从输出名称）检索。这取决于 STONITH 插件的性质。`pacemaker-fenced` 使用该列表来查找可以屏蔽目标节点的 STONITH 资源。只有出现在该列表中的节点 STONITH 资源才能关闭（屏蔽）。

如果 `pacemaker-fenced` 在运行中的 STONITH 资源提供的任何主机列表中找不到该节点，它将询问其他节点上的 `pacemaker-fenced` 实例。如果目标节点未显示在其他 `pacemaker-fenced` 实例的主机列表中，则屏蔽请求将以超时在源节点上结束。

我的 STONITH 资源为什么会偶尔失败？

如果广播通讯量过大，电源管理设备可能会停止运行。隔开监视操作。如果只是偶尔（最好是从不）需要屏蔽，则每隔几小时检查一次设备状态就已足够。

另外，其中的一些设备可能会拒绝同时与多方通讯。如果在群集尝试测试状态时将终端或浏览器会话保持打开状态，则这可能会产生问题。

A.5 历史记录

如何从发生故障的资源中检索状态信息或日志？

使用 `history` 命令及其子命令 `resource`：

```
root # crm history resource NAME1
```

这只会返回给定资源的完整转换日志。不过，您也可以调查多个资源，在第一个资源名称的后面追加更多的资源名称即可。

如果您遵循了某种命名约定（请参见一节），使用 `resource` 命令可以更轻松地调查一组资源。例如，以下命令将调查以 `db` 开头的所有原始资源：

```
root # crm history resource db*
```

查看 `/var/cache/crm/history/live/alice/ha-log.txt` 中的日志文件。

如何减少历史输出？

`history` 命令有两个选项：

- 使用 `exclude`
- 使用 `timeframe`

`exclude` 命令可让您设置附加的正则表达式用于从日志中排除特定的模式。例如，以下命令将排除所有 SSH、systemd 和内核讯息：

```
root # crm history exclude ssh|systemd|kernel.
```

使用 `timeframe` 命令可将输出限制为特定的范围。例如，以下命令将显示 8 月 23 日 12:00 到 12:30 的所有事件：

```
root # crm history timeframe "Aug 23 12:00" "Aug 23 12:30"
```

如何储存“会话”以便日后检查？

当您遇到 bug 或者需要进一步检查的事件时，储存所有当前设置的做法非常有用。您可以将储存的文件发送给支持人员，或者使用 `bzless` 进行查看。例如：

```
crm(live)history# timeframe "Oct 13 15:00" "Oct 13 16:00"  
crm(live)history# session save tux-test  
crm(live)history# session pack  
Report saved in '/root/tux-test.tar.bz2'
```

A.6 Hawk2

替换自签名证书

要避免在首次启动 Hawk2 时收到有关自我签名证书的警告，请将自动创建的证书替换为您自己的证书或官方证书颁发机构 (CA) 签名的证书：

1. 将 `/etc/hawk/hawk.key` 替换为私用密钥。
2. 将 `/etc/hawk/hawk.pem` 替换为 Hawk2 应当提供的证书。

将文件的所有权更改为 `root:haclient` 并使文件可被组访问：

```
chown root:haclient /etc/hawk/hawk.key /etc/hawk/hawk.pem
chmod 640 /etc/hawk/hawk.key /etc/hawk/hawk.pem
```

A.7 杂项

如何在所有群集节点上运行命令？

使用 `pssh` 命令可完成此任务。如果需要，请安装 `pssh`。创建一个文件（例如 `hosts.txt`），将要访问的所有 IP 地址或主机名都收集在其中。确保可以使用 `ssh` 登录到 `hosts.txt` 文件中列出的每台主机。如果一切准备就绪，请执行 `pssh` 并使用 `hosts.txt` 文件（选项 `-h`）和交互模式（选项 `-i`），如此例所示：

```
pssh -i -h hosts.txt "ls -l /corosync/*.conf"
[1] 08:28:32 [SUCCESS] root@venus.example.com
-rw-r--r-- 1 root root 1480 Nov 14 13:37 /etc/corosync/corosync.conf
[2] 08:28:32 [SUCCESS] root@192.168.2.102
-rw-r--r-- 1 root root 1480 Nov 14 13:37 /etc/corosync/corosync.conf
```

我的群集状态是什么？

要检查群集的当前状态，请使用程序 `crm_mon` 或 `crm status` 之一。这将显示当前的 DC 以及当前节点已知的所有节点和资源。

为什么群集的多个节点看不到彼此？

这可能有几个原因：

- 先查看配置文件 `/etc/corosync/corosync.conf`。检查群集中每个节点的多路广播或单路广播地址是否相同（使用关键字 `mcastaddr` 在 `interface` 部分中查找）。
- 检查您的防火墙设置。
- 检查您的交换机是否支持多路广播或单路广播地址。
- 检查节点间的连接是否已断开。这通常是错误配置防火墙的结果。这也可能是节点分裂情况（其中群集已分区）的原因。

为什么不能装入 OCFS2 设备？

检查日志讯息 (`sudo journalctl -n`) 中是否包含以下行：

```
Jan 12 09:58:55 alice pacemaker-execd: [3487]: info: RA output: [...]
ERROR: Could not load ocfs2_stackglue
Jan 12 16:04:22 alice modprobe: FATAL: Module ocfs2_stackglue not found.
```

在此案例中，是因为缺少内核模块 `ocfs2_stackglue.ko`。请根据安装的内核安装包 `ocfs2-kmp-default`、`ocfs2-kmp-pae` 或 `ocfs2-kmp-xen`。

如何创建包含所有群集节点分析报告？

在 `crm` 外壳中，可以使用 `crm report` 创建报告。此工具将会编译：

- 群集范围内的日志文件，
- 包状态，
- DLM/OCFS2 状态，
- 系统信息，
- CIB 历史记录，
- 内核转储报告的分析（如果安装了 `debuginfo` 包）。

通常需要结合以下命令运行 `crm report`：

```
root # crm report -f 0:00 -n alice -n bob
```

该命令将提取主机 `alice` 和 `bob` 上从凌晨 0 点开始的所有信息，并在当前目录中创建名为 `crm_report-日期.tar.bz2` 的 `*.tar.bz2` 存档，例如 `crm_report-Wed-03-Mar-2012`。如果您只对特定时间段感兴趣，请使用 `-t` 选项添加结束时间。



警告：删除敏感信息

`crm report` 工具会尝试从 CIB 和 PE 输入文件去除所有敏感信息，但是，它并不是万能的。如果您还有更多敏感信息，请通过 `-p` 选项（请参见手册页）提供其他模式。日志文件以及 `crm_mon`、`ccm_tool` 和 `crm_verify` 输出不会被清理。

以任何方式共享数据之前，请检查存档并删除不想泄露的所有信息。

使用其他选项自定义命令执行。例如，如果您有一个 Pacemaker 群集，那么您肯定想添加选项 `-A`。如果还有一个用户（除 `root` 和 `hacluster` 以外）对该群集拥有权限，可使用 `-u` 选项并指定此用户。如果您有一个非标准 SSH 端口，请使用 `-X` 选项添加该端口（例如，如果端口为 3479，则使用 `-X "-p 3479"`）。要了解更多选项，请参见 `crm report` 的手册页。

在 `crm report` 分析完所有相关日志文件并创建目录（或存档）后，请检查日志文件中有无大写的 `ERROR` 字符串。位于报告顶层目录中的最重要的文件有：

`analysis.txt`

比较在所有节点上都应保持一致的文件。

`corosync.txt`

包含 Corosync 配置文件的副本。

`crm_mon.txt`

包含 `crm_mon` 命令的输出。

`description.txt`

包含您节点上的所有群集包版本。另有节点特定的 `sysinfo.txt` 文件。它会链接到顶层目录。

可以使用此文件作为模板来描述您遇到的问题，然后将它发布到 <https://github.com/ClusterLabs/crmsh/issues>。

members.txt

所有节点的列表

sysinfo.txt

包含所有相关包名称及其版本的列表。此外，还有一个不同于原始 RPM 包的配置文件列表。

节点特定的文件将存储在以节点名称命名的子目录中。其中包含相应节点的 /etc 目录副本。

如果您需要简化自变量，请在配置文件 /etc/crm/crm.conf 的 report 部分设置默认值。更多信息请参见手册页 man 8 crmsh_hb_report。

A.8 更多信息

有关 Linux 上的高可用性的更多信息（包括配置群集资源以及管理和自定义高可用性群集），请参见 <http://clusterlabs.org/wiki/Documentation>。

B 命名约定

本指南针对群集节点和名称、群集资源与约束使用以下命名约定。

群集节点

群集节点使用人名：

alice、bob、charlie、doro 和 eris

群集站点名称

群集站点按城市命名：

amsterdam、berlin、canberra、dublin、fukuoka、gizeh、hanoi 和 istanbul

群集资源

原始资源	无前缀
组	前缀 <u>g-</u>
克隆资源	前缀 <u>cl-</u>
可升级克隆	(以前称为多状态资源) 前缀 <u>ms-</u>

限制

顺序约束	前缀 <u>o-</u>
位置约束	前缀 <u>loc-</u>
共置约束	前缀 <u>col-</u>

C 群集管理工具（命令行）

High Availability Extension 附带了一套全面的工具，帮助您从命令行管理群集。本章主要介绍管理 CIB 中的群集配置和群集资源所需的工具。用于管理资源代理的其他命令行工具或用于调试设置（和查错）的工具在附录 A “查错” 中有所介绍。



注意：使用 crmsh

此工具仅供专家使用。通常，crm 外壳 (crmsh) 是推荐的群集管理工具。

以下列表提供了一些与群集管理相关的任务，并简要介绍了完成这些任务所使用的工具：

监视群集的状态

`crm_mon` 命令可用于监视您的群集状态和配置。其输出包括节点数、uname、UUID、状态、群集中配置的资源及其各自的当前状态。`crm_mon` 的输出可显示在控制台上或打印到 HTML 文件。当具有不包含状态部分的群集配置文件时，`crm_mon` 会按文件中所指定的方式创建节点和资源概览。有关对此工具的用法及命令语法的详细介绍，请参见 `crm_mon` 手册页。

管理 CIB

`cibadmin` 命令是用于操作 CIB 的低级管理命令。它可用于转储、更新和修改全部或部分 CIB，删除整个 CIB 或执行其他 CIB 管理操作。有关对此工具的用法及命令语法的详细介绍，请参见 `cibadmin` 手册页。

管理配置更改

`crm_diff` 命令可帮助您创建和应用 XML 增补程序。这对可视化两个版本的群集配置之间的更改或保存更改可能非常有用，以便以后使用 `cibadmin` 应用更改。有关对此工具的用法及命令语法的详细介绍，请参见 `crm_diff` 手册页。

操作 CIB 属性

您可以使用 `crm_attribute` 命令来查询和操作 CIB 中使用的节点属性和群集配置选项。有关对此工具的用法及命令语法的详细介绍，请参见 `crm_attribute` 手册页。

验证群集配置

crm_verify 命令可检查配置数据库 (CIB) 的一致性和其他问题。它可检查包含配置的文件或连接到运行中的群集。它可报告两类问题。虽然警告解决方法已经传达到管理员，但是必须先修复错误，High Availability Extension 才能正常工作。**crm_verify** 可帮助创建新的或已修改的配置。您可以本地复制运行的群集中的 CIB，编辑它，使用 **crm_verify** 验证它，然后使用 **cibadmin** 使新配置生效。有关对此工具的用法及命令语法的详细介绍，请参见 **crm_verify** 手册页。

管理资源配置

crm_resource 命令可在群集上执行各种资源相关的操作。它可以修改已配置资源的定义，启动和停止资源，删除资源或在节点间迁移资源。有关对此工具的用法及命令语法的详细介绍，请参见 **crm_resource** 手册页。

管理资源故障计数

crm_failcount 命令可查询指定节点上每个资源的故障计数。此工具还可用于重设置故障计数，同时允许资源在它多次失败的节点上再次运行。有关对此工具的用法及命令语法的详细介绍，请参见 **crm_failcount** 手册页。

管理节点的备用状态

crm_standby 命令可操作节点的备用属性。备用模式中的所有节点都不再具备主管资源的资格，并且必须移动那里的所有资源。备用模式对执行维护任务（如内核更新）非常实用。从节点删除备用属性，使之再次成为群集中完全处于活动状态的成员。有关对此工具的用法及命令语法的详细介绍，请参见 **crm_standby** 手册页。

D 在没有 root 访问权的情况下运行群集报告

所有群集节点都必须能通过 SSH 相互访问。`crm report`（用于查错）等工具和 Hawk2 的历史记录浏览器要求节点之间采用无口令 SSH 访问方式，否则它们只能从当前节点收集数据。

如果无口令 SSH `root` 访问不符合法规要求，可以使用一种变通方法来运行群集报告。该变通方法主要包括以下几个步骤：

1. 创建专用的本地用户帐户（用于运行 `crm report`）。
2. 为该用户帐户配置无口令 SSH 访问（最好是使用非标准 SSH 端口）。
3. 为该用户配置 `sudo`。
4. 以该用户的身份运行 `crm report`。

默认情况下，`crm report` 在运行时先以 `root` 身份尝试登录远程节点，如果无法登录，则以 `hacluster` 用户身份登录。但是，如果您的本地安全策略阻止使用 SSH 进行 `root` 登录，则对所有远程节点执行脚本将会失败。即使尝试以 `hacluster` 用户身份运行脚本也会失败，因为这是一个服务帐户，其外壳设置为 `/bin/false`，因此会阻止登录。创建专用的本地用户是对高可用性群集中的所有节点成功运行 `crm report` 脚本的唯一可行做法。

D.1 创建本地用户帐户

在下面的示例中，我们将通过命令行创建名为 `hareport` 的本地用户。口令可以是符合安全要求的任何值。或者，您也可以使用 YaST 创建用户帐户并设置口令。

过程 D.1：创建用于运行群集报告的专用用户帐户

1. 启动外壳，然后创建主目录为 `/home/hareport` 的用户 `hareport`：

```
root # useradd -m -d /home/hareport -c "HA Report" hareport
```

2. 为该用户设置口令：

```
root # passwd hareport
```

3. 根据提示输入该用户的口令两次。

! 重要：需要在每个群集节点上使用同一个用户

要在所有节点上创建相同的用户帐户，请在每个群集节点上重复上述步骤。

D.2 配置无口令 SSH 帐户

过程 D.2：为非标准端口配置 SSH 守护程序

默认情况下，SSH 守护程序与 SSH 客户端将在端口 `22` 上通讯和侦听。如果您的网络安全方针要求将默认 SSH 端口更改为编号较高的备用端口，您需要修改守护程序的配置文件 `/etc/ssh/sshd_config`。

1. 要修改默认端口，请在该文件中搜索 `Port` 一行，取消注释该行，然后根据需要进行编辑。例如，可将该行设置为：

```
Port 5022
```

2. 如果您的组织不允许 `root` 用户访问其他服务器，请在该文件中搜索 `PermitRootLogin` 项，取消注释该项，然后将它设置为 `no`：

```
PermitRootLogin no
```

3. 或者，通过执行以下命令，在该文件的末尾添加相应的行：

```
root # echo "PermitRootLogin no" >> /etc/ssh/sshd_config
root # echo "Port 5022" >> /etc/ssh/sshd_config
```

4. 修改 `/etc/ssh/sshd_config` 后，重新启动 SSH 守护程序以使新设置生效：

```
root # systemctl restart sshd
```

! 重要：需要在每个群集节点上使用相同的设置

在每个群集节点上重复上述 SSH 守护程序配置。

过程 D.3：为非标准端口配置 SSH 客户端

如果要在群集中的所有节点上进行 SSH 端口更改，一种有效的做法是修改 SSH 配置文件 `/etc/ssh/sshd_config`。

1. 要修改默认端口，请在该文件中搜索 `Port` 一行，取消注释该行，然后根据需要进行编辑。例如，可将该行设置为：

```
Port 5022
```

2. 或者，通过执行以下命令，在该文件的末尾添加相应的行：

```
root # echo "Port 5022" >> /etc/ssh/sshd_config
```



注意：只需在一个节点上进行设置

只需在要运行群集报告的节点上进行上述 SSH 客户端配置。

或者，您可以使用 `-X` 选项并结合自定义 SSH 端口来运行 `crm report`，甚至可以将 `crm report` 指定为默认使用自定义 SSH 端口。有关细节，请参见[过程 D.5 “使用自定义 SSH 端口生成群集报告”](#)。

过程 D.4：配置 SSH 共享密钥

您可以使用 SSH 访问其他服务器，系统不会要求您输入口令。这种访问方法乍看之下似乎并不安全，但其实是非常安全的，因为用户只能访问共享了其公共密钥的服务器。共享密钥必须以使用该密钥的用户身份来创建。

1. 使用您为了运行群集报告而创建的用户帐户登录某个节点（在上面的示例中，该用户帐户为 `hareport`）。
2. 生成新密钥：

```
hareport > ssh-keygen -t rsa
```

默认情况下，此命令将生成 2048 位密钥。密钥的默认位置为 `~/.ssh/`。系统会提示您对该该密钥设置一个通行口令。但请勿输入通行口令，因为要进行无口令登录，就不能对密钥设置通行口令。

3. 生成密钥后，将公共密钥复制到其他每个节点（包括您在其中创建了该密钥的节点）：

```
hareport > ssh-copy-id -i ~/.ssh/id_rsa.pub HOSTNAME_OR_IP
```

在该命令中，您可以使用每个服务器的 DNS 名称、别名或 IP 地址。在复制过程中，系统会要求您接受每个节点的主机密钥，并且您需要提供 `hareport` 用户帐户的口令（只需要输入这一次）。

4. 在所有群集节点上共享密钥后，使用无口令 SSH 来测试您是否能够以 `hareport` 用户的身份登录其他节点：

```
hareport > ssh HOSTNAME_OR_IP
```

您应该会自动连接到远程服务器，系统不会要求您接受证书或输入口令。



注意：只需在一个节点上进行设置

如果您打算每次都从同一个节点运行群集报告，则在这个节点上执行上述过程便已足够。否则，您需要在每个节点上重复上述过程。

D.3 配置 `sudo`

使用 `sudo` 命令可让普通用户在提供或不提供口令的情况下快速变成 `root` 并发出命令。可向所有 `root` 级命令或者特定的命令授予 `sudo` 访问权限。`Sudo` 通常使用别名来定义整个命令字符串。

要配置 `sudo`，请使用 `visudo`（不是 `vi`）或 YaST。



警告：不要使用 `vi`

要从命令行配置 `sudo`，必须以 `root` 身份使用 `visudo` 编辑 `sudoers` 文件。使用任何其他编辑器可能会导致语法或文件权限错误，进而阻止 `sudo` 运行。

1. 以 `root` 身份登录。
2. 要打开 `/etc/sudoers` 文件，请输入 `visudo`。

3. 查找以下类别：[Host alias specification](#)、[User alias specification](#)、[Cmdnd alias specification](#) 和 [Runas alias specification](#)。

4. 将以下几项添加到 `/etc/sudoers` 中的相应类别：

```
Host_Alias CLUSTER = alice,bob,charlie ❶  
User_Alias HA = hareport ❷  
Cmdnd_Alias HA_ALLOWED = /bin/su, /usr/sbin/crm report * ❸  
Runas_Alias R = root ❹
```

- ❶ 主机别名定义 sudo 用户有权在哪个服务器（或特定范围内的服务器）上发出命令。在主机别名中，可以使用 DNS 名称或 IP 地址，或者指定整个网络范围（例如 `172.17.12.0/24`）。要限制访问范围，应该仅指定群集节点的主机名。
- ❷ 使用用户别名可将多个本地用户帐户添加到单个别名。但是，在这种情况下，您可以避开创建别名步骤，因为系统只会使用一个帐户。在上例中，我们添加了为运行群集报告而创建的 `hareport` 用户。
- ❸ 命令别名定义该用户可执行的命令。如果您要限制非 root 用户在使用 `sudo` 时可以访问的项目，命令别名将十分有用。在这种情况下，`hareport` 用户帐户需要对 `crm report` 和 `su` 命令拥有访问权限。
- ❹ `runas` 别名指定命令的运行帐户。在本例中为 `root`。

5. 搜索以下两行：

```
Defaults targetpw  
ALL ALL=(ALL) ALL
```

由于这两行与我们要创建的设置相冲突，因此请将其禁用：

```
#Defaults targetpw  
#ALL ALL=(ALL) ALL
```

6. 查找 [User privilege specification](#)。

7. 定义上述别名后，现在可以添加以下规则：

```
HA_CLUSTER = (R) NOPASSWD:HA_ALLOWED
```

`NOPASSWORD` 选项确保用户 `hareport` 无需提供口令就能执行群集报告。

! 重要：需要在每个群集节点上使用相同的 `sudo` 配置

必须在群集中的所有节点上指定这项 `sudo` 配置。无需为 `sudo` 做出其他更改，并且无需重新启动任何服务。

D.4 生成群集报告

要使用上面配置的设置运行群集报告，需要以 `hareport` 用户的身份登录某个节点。要启动群集报告，请使用 `crm report` 命令。例如：

```
root # crm report -f 0:00 -n "alice bob charlie"
```

此命令将在指定的节点上提取从凌晨 0 点开始的所有信息，并在当前目录中创建名为 `pcmk-日期.tar.bz2` 的 `*.tar.bz2` 存档。

过程 D.5：使用自定义 SSH 端口生成群集报告

1. 使用自定义 SSH 端口时，请结合使用 `-X` 和 `crm report` 来修改客户端的 SSH 端口。例如，如果自定义 SSH 端口为 `5022`，则使用以下命令：

```
root # crm report -X "-p 5022" [...]
```

2. 要为 `crm report` 永久设置自定义 SSH 端口，请启动交互式 `crm` 外壳：

```
crm options
```

3. 输入以下内容：

```
crm(live)options# set core.report_tool_options "-X -oPort=5022"
```

术语表

AutoYaST

AutoYaST 是能自动安装一个或多个 SUSE Linux Enterprise 系统而无需用户干预的系统。

bindnetaddr (绑定网络地址)

Corosync 管理器应绑定的网络地址。

boothd (投票间守护程序)

Geo 群集中的每个参与群集和仲裁方都会运行一个服务，即 `boothd`。它连接到其他站点上运行的投票间守护程序，并交换连接性细节。

CCM (一致性群集成员资格, consensus cluster membership)

CCM 确定组成群集的节点并在群集中共享此信息。任何节点或法定票数的新增和丢失都由 CCM 提供。群集的每个节点上都运行 CCM 模块。

CIB (群集信息库, cluster information base)

表示全部群集配置和状态（群集选项、节点、资源、约束和彼此之间的关系）。它会以 XML 的格式写入并驻存在内存中。主 CIB 保留并在 **DC (指定的协调程序)** 上进行维护，并复制到其他节点。对 CIB 的常规读写操作通过主 CIB 进行排序。

contrack 工具

可与内核内连接跟踪系统交互，以便对 iptables 启用有状态包检测。High Availability Extension 使用此工具来同步群集节点之间的连接状态。

crmsh

命令行实用程序 crmsh 可用于管理群集、节点和资源。

有关详细信息，请参见第 8 章“配置和管理群集资源（命令行）”。

CRM (群集资源管理器, cluster resource manager)

负责协调高可用性群集中的所有非本地交互的管理实体。High Availability Extension 使用 Pacemaker 作为 CRM。CRM 是作为 `pacemaker-controld` 实现的。它与多个组件交互：自身节点和其他节点上的本地资源管理器、非本地 CRM、管理命令、屏蔽功能以及成员资格层。

Csync2

可用于在群集中的所有节点间（甚至在 Geo 群集间）复制配置文件的同步工具。

DC (指定的协调程序)

DC 是从群集中的所有节点选择出来的。如果当前没有 DC，或者当前的 DC 出于任何原因退出群集，则就会按此方式选择 DC。DC 是群集中唯一可以决定需要在整个群集执行更改（例如节点屏蔽或资源移动）的实体。所有其他节点都从当前 DC 获取他们的配置和资源分配信息。

DLM (分布式锁管理器, distributed lock manager)

DLM 协调群集文件系统的磁盘访问和管理文件锁定以提高性能和可用性。

DRBD

DRBD 是为构建高可用性群集而设计的块设备。®整个块设备通过专用网络镜像，且视作网络 RAID-1。

Geo 群集

由多个分布于不同地理位置的站点组成，每个站点一个本地群集。站点通过 IP 通讯。站点之间的故障转移由更高级别的实体投票间协调。Geo 群集需要应对有限网络带宽和高延迟问题。储存异步复制。

Geo 群集 (分散在不同地理位置的群集, geographically dispersed cluster)

参见 [Geo 群集](#)。

LRM (本地资源管理器, local resource manager)

本地资源管理器位于每个节点上的 Pacemaker 层与资源层之间。它是作为 `pacemaker-execd` 守护程序实现的。通过此守护程序，Pacemaker 可以启动、停止和监视资源。

mcastaddr (多路广播地址)

Corosync 管理器使用 IP 地址进行多路广播。IP 地址可以为 IPv4 或 IPv6。

mcastport (多路广播端口)

用于群集通讯的端口。

metro 群集

使用光纤通道连接所有站点、可跨越多个建筑物或数据中心的单个群集。网络延迟通常很短（对 20 英里左右的距离而言不到 5 毫秒）。储存频繁复制（镜像或同步复制）。

pacemaker-controld (群集控制器守护程序)

CRM 是作为 pacemaker-controld 守护程序实现的。每个群集节点上都有一个实例。系统会选出一个 pacemaker-controld 实例来充当主实例，从而实现所有群集决策制定的集中化。如果选出的 pacemaker-controld 进程（或运行该进程的节点）发生失败，则会建立一个新的进程。

RA (资源代理, resource agent)

脚本充当代理来管理资源（例如，启动、停止或监视资源）。High Availability Extension 支持不同类型的资源代理：有关细节，请参见第 6.3.2 节“支持的资源代理类”。

Rear (Relax and Recover)

创建灾难恢复图像的管理员工具集。

RRP (冗余环网协议, redundant ring protocol)

该协议支持使用多个冗余局域网来从部分或整体网络故障中恢复。这样，只要一个网络运行正常，群集通讯就仍可继续。Corosync 支持 Totem Redundant Ring Protocol。

SBD (STONITH 块设备, STONITH Block Device)

通过经由共享块储存（SAN、iSCSI、FCoE 等）交换讯息提供节点屏蔽机制。还可以在无磁盘模式下使用。需要在每个节点上安装一个硬件或软件检查包，以确保能真正停止行为异常的节点。

SFEX (共享磁盘文件排他性, shared disk file exclusiveness)

SFEX 在 SAN 上提供存储保护。

SPOF (单一故障点, single point of failure)

一旦群集中任何组件出现故障，则会导致整个群集出现故障。

STONITH

“Shoot the other node in the head”（关闭其他节点）的首字母缩写。它表示一种关闭行为异常的节点以避免其在群集中制造麻烦的屏蔽机制。在 Pacemaker 群集中，节点级别屏蔽的实现为 STONITH。为此，Pacemaker 随附了一个屏蔽子系统 `pacemaker-fenced`。

主动/主动、主动/被动

针对服务在节点上如何运行的一种概念。主动-被动方案表示一个或多个服务正在主动节点上运行，而被动节点则等待主动节点出现故障。主动-主动方案表示每个节点既是主动节点

同时也是被动节点。例如，该节点正在运行某些服务，但也可以接管其他节点中的其他服务。它相当于 DRBD 概念中的主要/次要节点和双重主要节点。

仲裁方

在 Geo 群集中有助于达成一致性决定（例如，站点间的资源故障转移）的其他实例。仲裁方是一台以特殊模式运行一个或多个投票间实例的计算机。

切换

根据需要有计划地将服务转移到群集中的其他节点。请参见[故障转移](#)。

单路广播

一种将消息发送到单个网络目标的技术。Corosync 支持多路广播和单路广播。在 Corosync 中，单路广播作为 UDP 单路广播 (UDPU) 实施。

多路广播

一种用于网络内一对多通讯的技术，可用于群集通讯。Corosync 支持多路广播和单路广播。

屏蔽

描述了防止隔离的或失败的群集成员访问共享资源的概念。有两类屏蔽：资源级别屏蔽和节点级别屏蔽。资源级别屏蔽可确保对给定资源的排它访问。节点级别屏蔽可彻底防止故障节点访问共享资源，并可防止资源在状态不明的节点上运行。这种屏蔽通常采用一种简单但却粗暴的方式来完成，即重设置或关闭节点。

并发性违规

资源本应只可在群集中的一个节点上运行，但实际上正在多个节点上运行。

投票间

用于在 Geo 群集的不同站点之间管理故障转移进程的实例。它的目标是让多站点资源在一个且只有一个的站点上保持活动。如果某个群集站点发生故障，则会使用被视为站点间故障转移域的所谓的“票据”来实现。

故障转移

指资源或节点在某台服务器上出现故障、受影响的资源在另一个节点上启动的情况。

故障转移域

经过命名的一组群集节点的子集，有资格在节点出现故障时运行群集服务。

本地群集

一个位置的单个群集（例如，位于一个数据中心内的所有节点）。网络延迟可以忽略。储存通常由所有节点同步访问。

法定票数

在群集中，如果群集分区具有多数节点（或投票），则将其定义为具有仲裁（是“具有法定票数的”）。法定票数准确地区分了一个分区。它是算法的组成部分，用于防止多个断开的分区或节点继续运行而导致数据和服务损坏（节点分裂）。法定票数是屏蔽的先决条件，而屏蔽随后确保法定票数确实是唯一的。

灾难

关键基础设施因自然因素、人为因素、硬件故障或软件 bug 而意外中断。

灾难恢复

灾难恢复是指在发生灾难后将业务功能恢复到正常、稳定状态的过程。

灾难恢复计划

在对 IT 基础设施产生最低影响的前提下，从灾难中恢复的策略。

现有群集

术语“现有群集”指的是任何包括至少一个节点的群集。现有群集具有定义通讯通道的基本 Corosync 配置，但它们不一定已有资源配置。

票据

Geo 群集中使用的一个组件。票据授予在特定群集站点上运行某些资源的权限。一张票据某个时间内只能由一个站点所拥有。资源可按依赖性绑定到特定票据。仅当站点有定义好的票据时，才会启动相应资源。反之亦然，如果删除了票据，将会自动停止依赖于该票据的资源。

策略引擎 (PE)

策略引擎是作为 `pacemaker-schedulerd` 守护程序实现的。需要群集转换时，`pacemaker-schedulerd` 会根据当前状态和配置，计算群集的下一预期状态。它会确定需要安排哪些操作来实现下一种状态。

群集

高性能群集是一组为更快获得结果而共享应用程序负载的计算机（实际或虚拟）。高可用性群集主要用于确保服务的最大可用性。

群集分区

当一个或多个节点与群集的剩余节点之间的通讯失败时，即会发生群集分区。群集中的各节点被分割成不同分区，但仍然处于活动状态。他们只可与同一分区的节点进行通讯，并不了解未连接的节点。由于无法确认其他分区的节点是否丢失，所以开发了一种节点分裂方案（另请参见[节点分裂](#)）。

节点

是群集成员并对用户不可见的任何计算机（实际或虚拟）。

节点分裂

一种将群集节点分为两个或多个互不了解的组的方案（通过软件或硬件故障）。STONITH防止节点分裂情况对整个群集产生不利影响。也称为“分区的群集”方案。

术语“节点分裂”还用于DRBD中，但在DRBD中，它表示两个节点包含不同的数据。

负载均衡

能让多个服务器参与同一个服务并执行相同任务。

资源

Pacemaker 已知的任何类型的服务或应用程序。例如，IP 地址、文件系统或数据库。

术语“资源”也适用于DRBD，表示使用通用连接进行复制的一组块设备。

E GNU 许可证

此附录包含 GNU 自由文档许可证版本 1.2。

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law. A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public. It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.