Red Hat Enterprise Linux 7 High Availability Add-On 参考

Red Hat Enterprise Linux 7 高可用附加组件参考文档

Red Hat Enterprise Linux 7 High Availability Add-On 参考

Red Hat Enterprise Linux 7 高可用附加组件参考文档

法律通告

摘要

《Red Hat High Availability Add-On 参考》提供有关在 Red Hat Enterprise Linux 7 中安装、配置和管理 Red Hat High Availability Add-On 的参考信息。

目录

第 1 章 Red Hat High Availability Add-On 配置及管理参考概述

本文档论述了使用 Pacemaker 的 Red Hat High Availability Add-On 支持的选项及功能。有关分步基本配置示例，请参考*《Red Hat High Availability Add-On 管理》*。

可使用 **pcs** 配置界面或 **pcsd** GUI 界面配置 Red Hat High Availability Add-On 集群。

1.1. Red Hat Enterprise Linux 7.1 中新的和更改的功能

本小节列出了 Red Hat Enterprise Linux 7.1 发行本及之后的版本中包含的 Red Hat High Availablity Add-On 新功能及更改的功能。

Red Hat Enterprise Linux 7.1 包括以下文档及功能更新和更改。

- 现在 **pcs resource cleanup** 命令可以为所有资源重置资源状态和失败计数（failcount），如第 5.11 节 “集群资源清除” 所述。
- 可为 **pcs resource move** 命令指定 **lifetime** 参数，如 第 7.1 节 “在集群中手动移动资源” 所述。
- 从 Red Hat Enteprise Linux 7.1 开始，可使用 **pcs acl** 命令为本地用户设定权限，以便使用访问控制列表（ACL）允许本地用户对集群配置的只读或读写访问。有关 ACL 的详情，请查看 第 3.3 节 “设定用户权限”。
- 对 第 6.2.3 节 “按顺序排列的资源集” 和 第 6.3 节 “资源节点共置（Colocation）” 进行广泛更新及澄清。
- 第 5.1 节 “创建资源” 记录了 **pcs resource create** 命令的 **disabled** 参数，使用该参数代表那些已创建但不会自动启动的资源。
- 第 3.1.5 节 “配置仲裁选项” 记录了新的 **cluster quorum unblock** 功能，该功能可防止在建立仲裁时等待所有集群。
- 第 5.1 节 “创建资源” 记录了 **pcs resource create** 命令的 **before** 和 **after** 参数，可使用这些参数配置资源组排序。
- 从 Red Hat Enterprise Linux 7.1 发行本开始，可将集群配置备份为 tarball 文件，并在所有节点中使用 **pcs config** 命令的 **backup** 和 **restore** 选项恢复集群配置。有关这个功能的详情，请查看 第 2.8 节 “备份和恢复集群配置”。
- 另外对整个文档进行了小的修改和说明。

1.2. 安装 Pacemaker 配置工具

可使用 **yum install** 命令安装 Red Hat High Availability Add-On 软件包以及 High Availability 频道中的所有可用 fence 代理。

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

另外，可使用下面的命令安装 Red Hat High Availability Add-On 软件包以及只有您需要的 fence 代理。

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

以下命令显示可用 fence 代理列表。

```
# rpm -q -a | grep fence
```

```
fence-agents-rhevm-4.0.2-3.el7.x86_64
fence-agents-ilo-mp-4.0.2-3.el7.x86_64
fence-agents-ipmilan-4.0.2-3.el7.x86_64
...
```

lvm2-cluster 和 **gfs2-utils** 软件包是 ResilientStorage 频道的一部分，可根据需要使用以下命令安装。

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

警告

安装 Red Hat High Availability Add-On 软件包后，应确定设定了软件更新首选项，以防止自动安装任何内容。在运行的集群中安装可造成意外行为。

1.3. 配置 iptables 防火墙允许集群组件运行

Red Hat High Availability Add-On 需要为进入的流量启用以下端口：

- TCP：端口 2224、3121、21064
- UDP：端口 5405
- DLM（如果使用附带 clvm/GFS2 的 DLM 锁管理器）：端口 21064

通过执行以下命令，使用 **firewalld** 守护进程工具启用这些端口。

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

1.4. 集群及 Pacemaker 配置文件

Red Hat High Availability Add-On 的配置文件为 **corosync.conf** 和 **cib.xml**。请勿直接编辑这些文件，而是使用 **pcs** 或 **pcsd** 界面进行编辑。

corosync.conf 文件提供 **corosync** 使用的集群参数，后者是 Pacemaker 所在集群管理器。

cib.xml 是一个 XML 文件，代表集群配置和集群中所有资源的当前状态。这个文件由 Pacemaker 的集群信息基地（CIB）使用。会自动在整个集群中同步 CIB 的内容。

第 2 章 pcs 命令行界面

pcs 命令行界面通过为 **corosync.conf** 文件 **cib.xml** 提供界面，从而控制和配置 **corosync** 及 Pacemaker。

pcs 命令一般格式如下。

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

2.1. pcs 命令

pcs 命令如下。

- **cluster**

  配置集群选项和节点。有关 **pcs cluster** 的详情请参考 第 3 章 *集群创建及管理*。

- **resource**

  创建和管理集群资源。有关 **pcs cluster** 命令的详情请参考 第 5 章 *配置集群资源*、第 7 章 *管理集群资源* 和 第 8 章 *高级资源类型*。

- **stonith**

  将 fence 设备配置为与 Pacemaker 一同使用。有关 **pcs stonith** 命令的详情请参考 第 4 章 *Fencing：配置 STONITH*。

- **constraint**

  管理资源限制。有关 **pcs constraint** 命令的详情请查看 第 6 章 *资源限制*。

- **property**

  设定 Pacemaker 属性。有关使用 **pcs property** 命令设定属性的详情请查看 第 10 章 *Pacemaker 集群属性*。

- **status**

  查看当前集群和资源状态。有关 **pcs status** 命令的详情请查看 第 2.5 节 “显示状态”。

- **config**

  以用户可读格式显示完整集群配置。有关 **pcs config** 命令的详情请查看 第 2.6 节 “显示完整集群配置”。

2.2. pcs 用法帮助显示

可使用 **pcs** 的 **-h** 选项显示 **pcs** 命令的参数及那些参数的描述。例如：以下命令显示 **pcs resource** 命令参数，这只是输出结果的一部分。

```
# pcs resource -h
Usage: pcs resource [commands]...
Manage pacemaker resources
Commands:
    show [resource id] [--all]
        Show all currently configured resources or if a resource is specified
```

```
        show the options for the configured resource.  If --all is specified
        resource options will be displayed

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

2.3. 查看原始集群配置

虽然无法直接编辑集群配置文件，但可使用 **pcs cluster cib** 命令查看原始集群配置。

可如 第 2.4 节 “将配置更改保存到文件中” 所述使用 **pcs cluster cib *filename*** 将原始配置保存到指定文件中。

2.4. 将配置更改保存到文件中

使用 **pcs** 命令时，可使用 **-f** 选项将配置更改保存到文件中，而不会影响活跃的 CIB。

如果之前已配置了集群，且已有一个活跃 CIB，则可使用以下命令保存原始 xml 文件。

```
pcs cluster cib filename
```

例如：可使用以下命令将原始 xml 从 CIB 保存到名为 **testfile** 的文件。

```
pcs cluster cib testfile
```

以下命令在 **testfile1** 中创建一个资源，但不会将该资源添加到当前运行的集群配置中。

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

可使用以下命令将 **testfile** 的当前内容 push 回 CIB 中。

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

2.5. 显示状态

可使用以下命令显示集群和集群资源状态

```
pcs status commands
```

如果未指定 *commands* 参数，这个命令会显示有关该集群和资源的所有信息。指定 **resources**、**groups**、**cluster**、**nodes** 或 **pcsd** 选项则可以只显示具体集群组件的状态。

2.6. 显示完整集群配置

使用以下命令显示当前集群的完整配置。

```
pcs config
```

2.7. 显示当前 pcs 版本

以下命令显示正在运行的 **pcs** 版本。

```
pcs --version
```

2.8. 备份和恢复集群配置

从 Red Hat Enterprise Linux Release 7.1 开始，可使用下面的命令将集群配置备份为 tarball 文件。如果未指定文件名，则会使用标准输出作为文件名。

```
pcs config backup filename
```

使用下面的命令中所有节点中使用备份文件恢复集群配置。如果没有指定文件名，则会使用标准输入。使用 **--local** 选项则只会恢复当前节点中的文件。

```
pcs config restore [--local] [filename]
```

第 3 章 集群创建及管理

本章论述了如何使用 Pacemaker 执行基本集群管理，其中包括创建集群、管理集群组件及显示集群状态。

3.1. 集群创建

请按照以下步骤创建运行的集群：

1. 认证组成集群的节点。
2. 配置和同步集群节点。
3. 在集群节点中启动集群服务。

以下小节论述了用来执行这些步骤的命令。

3.1.1. 认证集群节点

以下命令在集群节点中为 **pcs** 守护进程认证 **pcs**。

- 每个节点中的 **pcs** 管理员的用户名必须为 **hacluster**。建议在每个节点中使用相同的 **hacluster** 密码。
- 如果未指定用户名或密码，系统会在执行该命令时提示您为每个节点指定那些参数。
- 如果未指定任何节点，且之前运行过该命令，则这个命令会在所有所有使用 **pcs cluster setup** 命令指定的节点中认证 **pcs**。

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

将授权令牌保存在 **~/.pcs/tokens**（或 **/var/lib/pcsd/tokens**）文件中。

3.1.2. 配置并启动集群节点

以下命令配置集群配置文件，并将配置同步到指定节点中。

- 若指定 **--start** 选项，该命令还会在指定节点中启动集群服务。必要时还可单独使用 **pcs cluster start** 命令启动集群服务。
- 指定 **--local** 选项后，该命令就只在本地节点中执行更改。

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

以下命令在指定节点中启动集群服务。

- 如果指定 **--all** 选项，该命令会在所有节点中启动集群服务。
- 如果未指定任何节点，则只在本地节点中启动集群服务。

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

3.1.3. 为集群配置超时值

使用 **pcs cluster setup** 命令创建集群时，会将集群超时值设定为默认值，这适用于大多数集群配置。但如果您的系统需要不同的超时值，则可以使用 **pcs cluster setup** 选项（参看 表 3.1 “超时值选项”）修改这些数值。

表 3.1. 超时值选项

选项	描述
--token *timeout*	以毫秒为单位设定未收到令牌后多长时间宣布令牌丢失（默认为 1000 毫秒）
--join *timeout*	以毫秒为单位设定等待加入信息的时间（默认为 50 毫秒）
--consensus *timeout*	以毫秒为单位设定启动新一轮成员关系配置前等待达成共识的时间（默认为 1200 毫秒）
--miss_count_const *count*	设定重新传送前检查令牌信息接收情况的最多次数（默认为 5 条信息）
--fail_recv_const *failures*	指定构成新配置前，在应当收到信息但却没有收到时执行的令牌轮换次数（默认为 2500 次失败）

例如：下面的命令可创建集群 **new_cluster**，并将令牌超时值设定为 10000 毫秒（10 秒），将加入集群超时值设定为 100 毫秒。

```
# pcs cluster setup --name new_cluster nodeA nodeB --token 10000 --join 100
```

3.1.4. 配置冗余环协议（RRP）

使用 **pcs cluster setup** 命令创建集群时，可使用冗余环协议，通过为每个节点同时指定两个接口配置集群。使用默认 udpu 传送指定集群节点时，可指定 ring 0 地址，后接 ‘,’，然后是 ring 1 地址。

例如：下面的命令配置有两个节点（节点 A 和节点 B）的集群 **my_rrp_clusterM**。节点 A 有两个接口：**nodeA-0** 和 **nodeA-1**。节点 B 有两个接口： **nodeB-0** 和 **nodeB-1**。要使用 RRP 将这些节点配置成一个集群，请执行以下命令。

```
# pcs cluster setup --name my_rrp_cluster nodeA-0,nodeA-1 nodeB-0,nodeB-1
```

有关在集群中使用 **udp** 传输配置 RRP 的详情，请查看 **pcs cluster setup** 的帮助页面。

3.1.5. 配置仲裁选项

Red Hat Enterprise Linux High Availability Add-On 集群使用 **votequorum** 服务避免出现裂脑（split-brain）的情况。会为集群的每个系统都分配大量投票，并只有在出现大多数投票时才允许执行集群操作。必须将这个服务上传至所有节点，或者根本不上传至任何节点。如果将其上传至集群节点的子网中，则会出现意外结果。有关配置和操作 **votequorum** 服务的详情，请查看 **votequorum**(5) man page。

当您了解到集群处于投票数不足（inquorate）的情况，但仍想继续进行资源管理时，则可以使用下面的命令防止集群在建立仲裁时等待所有节点投票。

```
# pcs cluster quorum unblock
```

在使用 **pcs cluster setup** 命令创建集群时，可设定一些仲裁配置的特殊功能。有关这些选项的概述请查看 表 3.2 “仲裁选项”。

表 3.2. 仲裁选项

选项	描述

选项	描述
--wait_for_all	启用该选项后，只有在至少有一次可同时看到所有节点后，才会第一次建立 quroate 状态。
--auto_tie_breaker	启用后，在决定性情况下，集群可以接受同时有 50% 的节点失败。集群分区或一组仍与 **auto_tie_breaker_node** 中配置的 **nodeid**（如果没有配置，则为最低 **nodeid**）沟通的节点仍保持 quorate 状态。其他节点则将处于 inquorate 状态。
--last_man_standing	启用后，集群可动态重新计算 **expected_votes**，并在具体情况下进行仲裁。启用这个选项时必须启用 **wait_for_all**，并指定 **last_man_standing_window**。
--last_man_standing_window	集群丢失节点后，等待重新计算 **expected_votes** 和仲裁的时间（单位：毫秒）。

有关配置和使用这些选项的详情，请查看 **votequorum**(5) man page。

3.2. 管理集群节点

以下小节论述了用来管理集群节点的命令，其中包括启动和停止集群服务，以及添加和删除集群节点的命令。

3.2.1. 停止集群服务

以下命令在指定节点中停止集群服务。使用附带 **--all** 选项的 **pcs cluster start** 可在所有节点中定制集群服务。如果未指定任何节点，则只在本地节点中停止集群服务。

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

可使用以下命令强制停止本地节点中的集群服务，该命令执行 **kill -9** 命令。

```
pcs cluster kill
```

3.2.2. 启用和禁用集群服务

使用以下命令将集群服务配置为在指定节点启动时运行。

- 如果指定 **--all** 选项，该命令会在所有节点中启用集群服务。
- 如果未指定任何节点，则只在本地节点中启用集群服务。

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

使用以下命令将集群服务配置为在指定节点启动时不运行。

- 如果指定 **--all** 选项，该命令会在所有节点中禁用集群服务。
- 如果未指定任何节点，则只在本地节点中禁用集群服务。

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

3.2.3. 添加和删除集群节点

以下命令可在现有集群中添加新节点。这个命令还可以将集群配置文件 **corosync.conf** 同步到集群的所有节点中，包括新添加的节点。

```
pcs cluster node add node
```

以下命令可关闭指定节点，并在集群的其他节点中将其从集群配置文件 **corosync.conf** 中删除。有关从集群节点中完全删除与该集群有关的所有信息，并因此永久删除该集群的相关信息，请参考 第 3.4 节 “删除集群配置”。

```
pcs cluster node remove node
```

3.2.4. 待机模式

以下命令可让指定节点进入待机模式。该指定节点不再托管资源。目前节点中活跃的所有资源都将被移动到另一个节点中。如果指定 **--all**，则这个命令会让所有节点进入待机模式。

可在更新资源软件包时使用这个命令。还可以在测试配置时使用这个命令模拟恢复过程，而无需真的关闭某个节点。

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

以下命令将指定节点从待机模式中移除。运行这个命令后，该指定节点就可以托管资源。如果指定 **--all** 选项，则这个命令会将所有节点从待机模式中删除。

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

注：执行 **pcs cluster standby** 命令时，可添加对资源的限制，以防止这些资源在指定节点中运行。执行 **pcs cluster unstandby** 命令可删除这些限制。不一定要将这些资源移回指定节点，这些资源根据资源的最初配置情况在那个节点中运行。有关资源限制的详情，请参考 第 6 章 *资源限制*。

3.3. 设定用户权限

默认情况下，root 用户及所有组 **haclient** 中的用户都可以读取/写入该集群配置文件。从 Red Hat Enteprise Linux 7.1 开始，可使用 **pcs acl** 命令为本地用户设定权限，以便使用访问控制列表（ACL）允许本地用户对集群配置文件的只读或读写访问。

分两步为本地用户设定权限：

1. 执行 **pcs acl role create...** 命令创建 *角色*，并为那个角色定义权限。
2. 使用 **pcs acl user create** 命令为用户分配您创建的角色。

以下示例步骤为名为 **rouser** 的本地用户提供集群配置的只读访问。

1. 这个步骤要求在本地系统中有用户 **rouser**，同时用户 **rouser** 是组 **haclient** 的成员。

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

2. 使用 **enable-acl** 集群属性启用 Pacemaker ACL。

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

3. 使用只读权限为 cib 创建名为 **read-only** 的角色。

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

4. 在 pcs ACL 系统中创建用户 **rouser**，并为那个用户分配 **read-only** 角色。

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

5. 查看当前 ACL。

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

以下示例步骤为名为 **wuser** 的本地用户提供集群配置的写入访问。

1. 这个步骤要求在本地系统中有用户 **wuser**，同时用户 **wuser** 是组 **haclient** 的成员。

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

2. 使用 **enable-acl** 集群属性启用 Pacemaker ACL。

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

3. 使用写入权限为 cib 创建名为 **read-only** 的角色。

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

4. 在 pcs ACL 系统中常见用户 **wuser**，并为那个用户分配 **write-access** 角色。

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

5. 查看当前 ACL。

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

有关集群 ACL 的详情请查看 **pcs acl** 命令的帮助页面。

3.4. 删除集群配置

要删除所有集群配置文件，停止所有集群服务，然后永久删除集群，请使用以下命令。

警告

这个命令可永久移除已创建的集群配置。建议删除集群前运行 **pcs cluster stop**。

```
pcs cluster destroy
```

3.5. 显示集群状态

以下命令显示集群及集群资源的当前状态。

```
pcs status
```

可使用以下命令显示集群当前状态的信息子集。

以下命令显示集群状态，但不显示集群资源。

```
pcs cluster status
```

以下命令显示集群资源状态。

```
pcs status resources
```

第 4 章 Fencing：配置 STONITH

STONITH 是 Shoot-The-Other-Node-In-The-Head 的首字母缩写，该工具可保护数据，以防其被恶意节点或并行访问破坏。

如果某个节点没有反应，并不代表没有数据访问。能够 100% 确定数据安全的唯一方法是使用 SNOITH 隔离该节点，这样我们才能确定在允许从另一个节点访问数据前，该节点已确实离线。

SNOITH 还可在无法停止集群的服务时起作用。在这种情况下，集群使用 STONITH 强制整个节点离线，以便在其他位置安全地启动该服务。

4.1. 可用 STONITH（Fencing）代理

使用以下命令查看所有可用 STONITH 代理列表。可先指定一个过滤器，然后这个命令就只显示与该过滤器匹配的 STONITH 代理。

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

4.2. Fencing 设备的常规属性

注意

可按照设定常规资源的方式设定 **target-role**，以便禁用 fencing 设备/资源。

注意

为防止在具体节点中使用 fencing 设备，可如预期采用位置限制。

表 4.1 “Fencing 设备的常规属性” 论述了可为 fencing 设备设定的常规属性。有关为具体 fencing 设备设定 fencing 属性的详情，请参考 第 4.3 节 “显示具体设备 Fencing 选项”。

注意

有关更多高级 fencing 配置属性的详情，请参考 第 4.9 节 “附加 Fencing 配置选项”。

表 4.1. Fencing 设备的常规属性

项	类型	默认值	描述
stonith-timeout	time	60s	在每个 stonith 设备中等待 STONITH 动作完成的时间，可覆盖 **stonith-timeout** 集群属性。
priority	整数	0	stonith 资源的优先权。这些设备将按照优先权从高到低的顺序排列。

项	类型	默认值	描述
pcmk_host_map	字符串		为不支持主机名的设备提供的主机名和端口号映射。例如：**node1:1;node2:2,3** 让集群的节点 1 使用端口 1，节点 2 使用端口 2 和 3。
pcmk_host_list	字符串		这台设备控制的机器列表（自选，**pcmk_host_check=static-list** 除外）。
pcmk_host_check	字符串	dynamic-list	如何确定该设备控制的机器。允许值为：**dynamic-list**（查询该设备），**static-list**（检查 **pcmk_host_list** 属性），none（假设每台设备都可以隔离所有机器）。

4.3. 显示具体设备 Fencing 选项

使用以下命令查看指定 STONITH 代理的选项。

```
pcs stonith describe stonith_agent
```

例如：以下命令显示使用 telnet/SSH 的 APC 的隔离代理选项。

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

4.4. 创建 Fencing 设备

使用以下命令创建 stonith 设备

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

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

如果为几个节点使用单一 fence 设备，请为每个节点使用不同的端口，但不需要为每个节点分别创建设备。相反，可使用 **pcmk_host_map** 选项定义每个节点使用的端口。例如：使用以下命令创建名为 **myapc-west-13** 的单一 fencing 设备，该设备使用的 APC 电源开关为 **west-apc**，同时节点 **west-13** 使用端口 15。

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

但在下面的示例中则使用 APC 电源开关 **west-apc** 隔离使用端口 15 的节点 **west-13**、使用端口 17 的节点 **west-14**、使用端口 18 的节点 **west-15**、以及使用端口 19 的节点 **west-16**。

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

4.5. 使用 unfencing 配置基于存储的 Fence 设备

创建 SAN/存储 fence 设备时（即使用非基于电源的 fencing 代理），必须在创建 **stonith** 设备时设定元数据选项 **provides=unfencing**。这样可保证隔离的节点在节点重启前处于未隔离状态，并在该节点中启动集群服务。

配置基于电源的 fence 设备时不一定要设定 **provides=unfencing** 元数据选项，因为该设备本身为节点提供电源以便其引导（并尝试重新加入集群）。在这种情况下的引导动作表示出现 unfencing 状态。

以下命令配置名为 **my-scsi-shooter**，使用 **fence_scsi** fence 代理为该设备启用 unfencing 的 stonith 设备。

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

4.6. 显示 Fencing 设备

以下命令显示所有当前配置的 fencing 设备。如果指定 *stonith_id*，则该命令只显示配置的 stonith 设备选项。如果指定 **--full** 选项，则显示所有配置的 stonith 选项。

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

4.7. 修改和删除 Fencing 设备

使用以下命令在当前配置的 fencing 设备中修改或添加选项。

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

使用以下命令从当前配置中删除 fencing 设备。

```
pcs stonith delete stonith_id
```

4.8. 使用 Fence 设备管理节点

可使用以下命令手动隔离某个节点。如果指定 **--off**，则会使用 **off** API 调用 stonith，从而关闭节点，而不是重启节点。

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

使用以下命令确定指定的节点目前是否已被关闭。

注意

如果指定的节点仍在运行集群软件或一般由集群控制的服务，则可能已出现数据受损/集群失败。

```
pcs stonith confirm node
```

4.9. 附加 Fencing 配置选项

表 4.2 “Fencing 设备的高级属性” 总结了可为 fencing 设备设定的附加属性。注：这些属性只能用于高级配置操作。

表 4.2. Fencing 设备的高级属性

项	类型	默认值	描述
pcmk_host_argument	字符串	重点	替换端口的备用数据。有些设备不支持标准端口参数，或可能提供额外参数。使用这个命令指定可替换具体设备参数以表示被隔离的机器。可使用参数值 none 让集群不要提供任何附加参数。
pcmk_reboot_action	字符串	reboot	**reboot** 命令以外的可替换命令。有些设备不支持标准命令，或可能提供附加命令。使用这个命令可指定可替换的设备具体命令，以便执行重启动作。
pcmk_reboot_timeout	time	60s	**stonith-timeout** 之外用于指定执行重启动作的可替换超时参数。有些设备比一般需要更多/少的时间。使用这个选项指定指定重启动作的可替换具体设备超时时间。
pcmk_reboot_retries	整数	2	在超时期限内重试 **reboot** 命令的最多次数。有些设备不支持多点连接。如果设备忙于其他任务，则操作可能会失败。如果还有时间，则 Pacemaker 将自动重新尝试该操作。使用这个选项更改 Pacemaker 在放弃前重新尝试重启的次数。
pcmk_off_action	字符串	关	可替换 **off** 的命令。有些设备不支持标准命令或可能提供附加命令。使用这个命令指定执行 off 动作的可替换具体设备命令。

项	类型	默认值	描述
pcmk_off_timeout	time	60s	指定备用超时用于 off 操作，而不是 **stonith-timeout**。有些设备需要比一般状况更多或更少的时间。使用这个选项为 off 操作指定备用、具体设备的超时。
pcmk_off_retries	整数	2	在超时期限内尝试 off 命令的最多次数。有些设备不支持多个连接。如果设备忙于另一个任务，操作可能会失败。如果还有时间，则 Pacemaker 将自动重试该操作。使用这个选项更改 Pacemaker 在放弃前重试 off 操作的次数。
pcmk_list_action	字符串	list	**list** 的可替换命令。有些设备不支持标准命令，或可能提供附加命令。使用这个命令指定可替换的设备具体命令，以便实施这个 list 操作。
pcmk_list_timeout	time	60s	指定备用超时用于 list 操作，而不是 **stonith-timeout**。有些设备需要比一般状况更多或更少的时间。使用这个选项为 list 操作指定备用、具体设备的超时。
pcmk_list_retries	整数	2	在超时期限内尝试 **list** 命令的最多次数。有些设备不支持多个连接。如果设备忙于另一个任务，操作可能会失败。如果还有时间，则 Pacemaker 将自动重试该操作。使用这个选项更改 Pacemaker 在放弃前重试 list 操作的次数。
pcmk_monitor_action	字符串	monitor	**monitor** 命令以外的可替换命令。有些设备不支持标准命令，或可能提供附加命令。使用这个命令可指定可替换的设备具体命令，以便执行 monitor 动作。
pcmk_monitor_timeout	time	60s	指定备用超时用于 monitor 操作，而不是 **stonith-timeout**。有些设备需要比一般状况更多或更少的时间。使用这个选项为 monitor 操作指定备用、具体设备的超时。
pcmk_monitor_retries	整数	2	在超时期限内尝试 **monitor** 命令的最多次数。有些设备不支持多个连接。如果设备忙于另一个任务，操作可能会失败。如果还有时间，则 Pacemaker 将自动重试该操作。使用这个选项更改 Pacemaker 在放弃前重试 monitor 操作的次数。
pcmk_status_action	字符串	状态	**status** 命令以外的可替换命令。有些设备不支持标准命令，或可能提供附加命令。使用这个命令可指定可替换的设备具体命令，以便执行 status 动作。
pcmk_status_timeout	time	60s	指定备用超时用于 status 操作，而不是 **stonith-timeout**。有些设备需要比一般状况更多或更少的时间。使用这个选项为 status 操作指定备用、具体设备的超时。
pcmk_status_retries	整数	2	在超时期限内尝试 status 命令的最多次数。有些设备不支持多个连接。如果设备忙于另一个任务，操作可能会失败。如果还有时间，则 Pacemaker 将自动重试该操作。使用这个选项更改 Pacemaker 在放弃前重试 status 操作的次数。

4.10. 配置 Fencing 等级

Pacemaker 通过名为 fencing 拓扑功能支持包含多个设备的 fencing 节点。要实施拓扑，请以常规方式创建独立的设备，然后再配置文件的 fencing-topology 部分，定义一个或多个 fencing 等级。

- 会按数字升序顺序尝试各个等级，从等级 1 开始。
- 如果某个设备失败，则会终止当前等级进程。不会对那个等级中的设备进行进一步操作，而是尝试下一个等级。
- 如果成功隔离所有设备，那么已成功应用该等级，且不会再试其他等级。
- 该等级通过（成功）后操作即完成。

使用以下命令为节点添加 fencing 等级。为这些设备给出用逗号分开的 stonith id 列表，代表该节点在那个等级中要尝试的 id。

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

以下命令列出当前配置的所有 fencing 等级。

```
pcs stonith level
```

在下面的示例中，为节点 **rh7-2** 配置两个 fence 设备：一个是名为 **my_ilo** 的 ilo fence 设备，一个是名为 **my_apc** 的 apc fence 设备。这些命令设定 fence 等级，以便设备 **my_ilo** 失败，且无法隔离该节点时，Pacemaker 会尝试使用设备 **my_apc**。这个实例还显示配置这些等级后的 **pcs stonith level** 命令输出结果。

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

以下命令删除指定节点和设备的隔离等级。如果没有指定节点或设备，则会删除该隔离等级。

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

以下命令清除指定节点或 stonith id 的隔离等级。如果未指定节点或 stonith id，则会删除所有隔离等级。

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

如果指定一个以上 stonith id，则必须使用逗号将其分开，且不能添加空格，如下例所示。

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

以下命令确认在 fence 等级中指定的所有 fence 设备和节点。

```
pcs stonith level verify
```

第 5 章 配置集群资源

本章论述了在集群中配置资源的信息。

5.1. 创建资源

使用以下命令创建集群资源。

```
pcs resource create resource_id standard:provider:type|type [resource options]
[op operation_action operation_options [operation_action  operation_options]...]
[meta meta_options...] [--clone clone_options |
--master master_options | --group group_name
[--before  resource_id | --after resource_id]] [--disabled]
```

指定 **--group** 选项后，会将该资源添加到在此命名的资源组中。如果该资源组不存在，这个命令会创建该资源组，并将这个资源添加到该组中。有关资源组的详情，请查看 第 5.5 节 “资源组”。

--before 和 **--after** 选项指定所添加资源与某个资源组中已存在资源间的相对位置。

指定 **--disabled** 选项表示不会自动启动该资源。

下面的命令创建一个名为 VirtualIP，使用 **ocf** 标准，**heartbeat** 提供程序，以及类型 **IPaddr2** 的资源。这个资源的浮动地址为 192.168.0.120，该系统会每 30 秒检查一次，确定该资源是否运行。

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

另外，可省略 *standard* 和 *provider* 字段，使用以下命令创建资源。这样就默认使用 **ocf** 标准以及 **heartbeat** 提供程序。

```
# pcs resource create VirtualIP IPaddr2 ip=192.168.0.120 cidr_netmask=24 op
monitor interval=30s
```

使用以下命令删除配置的资源。

```
pcs resource delete resource_id
```

例如：以下命令可删除资源 ID 为 **VirtualIP** 的现有资源。

```
# pcs resource delete VirtualIP
```

- 有关 **pcs resource create** 命令中 *resource_id*、*standard*、*provider* 和 *type* 字段的详情请参考 第 5.2 节 “资源属性”。
- 有关为每个资源定义资源参数的详情请参考 第 5.3 节 “具体资源参数”。
- 有关定义资源元数据选项以便集群使用，从而决定资源行为的详情，请参考 第 5.4 节 “资源元数据选项”。
- 有关定义某个资源中的操作的详情请参考 第 5.6 节 “资源操作”。
- 指定 **--clone** 创建克隆资源。指定 **--master** 创建主/辅资源。有关资源克隆以及有多种模式的资源的详情，请查看 第 8 章 *高级资源类型*。

5.2. 资源属性

5.2. 资源属性

为某个资源定义的属性会告知集群应在该资源中使用哪个脚本，在哪里找到该脚本，以及应遵循什么标准。有关这些属性的论述，请参考 表 5.1 “资源属性”。

表 5.1. 资源属性

项	描述
resource_id	资源名称
standard	脚本遵循的标准。允许值为：**ocf**, **service**, **upstart**, **systemd**, **lsb**, **stonith**
type	要使用的资源代理名称，例如 **IPaddr** 或者 **Filesystem**
provider	OCF 规范允许多个供应商提供同一资源代理。大多数 Red Hat 提供的代理使用 **heartbeat** 作为提供程序。

在 表 5.2 “显示资源属性的命令” 中总结了显示可用与创建资源的资源属性的命令

表 5.2. 显示资源属性的命令

pcs 显示命令	输出结果
pcs resource list	显示所有可用资源。
pcs resource standard	显示可用资源代理标准。
pcs resource providers	显示可用资源代理提供程序列表。
pcs resource list *string*	显示根据指定字符串过滤的可用资源列表。可使用这个命令显示根据标准名称、提供程序或类型过滤的资源。

5.3. 具体资源参数

可在每个资源中使用以下命令显示为那个资源设定的参数。

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

例如：以下命令显示为类型资源 **LVM** 设定的参数。

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

5.4. 资源元数据选项

除具体资源参数外，还可为任意资源配置附加资源选项。这些选项由集群用来决定资源的行为方式。有关此选项的详情请参考 表 5.3 “资源元数据选项”。

表 5.3. 资源元数据选项

项	默认值	描述
priority	**0**	如果无法保证所有资源都处于活跃状态，则集群会停止低优先权资源，以便让高优先权资源保持活跃状态。

项	默认值	描述
target-role	**Started**	该集群应让这个资源保持为何种状态？允许值为： * *Stopped* - 强制资源停止 * *Started* - 允许起源启动（在多状态资源的情况下，不能将其提升至主资源） * *Master* - 允许资源启动，并在适当时提升。
is-managed	**true**	集群是否允许启动和停止该资源？允许值为：**true**，**false**
resource-stickiness	0	表示该资源留在原有位置的倾向值。
requires	Calculated	表示在什么条件下启动该资源。 默认为 **fencing**，但在下属条件下除外。可能值为： * **nothing** - 该集群总是可以启动该资源。 * **quorum** - 只有在大多数配置的节点活跃的情况下该集群方可启动这个资源。如果 **stonith-enabled** 为 **false**，或资源的 **standard** 为 **stonith**，这就是默认值。 * **fencing** - 只有在大多数配置的节点活跃，*同时*已关闭所有失败或未知节点的情况下该集群方可启动这个资源。 * **unfencing** - 只有在大多数配置的节点活跃，*同时*已关闭所有失败或未知节点的情况下，该集群只能在 *未隔离* 的节点中启动该资源。如果已为 fencing 失败设定 **provides=unfencing stonith** 元数据选项，则这个值就是默认值。有关 **provides=unfencing stonith** 元数据选项的详情，请参考 第 4.5 节 “使用 unfencing 配置基于存储的 Fence 设备”。
migration-threshold	**INFINITY**（禁用）	将某个节点标记为无权托管这个资源前，这个资源可在该节点中出现失败的次数。有关配置 **migration-threshold** 选项的详情，请参考 第 7.2 节 “因宕机移动资源”。
failure-timeout	**0**（禁用）	与 **migration-threshold** 选项一同使用，代表将其视为没有发生过失败，并可能允许该资源返回其失败的节点前需要等待的秒数。有关配置 **failure-timeout** 选项的详情请参考 第 7.2 节 “因宕机移动资源”。
multiple-active	**stop_start**	如果发现该资源从未在一个以上节点中活跃时，集群该如何响应。允许值为： * **block** - 将该资源标记为 unmanaged。 * **stop_only** - 停止所有活跃实例并保留原状。 * **stop_start** -停止所有活跃实例并只在一个位置启动该资源。

请使用以下命令更改资源选项的默认值。

```
pcs resource defaults options
```

例如：以下命令可将 **resource-stickiness** 的默认值设定为 100。

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

省略 **pcs resource defaults** 中的 *options* 参数，则会为资源选项显示当前配置的默认值列表。以下示例显示在将 **resource-stickiness** 默认值重新设定为 100 时这个命令的输出结果。

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

无论是否已重新设定资源元数据选项的默认值，都可将某个特定资源的资源选项设定为与创建该资源时的默认值不同的数值。以下显示了用来为资源元数据选项指定值的 **pcs resource create** 命令格式。

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

例如：以下命令创建了 **resource-stickiness** 值为 50 的资源。

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120
cidr_netmask=24 meta resource-stickiness=5O
```

还可以使用以下命令为现有资源、组、克隆的资源或主资源设定资源元数据选项。

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

在下面的示例中有一个名为 **dummy_resource** 的现有资源。这个命令将 **failure-timeout** 元数据选项设定为 20 秒，以便该资源可在 20 秒内尝试在同一节点中重启。

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

执行这个命令后，可显示该资源值，以便确认将其设定为 **failure-timeout=20s**。

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

有关资源克隆元数据选项的详情请查看 第 8.1 节 “资源克隆”。有关资源主元数据选项，请查看 第 8.2 节 “多状态资源：有多个模式的资源”。

5.5. 资源组

集群的最常用元素之一是需要放在同一位置的资源组，按顺序启动，并以相反顺序停止。要简化这个配置，Pacemaker 支持组概念。

使用以下命令创建资源组，指定该组中包含的资源。如果该组不存在，这个命令可创建该组。如果该组存在，这个命令会在该组中添加附加资源。这些资源会按照这个命令中指定的顺序启动，并以与启动顺序的相反顺序停止。

```
pcs resource group add group_name resource_id [resource_id] ... [resource_id]
[--before resource_id | --after resource_id
```

可使用这个命令的 **--before** 和 **--after** 选项指定所添加资源与该组中现有资源间的相对位置。

还可以使用以下命令在创建该资源时，在现有组中添加新资源。会将您创建的资源添加到名为 *group_name* 的组中。

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

使用以下命令从组中删除资源。如果该组中没有资源，这个命令会将该组删除。

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

以下命令列出目前配置的资源组。

```
pcs resource group list
```

以下示例创建名为 **shortcut** 的资源组，该资源组包含现有资源 **IPaddr** 和 **Email**。

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

一个组所包含的资源数是无限的。资源的基本属性如下。

- 资源按照其指定顺序启动（在这个示例中，首先是 **Public-IP**，然后是 **Email**）。
- 资源按照其指定顺序的相反顺序停止（首先是 **Email**，然后是 **Public-IP**）。

如果该组中的某个无法在任何位置运行，那么在该资源后指定的资源将都无法运行。

- 如果 **Public-IP** 无法在任何位置运行，那么 **Email** 也不能。
- 如果 **Email** 无法在任何位置运行，则不会以任何方式影响 **Public-IP**。

显然，随着组的增大，创建资源组的配置工作会明显减少。

5.5.1. 组选项

资源组继承其所包含资源的选项：**priority**、**target-role**、**is-managed**。有关资源选项的详情，请参考 表 5.3 “资源元数据选项”。

5.5.2. 组粘性

在各组中附加的粘性（stickiness）表示某个资源留在其所在位置的倾向性。该组中的每个活跃资源的 stickiness 值相加极为该组的总粘性值。因此，如果默认 **resource-stickiness** 为 100，且该组有七个成员，其中五个是活跃成员，则该组作为一个整体，其倾向于留在当前位置的得分就是 500。

5.6. 资源操作

为保证资源正常工作，可在资源定义中添加监控操作。如果没有为资源指定监控操作，默认情况下 **pcs** 命令会创建一个监控操作，操作间隔由资源代理决定。如果资源代理未提供默认监控间隔，则 pcs 命令将创建一个以 60 秒为间隔的监控操作。

在 表 5.4 “操作属性” 中总结了资源监控操作的属性。

表 5.4. 操作属性

项	描述
id	该操作的特定名称。配置操作时系统会分配这个名称。
name	要执行的操作。常用值为：**monitor**, **start**, **stop**
interval	执行该操作的频率（秒）。默认值为：**0**，就是说从不执行。
timeout	宣布操作失败前等待的时间长度。如果发现您的系统中包含某个资源，该资源需要很长时间完成启动或停止，或在启动时执行非循环监控操作，所需时间超过了系统允许的宣布该操作失败所需时间，则可增大其默认值 20，或 "op defaults" 部分的 **timeout** 值。
on-fail	操作从未失败时采用的动作。允许值为： * **ignore** - 假设资源未失败 * **block** - 不对该资源执行任何进一步的操作。 * **stop** - 停止该资源且不在任何其他位置启动。 * **restart** - 停止该资源并重新启动（可能是在不同节点中）。 * **fence** - 使用 STONITH 隔离资源失败的节点。 * **standby** - 将*全部*资源从资源失败的节点中移除。 启用 STONITH 时的默认操作为 **stop**，否则为 **block**。所有其他操作默认为 **restart**。
enabled	如果为 **false**，则将该操作视为不存在。允许值为：**true**, **false**

可在创建资源时使用以下命令配置监控操作。

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

例如：以下命令创建附带监控操作的 **IPaddr2** 资源。新资源名为 **VirtualIP**，IP 地址 192.168.0.99，子网掩码为 **eth2** 中的 24。监控操作每隔 30 秒执行一次。

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99 cidr_netmask=24 nic=eth2 op monitor interval=30s
```

```
# pcs resource create my_address IPaddr2 ip=10.20.30.40 cidr_netmask=24 op monitor
```

也可以使用以下命令在现有资源中添加监控操作。

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

使用以下命令删除配置的资源操作。

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

注意

必须指定具体操作属性方可正确删除现有操作。

要更改监控选项值，可删除现有操作，然后添加新操作。例如：可使用以下命令创建 **VirtualIP**。

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

默认情况下这个命令创建三个操作。

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

请执行以下命令更改停止超时操作。

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

# pcs resource show VirtualIP
 Resource: VirtualIP (class=ocf provider=heartbeat type=IPaddr2)
  Attributes: ip=192.168.0.99 cidr_netmask=24 nic=eth2
  Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
              monitor interval=10s timeout=20s (VirtualIP-monitor-interval-10s)
              stop interval=0s timeout=40s (VirtualIP-name-stop-interval-0s-timeout-40s)
```

请使用以下命令获取监控操作全局默认值。

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

例如：使用以下命令为所有监控操作将全局 **timeout** 值设定为 240s。

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

要显示当前配置的监控操作默认值，请不要在执行 **pcs resource op defaults** 命令是指定任何选项。

例如：以下命令显示集群的默认监控操作，已为该操作配置 **timeout** 值 240s。

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

5.7. 显示配置的资源

请使用以下命令显示所有配置的资源。

```
pcs resource show
```

例如：如果已使用资源 **VirtualIP** 和 **WebSite** 配置系统，则 **pcs resource show** 命令的输出结果如下。

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

要显示所有配置的资源列表及为那些资源配置的参数，请使用 **pcs resource show** 命令的 **--full** 选项，如下例所示。

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

请使用以下命令显示某个资源的配置的参数。

```
pcs resource show resource_id
```

例如：以下命令显示目前为资源 **VirtualIP** 配置的参数。

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

5.8. 修改资源参数

请使用以下命令修改配置资源的参数。

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

下列命令显示资源 **VirtualIP** 已配置参数的初始值，该命令会更改 **ip** 参数值及后面的 update 命令值。

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

5.9. 多监控操作

可使用资源代理可支持的多监控操作数配置单一资源。这样就可以每分钟进行一次简单检查，逐渐以较长的间隔执行较深入的检查。

注意

配置多监控操作时，请保证在同一间隔中不能同时执行两个操作。

要为在不同等级中支持更深度检查的资源配置附加监控操作，可添加 **OCF_CHECK_LEVEL=*n*** 选项。

例如：如果配置以下 **IPaddr2** 资源，则会默认创建一个监控操作，即每隔 10 秒执行一次，超时值为 20 秒。

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

如果该虚拟 IP 支持深度 10 的不同检查，则以下命令可造成 Pacemaker 除每隔 10 秒执行一次常规虚拟 IP 检查外，还要每隔 60 秒执行更高级的监控检查。（注：不应也以 10 秒为间隔配置附加监控操作。）

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

5.10. 启用和禁用集群资源

以下命令启用 **resource_id** 指定的资源。

```
pcs resource enable resource_id
```

以下命令禁用 **resource_id** 指定的资源。

```
pcs resource disable resource_id
```

5.11. 集群资源清除

如果资源失败，则会在显示集群状态时出现失败信息。如果要处理那个资源，则可使用 **pcs resource cleanup** 命令清除那个失败状态。这个命令会重置资源状态和失败计数，让集群忘记资源操作历史记录，并重新探测当前状态。

以下命令清除 *resource_id* 指定的资源。

```
pcs resource cleanup resource_id
```

如果没有指定 *resource_id*，则这个命令会为所有资源重置资源状态和失败计数。

第 6 章 资源限制

通过为那个资源配置限制条件决定集群中某个资源的行为。可配置以下限制分类：

- **位置** 限制 — 位置限制决定某个资源可在哪个节点中运行。有关位置限制的论述请参考 第 6.1 节 “位置限制”。
- **顺序** 限制 — 顺序限制决定资源的运行顺序。有关数序限制的论述请参考 第 6.2 节 “顺序限制”。
- **节点共置（colocation）** 限制 — 节点共置限制决定相对于其他资源应放置资源的位置。有关节点共置限制的论述请参考 第 6.3 节 “资源节点共置（Colocation）”。

为简化配置一组限制，以便将一组资源放在一起以保证资源按顺序启动，并按相反的顺序停止，Pacemaker 支持资源组的概念。有关资源组的详情请参考 第 5.5 节 “资源组”。

6.1. 位置限制

位置限制可决定可运行这些资源的节点。可将位置限制配置为决定是否要避免在指定节点中运行某个资源。

在 表 6.1 “位置限制选项” 中总结了配置位置限制的选项。

表 6.1. 位置限制选项

项	描述
rsc	资源名称
node	节点名称
score	确定某个资源应该或避免在某个节点中运行的指数。 **INFINITY** 值将 "should" 改为 "must"；**INFINITY** 是资源位置限制的默认分值。

以下命令为资源创建位置限制以便首先使用指定的一个或多个节点。

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

以下命令为资源创建位置限制以避免指定的节点。

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

指定可运行某个资源的节点有另外两种方式：

- 选择加入集群 — 将集群配置为默认不允许任何资源在任何位置运行，然后为具体资源选择性启用节点。有关配置选择加入集群的步骤请参考 第 6.1.1 节 “配置“选择加入”集群”。
- 选择退出集群 — 将集群配置为默认允许所有资源在任意位置运行的集群，然后为资源创建位置限制，不允许其在具体节点中运行。有关配置选择退出集群的步骤请参考 第 6.1.2 节 “配置“选择退出”集群”。

应配置选择加入集群还是选择退出集群要视个人喜好及集群组成而定。如果您的大多数资源可在大多数节点中运行，那么可采用较简单的配置获得选择退出集群。另一方面，如果大多数资源只能在节点小子集中运行，那么选择加入配置应该更简单。

6.1.1. 配置“选择加入”集群

要创建选择加入集群，请将 **symmetric-cluster** 集群属性设定为 **false**，默认不允许资源在任意位置运行。

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

为独立资源启用节点。以下命令可配置位置限制，以便资源 **Webserver** 首选节点 **example-1**，资源 **Database** 首选节点 **example-2**，如果这两个资源的首选节点宕机，则可故障切换至节点 **example-3**。

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

6.1.2. 配置“选择退出”集群

要创建选择退出集群，请将 **symmetric-cluster** 集群属性设定为 **true**，以允许资源默认在任意位置运行。

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

然后使用以下命令生成与 第 6.1.1 节 “配置“选择加入”集群” 相当的配置。如果两个资源的首选节点宕机，则会故障切换至节点 **example-3**，因为每个节点都有一个隐含得分 0。

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

注：没有必要在这些命令中指定 INFINITY 数，因为那就是该得分的默认值。

6.2. 顺序限制

顺序限制决定资源运行顺序，可配置顺序限制以决定资源的启动和停止顺序。

使用以下命令配置顺序限制。

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

在 表 6.2 “顺序限制属性” 中总结了配置顺序限制的属性和选项。

表 6.2. 顺序限制属性

项	描述
resource_id	执行动作的资源名称。

项	描述
action	在资源中执行的动作。*action* 属性的可能值为： * **start** - 启动该资源。 * **stop** - 停止该资源。 * **promote** - 将某个资源从辅资源提升为主资源。 * **demote** - 将某个资源从主资源降级为辅资源。 如果未指定动作，则默认动作为 **start**。有关主资源和辅资源的详情请参考 第 8.2 节 “多状态资源：有多个模式的资源”。
kind option	如何加强限制。**kind** 选项的可能值如下： * **Optional** - 只可在两个资源均启动和/或停止时使用。有关 optional 排序的详情请参考 第 6.2.2 节 “顾问排序”。 * **Mandatory** - 总是使用该值（默认值）。如果指定的第一个资源停止或无法启动，则必须停止第二个指定的资源。有关 mandatory 排序的详情，请参考 第 6.2.1 节 “Mandatory 排序”。 * **Serialize** - 确定不会对一组资源同时执行 stop/start 操作。
symmetrical 选项	如果使用默认值 true，则以相反的顺序停止这些资源。默认值为：**true**

6.2.1. Mandatory 排序

强制排序表示如果指定的第一个资源未处于活跃状态，则无法运行指定的第二个资源。这是 **kind** 选项的默认值。采用这个默认值可保证您指定的第二个资源会在指定的第一个资源更改状态时有所响应。

- 如果指定的第一个资源正在运行，并已被停止，则指定的第二个资源也会停止（如果该资源正在运行）。
- 如果指定的第一个资源没有运行，且无法启动，则指定的第二个资源也会停止（如果该资源正在运行）。
- 如果指定的第一个资源已重启，同时指定的第二个资源正在运行，则指定的第二个资源会停止，然后重启。

6.2.2. 顾问排序

为另一个限制指定 **kind=Optional** 选项后，可将该限制视为自选，且只在同时停止和/或启动两个资源时才会产生影响。对第一个指定资源进行的任何更改都不会对第二个指定的资源产生影响。

以下命令为资源 **VirtualIP** 和 **dummy_resource** 配置一个顾问 排序限制。

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

6.2.3. 按顺序排列的资源集

一般是由管理员创建按顺序排列的资源链，比如资源 A 在资源 B 前启动，资源 B 在资源 C 前提动。可使用以下命令配置按顺序排列的资源链。这些资源会以指定的顺序启动。

可使用 **pcs constraint order set** 命令为一组或多组资源创建排序限制。

可使用 **pcs constraint order set** 命令的以下 **options** 参数为一组资源设定以下选项。

- **sequential**，可将其设定为 **true** 或 **false**，表示一组节点共置限制资源是否为按顺序排列的资源组。
- **require-all**，可将其设定为 **true** 或 **false**，表示是否该组中的所有资源都必须启用。
- **action**，可将其设定为 **start**、 **promote**、**demote** 或 **stop**，如 表 6.2 “顺序限制属性” 所述。
- **role**，可将其设定为 **Stopped**、**Started**、**Master** 或 **Slave**。有关多状态资源的详情，请查看 第 8.2 节 “多状态资源：有多个模式的资源”。

可使用 **pcs constraint order set** 命令的以下 **setoptions** 参数为一组资源设定以下限制选项。

- **id**，为定义的限制提供名称。
- **score**，表示这个限制的倾向性。有关这个选项的详情，请查看 表 6.3 “节点共置限制的属性”。

```
pcs constraint order set resource1 resource2 [resourceN]... [options] [set resourceX resourceY ... [options]] [setoptions [constraint_options]]
```

如果有三个名为 **D1**、**D2** 和 **D3** 的资源，则可使用以下命令将其配置为按顺序排列的资源集。

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

6.2.4. 从排序限制中删除资源

使用以下命令从任意排序限制中删除资源。

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

6.3. 资源节点共置（Colocation）

节点共置限制可根据另一个资源的位置决定某个资源的位置。

在两个资源间创建节点共置限制有一个很重要的负面作用：它会影响为某个节点分配资源的顺序。因为无法将资源 A 放在相对资源 B 的位置，除非您知道资源 B 在哪里。因此当创建节点共置限制时，关键是要考虑是应将资源 A 与资源 B 共置，还是将资源 B 与资源 A 共置。

另一个要记住的是在创建节点共置限制时，假设资源 A 与资源 B 共置，集群会在决定为资源 B 选择哪个节点时考虑资源 A 的属性。

以下命令可创建节点共置限制。

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

有关主资源和辅资源的详情请查看 第 8.2 节 “多状态资源：有多个模式的资源”。

在 表 6.3 “节点共置限制的属性” 中总结了配置节点共置限制的属性和选项。

表 6.3. 节点共置限制的属性

项	描述
source_resource	共置资源。如果对该限制不满意，集群会决定根本不允许该资源运行。

项	描述
target_resource	共置目标。集群将决定首先将这个资源放在哪里，然后决定在哪里防止源资源。
score	正数值代表资源应在同一节点中运行。负数值代表资源不应再同一节点中运行。默认值，即 + **INFINITY** 值代表 *source_resource* 必须作为 *target_resource* 在同一节点中运行。- **INFINITY** 值代表 *source_resource* 一定不能作为 *target_resource* 在同一节点中运行。

6.3.1. 强制放置

限制数值为 **+INFINITY** 或 **-INFINITY** 时会出现强制性放置。在这种情况下，如果对限制条件不满意，则不允许 *source_resource* 运行。对于 **score=INFINITY**，这包括 *target_resource* 不活跃的情况。

如果需要 **myresource1** 一直作为 **myresource2** 在同一机器中运行，则应添加以下限制：

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

因为使用了 **INFINITY**，如果 **myresource2** 无法在任何集群节点中运行（无论是什么原因），则不允许 **myresource1** 运行。

另外，您可能想要使用相反的配置，即在某个集群中，**myresource1** 无法在同一机器中作为 **myresource2** 运行。在这种情况下请使用 **score=-INFINITY**。

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

指定 **-INFINITY** 即可绑定该限制。因此如果最后能运行该资源的位置是 **myresource2** 所在位置，那么 **myresource1** 可能无法在任意位置运行。

6.3.2. 建议配置

如果强制配置是有关“一定要”和“一定不要”的配置，那么建议配置则属于“我更喜欢”的备用配置。对已得分大于 **-INFINITY** 和小于 **-INFINITY** 的限制，集群会尝试并满足您的要求，但如果备用配置要停止一些集群服务，则集群会忽略他们。建议节点共置限制可与配置的其他元素合并，以起到强制配置的作用。

6.3.3. 资源组共置

可使用 **pcs constraint colocation set** 命令为一组或多组资源创建节点共置限制。

可使用 **pcs constraint colocation set** 命令的以下 **options** 参数为一组资源设定以下选项。

- **sequential**，可将其设定为 **true** 或 **false**，表示一组节点共置限制资源是否为按顺序排列的资源组。
- **require-all**，可将其设定为 **true** 或 **false**，表示是否该组中的所有资源都必须启用。
- **action**，可将其设定为 **start**、 **promote**、**demote** 或 **stop**，如 表 6.2 “顺序限制属性” 所述。
- **role**，可将其设定为 **Stopped**、**Started**、**Master** 或 **Slave**。有关多状态资源的详情，请查看 第 8.2 节 “多状态资源：有多个模式的资源”。

可使用 **pcs constraint colocation set** 命令的以下 **setoptions** 参数为一组资源设定以下选项。

- **kind**，表示如何强制执行限制。有关这个选项的详情，请查看 表 6.2 “顺序限制属性”。
- **symmetrical**，表示停止资源的顺序。如果为 true（即默认选项），则会以相反的顺序停止资源。默认值为 **true**。

- **id**，为定义的限制提供名称。

以下命令可在一组或多组资源中创建节点共置限制。

```
pcs constraint colocation set resource1 resource2 [resourceN]... [options] [set resourceX resourceY ... [options]] [setoptions [constraint_options]]
```

6.3.4. 删除节点共置限制

使用以下命令删除附带 *source_resource* 的节点共置限制。

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

6.4. 显示限制

可使用一些命令显示已配置的限制。

以下命令列出所有当前位置、排序及节点共置限制。

```
pcs constraint list|show
```

以下命令列出所有当前位置限制。

- 如果指定 **resources**，则会显示每个资源的位置限制。这是默认行为。
- 如果指定 **nodes**，则会显示每个节点的位置限制。
- 如果指定具体资源或节点，则只会显示那些资源或节点的信息。

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

以下命令列出所有当前排序限制。如果指定 **--full** 选项，则限制内部限制 ID。

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

以下命令列出所有当前节点共置限制。如果指定 **--full** 选项，则限制内部限制 ID。

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

以下命令列出参考具体资源的限制。

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

第 7 章 管理集群资源

本章论述了可用来管理集群资源的各种命令，按照以下顺序提供信息。

- 第 7.1 节 “在集群中手动移动资源”
- 第 7.2 节 “因宕机移动资源”
- 第 7.4 节 “启用、禁用和禁止集群资源”
- 第 7.5 节 “禁用监控操作”

7.1. 在集群中手动移动资源

可覆盖该集群，并强制将资源从其当前位置移开。在两种情况您可能需要这样做：

- 维护节点时需要将所有在那个节点中运行的所有资源移至不同的节点
- 需要移动的单一资源

为了将在某个节点运行的所有资源移至不同的节点，则需要让该节点处于待用状态。有关让集群节点处于待用状态的详情，请查看 第 3.2.4 节 “待机模式”。

请使用下面的命令指定所定义节点的 *resource_id*，将资源移出其目前正在运行的节点。

```
pcs resource move resource_id
```

如果要指定要运行所移动资源的节点，请使用以下命令指定 **destination_node**。

```
pcs resource move resource_id destination_node
```

使用以下命令将该资源返回其最初运行的节点，让集群恢复正常操作。这样可删除 **move resource_id** 命令定义的限制。

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

注：执行 **pcs resource move** 命令时，会在该资源中添加限制，以防止其在指示的节点中运行。执行 **pcs resource clear** 命令时，会删除这些限制。这样就无需将该节点移回指示的节点。此时该资源运行的地点由资源最初配置的位置决定。

可自选为 **pcs resource move** 命令配置 **lifetime** 选项，以指定保留该限制的时间。可根据 ISO 8601 中规定的格式指定 **lifetime** 参数单位，该标准要求使用大写字母指定单位，比如 Y（代表年份），M（代表月份），W（代表星期），D（代表日期），H（代表小时），M（代表分钟），和 S（代表秒）。

为区别分钟单位（M）和月份单位（M），必须在分钟数前指定 PT 以代表分钟数。例如：**lifetime** 参数 5M 代表每隔五个月，而 **lifetime** 参数 PT5M 则代表每隔五分钟。

lifetime 参数是根据 **cluster-recheck-interval** 集群属性定义的间隔进行检查。默认情况下，这个数值是 15 分钟。如果您的配置要求使用这个参数进行更频繁的检查，则可使用下面的命令重新设置这个数值。

```
pcs property set cluster-recheck-interval=value
```

下面的命令可将资源 **resource1** 移动到节点 **example-node2**，并防止其在一小时三十分钟内被一会其最初运行的节点中。

```
pcs resource move resource1 example-node2 lifetime=PT1H30M
```

下面的命令可将资源 **resource1** 移动到节点 **example-node2**，并防止其在三十分钟内被移回其最初运行的节点中。

```
pcs resource move resource1 example-node2 lifetime=PT30M
```

有关资源限制的详情，请查看 第 6 章 *资源限制*。

7.2. 因宕机移动资源

创建资源时，可配置该资源，以便规定在失败多少次后（使用 **migration-threshold** 选项为该资源定义），将其移动到一个新节点。达到阈值后，就不再允许该节点运行失败的资源，除非：

- 管理员使用 **pcs resource failcount** 命令手动重置资源失败计数。
- 已达到该资源的 **failure-timeout** 值。

默认不定义阈值。

注意

为某个资源设定 **migration-threshold** 与配置资源迁移不同，后者是将该资源移至另一个位置，而不会丢失状态。

以下示例在名为 **dummy_resource** 的资源组合中添加迁移阈值 10，表示在十次失败后会将该资源移至一个新节点。

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

使用以下命令在默认值中为整个集群添加迁移阈值。

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

请使用 **pcs resource failcount** 命令确定资源的当前设备状态及限制。

迁移阈值概念有两个例外情况，在某个资源无法启动或无法停止时会出现这两种例外。启动操作失败会将 failcount 设定为 **INFINITY**，因此结果总是将资源立即移开。

停止操作失败稍有不同，且更严重。如果某个资源无法停止，且同时启用了 STONITH，那么集群就会隔离该节点，以便在其他位置启动该资源。如果未启用 STONITH，则该集群就无法继续运行该资源，也无法尝试着其他位置启动该资源，但仍会在失败超时后继续尝试停止该资源。

7.3. 由于连接更改而移动资源

将集群设定为在丧失外部连接后移动资源，这是一个两步的过程。

1. 在集群中添加 **ping** 资源。**ping** 资源使用同名的系统程序测试是否可连接到一组机器（由 DNS、主机名或 IPv4 地址指定），并使用所得结果维护节点属性 **pingd**。
2. 为资源配置位置限制，以便在连接丧失后将该资源移动到不同的节点。

表 5.1 “资源属性” 论述了可为 **ping** 资源设定的属性。

表 7.1. Ping 资源属性

项	描述
dampen	执行进一步更改要等待的时间（dampening）。这样可防止由于集群节点意识到连接丢失的时间稍有不同而造成资源在集群间不断跳转。
multiplier	将连接的 ping 节点数与这个值相乘得到一个分数。这在配置多个 ping 节点时有帮助。
host_list	为确定当前连接状态而要联络的机器。允许值包括可解析的 DNS 主机名、IPv4 和 IPv6 地址。

以下示例命令创建了 **ping** 资源，用来确认与 **www.example.com** 连接性的 。在实践中，可以确认网络网关/路由器的连接性。将 **ping** 资源配置为克隆，以便在所有集群节点中都可以运行。

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

以下示例为名为 **Webserver** 的现有资源配置了位置限制。这样就会在其目前所运行的主机无法 ping **www.example.com** 时，将 **Webserver** 资源移至能够 ping **www.example.com** 的主机中。

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

7.4. 启用、禁用和禁止集群资源

除 第 7.1 节 “在集群中手动移动资源” 所述 **pcs resource move** 命令外，还有其他不同的命令可用于控制集群资源行为。

使用以下命令可手动停止运行的资源，并防止集群再次启动该资源。根据其余的配置（限制、选项、失败等等），该资源可能仍保持启动状态。如果指定 **--wait** 选项，**pcs** 会等待 30 秒（或 ‘n’ 秒）以便资源停止。如果资源停止，则会返回数值 0，否则就返回数值 1。

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

可使用以下命令允许集群启动某个资源。该资源可能仍处于停止状态，具体要看其他配置。如果指定 **--wait** 选项，**pcs** 会等待 30 秒（或 ‘n’ 秒）以便资源启动。如果资源启动，则会返回数值 0，否则就返回数值 1。

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

使用以下命令防止资源在指定节点中运行。如果未指定节点，则防止其在当前节点中运行。

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

注：执行 **pcs resource ban** 命令时，会在该资源中添加限制，以防止其在指示的节点中运行。执行 **pcs resource clear** 命令时会删除这些限制。这样就无需将该节点移回指示的节点。此时该资源运行的地点由资源最初配置的位置决定。有关资源限制的详情，请参考 第 6 章 *资源限制*。

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

可使用 **pcs resource** 命令的 **debug-start** 参数强制在当前节点中启动指定的资源，忽略集群建议，并显示启动的资源的输出结果。这主要用于 debug 资源。在集群中启动资源通常由 Pacemaker 指定，而不是直接使用 **pcs** 命令。如果您的资源未启动，一般是由于资源的错误配置（参看系统日志）限制了资源的启动，或者该资源已被禁用。可使用这个命令测试资源配置，但一般不用于在集群中启动资源。

debug-start 命令的格式如下。

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

7.5. 禁用监控操作

停止循环监控的最简单方法是将其删除。但有时您可能只想暂时将其禁用。在这种情况下，可在操作定义中添加 **enabled="false"**。如果要恢复监控操作，请在操作定义中设定 **enabled="true"**。

7.6. 管理的资源

可将某个资源设定为非管理模式，即代表该资源仍在配置文件中，但 Pacemaker 不会管理该资源。

以下命令将所示资源设定为非管理模式。

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

以下命令将资源设定为管理模式，即默认状态。

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

可使用 **pcs resource manage** 或 **pcs resource unmanage** 命令指定资源组名称。该命令可在该组的所有资源操作，以便可使用单一命令管理或不管理某个组中的所有资源，然后分别管理所包含的资源。

第 8 章 高级资源类型

本章介绍了 Pacemaker 支持的高级资源类型。

8.1. 资源克隆

可以克隆某个资源，以便其在多个节点中处于活跃状态。例如：可以使用克隆的资源配置一个 IP 资源的多个实例，以便在整个集群中分布，保持节点平衡。可以克隆提供资源代理支持的任意资源。一个克隆包括一个资源或一个资源组。

注意

只有能够同时在多个节点中处于活跃状态的资源适合克隆。例如：不应克隆共享内存设备中挂载非集群文件系统（比如 **ext4**）的 **Filesystem** 资源。因为集群无法识别 **ext4** 分区，这个文件系统不适合同时在多个节点中进行读取/写入操作。

8.1.1. 创建及删除克隆的资源

使用以下命令同时创建并克隆某个资源。

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

克隆的名称将是 ***resource_id*-clone**。

不能使用单一命令创建并克隆资源组。

另外，还可以使用以下命令克隆之前创建的资源或资源组。

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

克隆的名称为 ***resource_id*-clone** 或 ***group_name*-clone**。

注意

只需要在一个节点中执行资源配置变更。

注意

配置限制时永远使用该组或克隆的名称。

创建资源克隆时，该克隆会沿用该资源的名称，并在该名称中附加 **-clone** 字样。以下命令可创建 **apache** 类型资源，名为 **webfarm**，并克隆该资源，将其命名为 **webfarm-clone**。

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

使用以下命令删除资源或资源组克隆。这不会删除该资源或资源组本身。

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

有关此选项的详情请查看 第 5.1 节 “创建资源”。

表 8.1 “资源克隆选项” 论述了可为克隆的资源指定的选项。

表 8.1. 资源克隆选项

项	描述
priority, target-role, is-managed	从克隆的资源中继承的选项，如 表 5.3 “资源元数据选项” 所述。
clone-max	需要多少资源副本方可启动。默认为该集群中的节点数。
clone-node-max	需要多少资源副本方可在单一节点中启动。默认值为 **1**。
notify	停止或启动克隆的副本时，应在开始操作前，以及操作成功后告知其他所有副本。可用值为：**false** 和 **true**。默认值为 **false**。
globally-unique	每个克隆副本的功能是否有所不同？允许值为：**false**，**true** 如果这个选项的值为 **false**，这些资源会在运行时有完全相同的行为，因此每台机器中只需要有一个克隆副本即可。 如果这个选项值为 **true**，一台机器中运行的克隆副本就与另一个实例不相当，无论该实例是在另一个节点中运行，还是在同一节点中运行。如果 **clone-node-max** 大于 1，则默认值为 **true**，否则默认值为 **false**。
ordered	是否应连续复制（而不是平行复制）。允许值为 **false** 和 **true**。默认值为 **false**。
interleave	更改排序限制行为（clones/masters 之间）以便在实例可在其同级别实例启动/停止时尽快启动/停止（而不是等待每个克隆中的每个实例都启动/停止）。允许值：**false** 和 **true**。默认值为 **false**。

8.1.2. 克隆限制

在大多数情况下，每个克隆在每个活跃集群节点中都有一个副本。但也可以将资源克隆的 **clone-max** 设定为小于该集群中的节点总数。在这种情况下，可使用资源位置限制选择集群应为哪些节点首先分配副本。这些限制的编写与那些用于常规资源的限制没有什么不同，只是必须使用该克隆的 id。

以下命令可为集群生成位置限制，以便首选将资源克隆 **webfarm-clone** 分配给 **node1**。

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

克隆的排序限制行为稍有不同。在下面的示例中，**webfarm** 会等到所有要启动的克隆副本都完成后方开始执行。只有在没有副本可启动时，方可防止 **webfarm** 启动。另外，克隆将在其停止前等待 **webfarm** 停止。

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

克隆与常规（或组）资源共置表示该资源可在有该克隆活跃副本的任意机器中运行。集群将根据该克隆所运行的位置，以及该资源自身位置属性选择副本。

还可以配置克隆共置。在此类情况下，将一组克隆允许的位置限制为克隆处于活跃状态或将处于活跃状态的节点。然后根据常规执行分配操作。

以下命令创建一个节点共置限制，以保证资源 **webfarm** 在其克隆的同一节点中运行。

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

8.1.3. 克隆粘性

要取得稳定的分配模式，默认情况下克隆会有一些粘性。如果没有为 **resource-stickiness** 分配任何数值，则该克隆将使用数值 1。请使用较小的数值，这样对其他资源的 score 计算影响较小，但也足以防止 Pacemaker 在集群中进行不必要的副本移动。

8.2. 多状态资源：有多个模式的资源

多状态资源是特殊的克隆资源。它们允许实例使用处于两个操作模式之一，即 **Master** 和 **Slave**。模式的名称没有任何特殊含义，只是在启动实例时，必须使用 **Slave** 状态。

可使用下面的一个命令创建作为主/辅克隆的资源。

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

master/slave 克隆的名称应为 ***resource_id*-master**。

另外，可使用以下命令，根据之前创建的资源或资源组创建 master/slave 资源：使用这个命令时，可指定 master/slave 克隆名称。如果未指定名称，则 master/slave 克隆的名称将为 ***resource_id*-master** 或 ***group_name*-master**。

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

有关此选项的详情请查看 第 5.1 节 “创建资源”。

在 表 8.2 “多状态资源的属性” 中论述了可为多状态资源指定的选项。

表 8.2. 多状态资源的属性

项	描述
id	多状态资源的名称
priority, **target-role**, **is-managed**	请查看 表 5.3 “资源元数据选项”。
clone-max, **clone-node-max**, **notify**, **globally-unique**, **ordered**, **interleave**	请查看 表 8.1 “资源克隆选项”。
master-max	可将多少资源副本提升至 **master** 状态；默认 1。
master-node-max	在单一节点中可将多少资源副本推广至 **master** 状态；默认 1。

8.2.1. 监控多状态资源

如只要为 master 添加监控操作，可在该资源中添加附加监控操作。注：资源中的每个监控操作必须采用不同的间隔。

以下示例以 11 秒为间隔在 master 资源中为 **ms_resource** 设置监控操作。这个监视器资源是默认的 10 秒间隔监控操作之外的资源。

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

8.2.2. 多状态限制

在大多数情况下，多状态资源可在每个活跃集群节点中有单一副本。如果不是这种情况，则可使用资源位置限制指明首先为其分配副本的集群节点。这些限制与常规资源的编写没有任何不同。

有关资源位置限制的详情，请参考 第 6.1 节 “位置限制”。

可创建节点共置限制，用来指定该资源是 master 还是 slave 资源。以下命令创建资源节点共置限制。

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

有关资源位置限制的详情，请查看 第 6.3 节 “资源节点共置（Colocation）”。

配置包含多状态资源的排序限制时，可为资源指定的操作之一是 **promote**，表示将该资源从 slave 状态提升至 master。此外，可指定 **demote**，表示将该资源从 master 降级为 slave。

配置排序限制的命令如下。

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

有关资源排序限制的详情，请查看 第 6.2 节 “顺序限制”。

8.2.3. 多状态粘性

要取得稳定的分配模式，默认情况下多状态资源会有些粘性。如果未分配 **resource-stickiness** 值，则多状态资源将使用默认值 1。请使用较小的数值，会对其他资源的 score 计算影响较小，但也足以防止 Pacemaker 在集群中进行不必要的副本移动。

8.3. 监控资源的事件通知

Pacemaker 集群是一个事件驱动系统，其中事件可以是资源失败，或配置更改。**ocf:pacemaker:ClusterMon** 资源可监控集群状态，并触发每个集群事件警报。这个资源以常规间隔在后端运行 **crm_mon**，并使用 **crm_mon** 功能发送电子邮件信息（SMTP）。它还可以使用 **extra_options** 参数执行外部程序。

以下示例配置名为 **ClusterMon-SMTP** 的 **ClusterMon** 资源，该资源将发送电子邮件通知。Pacemaker 事件会导致从 **pacemaker@nodeX.example.com** 使用电子邮件主机 **mail.example.com** 向 **pacemaker@example.com** 发送电子邮件。这个资源可作为克隆创建，以便在集群的所有节点中运行。

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

以下示例配置名为 **ClusterMon-External** 的 **ClusterMon** 资源，运行程序 **/usr/local/bin/example.sh**，以便决定如何就集群通知进行操作。这个资源可作为克隆创建，以便在集群的每个节点中运行。

```
# pcs resource create ClusterMon-External ClusterMon --clone user=root \
update=30 extra_options="-E /usr/local/bin/example.sh -e 192.168.12.1"
```

8.4. pacemaker_remote 服务

pacemaker_remote 服务允许将没有在 **corosync** 中运行的节点整合到该集群，并让该集群如管理真正集群节点一样管理这些资源。就是说 Pacemaker 集群现在可以管理虚拟环境（KVM/LXC）以及处于虚拟环境中的资源，而无需该虚拟环境真的在 **pacemaker** 或 **corosync** 中运行。

使用以下术语描述 **pacemaker_remote** 服务。

- *集群节点* - 运行高可用服务（**pacemaker** 和 **corosync**）的节点。
- *远程节点* — 运行 **pacemaker_remote** 远程整合至集群而无需成为 **corosync** 集群成员。
- *容器* — 包含额外资源的 Pacemaker 资源。例如：包含 webserver 资源的 KVM 虚拟机资源。
- *容器远程节点* — 运行 **pacemaker_remote** 服务的虚拟机远程节点。这论述了具体远程节点使用案例，其中使用集群管理的虚拟机资源可作为远程节点由集群启动，并整合至该集群中。
- *pacemaker_remote* — 可在 Pacemaker 集群环境及独立（非集群）环境中，由虚拟机节点（KVM 和 LXC）执行远程应用程序管理的服务守护进程。这个服务是 Pacemaker 本地资源管理守护进程（LRMD）的高级版本，可远程管理和监控虚拟机 LSB、OCF、upstart 及 systemd 资源。还可允许 **pcs** 在远程节点中工作。
- *LXC* — 由 **libvirt-lxc** Linux 容器驱动程序定义的 Linux 容器。

运行 **pacemaker_remote** 服务的 Pacemaker 集群有以下特征。

- 运行 **pacemaker_remote** 服务的虚拟远程节点（只需要在虚拟机端执行很少的配置）。
- 集群栈（**pacemaker** 和 **corosync**），在集群节点中运行，启动虚拟机并立即连接到 **pacemaker_remote** 服务，允许虚拟机整合至集群中。

虚拟机远程节点和集群节点之间的主要不同是远程节点不在集成栈中运行。这意味着远程节点不计入仲裁。另一方面，这也意味着远程节点没有与集群栈相关的伸缩性限制。除仲裁限制外，就资源管理而言，远程节点行为与集群节点一致。该集群完全有能力管理和监控每个远程节点中的资源。可根据远程节点构建限制，使其处于待命模式，或执行在任何集群节点中执行的操作。在集群状态输出状态中出现的远程节点与集群节点执行同样的操作。

8.4.1. 容器远程节点资源选项

将虚拟机或 LXC 资源配置为远程节点后，可创建管理虚拟机的 **VirtualDomain** 资源。请使用命令查看可为 **VirtualDomain** 资源设定的选项描述。

```
# pcs resource describe VirtualDomain
```

除 **VirtualDomain** 资源选项外，还可配置元数据选项，以便将该资源作为远程节点启用，并定义连接参数。有关元数据选项的详情，请查看 表 8.3 “将 KVM/LXC 资源配置为远程节点的元数据选项”。

表 8.3. 将 KVM/LXC 资源配置为远程节点的元数据选项

项	默认值	描述
remote-node	<none>	这个资源定义的远程节点名称。这可同时将该资源作为远程节点启用，并定义识别该远程节点的特定名称。如果没有设定其他参数，还会将这个值假设为在端口 3121 进行连接的主机名。*警告*：这个值不能与任何资源或节点 ID 重叠。
remote-port	3121	配置自定义端口以便虚拟机连接到 **pacemaker_remote**。
remote-addr	作为主机名使用的 **remote-node** 值	远程节点名称不是该虚拟机的主机名时要连接到的 IP 地址或主机名

项	默认值	描述
remote-connect-timeout	60s	等待处理虚拟机连接超时前所等待的时间

以下命令创建名为 **vm-guest1** 的 **VirtualDomain** 资源，即可使用 **remote-node** meta 属性运行资源的远程节点。

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

8.4.2. 主机及虚拟机认证

集群节点及远程节点之间的认证和加密是通过使用 PSK 加密/认证的 TLS 在 TCP 端口 3121 进行。这就是说集群节点和远程节点必须共享同一私钥。默认情况下该密钥位于集群节点和远程节点的 **/etc/pacemaker/authkey** 中。

8.4.3. 更改默认 pacemaker_remote 选项

如果需要更改 Pacemaker 或 **pacemaker_remote** 的默认端口或 **authkey** 位置，可设定一个环境变量，用来影响这两个守护进程。可通过将其放在 **/etc/sysconfig/pacemaker** 文件中（如下所示）启用这些环境变量。

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

8.4.4. 配置概述：KVM 远程节点

本小节提供了用来在虚拟机中启动 Pacemaker 的步骤，以及使用 **libvirt** 和 KVM 虚拟机将该机器作为远程节点整合的高级概述。

1. 安装虚拟化软件，并在集群节点中启用 **libvirtd** 后，请在每个集群节点和虚拟机的 **/etc/pacemaker/authkey** 中放一个 **authkey**。这样就可保证远程通讯和认证。

   使用以下命令创建 **authkey**。

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

2. 在每台虚拟机中安装 **pacemaker_remote** 软件包，启动 **pacemaker_remote** 服务，并将其设定为启动时运行，并在防火墙中为 TCP 端口 3121 开一个通道。

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

3. 为每台虚拟机分配一个静态网络地址和独特的主机名。

4. 要创建 **VirtualDoman** 资源代理以便管理虚拟机，Pacemaker 要求将该虚拟机的 xml 配置文件转储为一个磁盘中的一个文件。例如：如果创建名为 **guest1** 的虚拟机，使用以下命令将 xml 转储至主机的一个文件中。

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

5. 创建 **VirtualDoman** 资源，配置 **VirtualDoman** 资源元数据选项，以便说明该虚拟机是可运行资源的远程节点。

   在下面的示例中，元数据属性 **remote-node=guest1** 告知 pacemaker 这个资源是使用主机名 **guest1** 的远程节点，可整合至集群中。该集群会在主机名 **guest1** 的虚拟机启动后，尝试连接器其 **pacemaker_remote** 服务。

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

6. 创建 **VirtualDoman** 资源后，可象对待集群中的其他节点一样对待该远程节点。例如：可创建资源并为在该远程节点中运行的资源生成资源限制。

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

   将该远程节点整合至集群中后，可在远程节点中执行 **pcs** 命令，就好象该远程节点正在运行 Pacemaker。

第 9 章 Pacemaker 规则

可使用规则让您的配置更动态。常用示例是设定一个工作时间内使用的 **resource-stickiness** 值，防止将资源移回其最喜欢的位置，同时设定另一个用于周末（没有人会注意到中断）的值。

规则的另一个用途是将机器根据时间分配给不同的进程组（使用节点属性），然后在创建位置限制时使用那个属性。

每个规则可包含大量表达式、日期表达式、甚至其他规则。根据规则 **boolean-op** 字段合并表达式结果，以决定该规则最终评估结果是 **true** 还是 **false**。接下来会出现的结果要看使用该规则的上下文。

表 9.1. 规则属性

项	描述
role	将规则限制为仅作该资源处于那个角色时应用。允许值为：**Started**、**Slave**、和 **Master**。注：使用 **role="Master"** 的规则决定快乐实例的起始位置，只影响要推广的活跃实例。
score	将该规则评估为 **true** 时应用的 score。规则中使用的限制是位置限制的一部分。
score-attribute	将该规则评估为 **true** 时要查找和使用的节点属性。规则中的限制是位置限制的一部分。
boolean-op	如何合并多个表达式对象的结果。允许值为：**and** 和 **or**. The default value is **和**。

9.1. 节点属性表达式

节点属性表达式是用来控制根据一个节点或多个节点所定义属性的资源。

表 9.2. 表达式属性

项	描述
value	用户提供的用来进行比较的值
attribute	要测试的节点属性
type	决定如何测试 value。允许值为：**string**、**integer**、**version**
operation	要执行的比较。允许值为： * **lt** - 如果节点属性值小于 **value** 则为 true。 * **gt** - 如果节点属性值大于 **value** 则为 true。 * **lte** - 如果节点属性值小于等于 **value** 则为 ture。 * **gte** - 如果节点属性值大于等于 **value** 则为 ture。 * **eq** - 如果节点属性值等于 **value** 则为 true。 * **ne** - 如果节点属性值等于 **value** 则为 true。 * **defined** - 如果该节点有命名的属性则为 true。 * **not_defined** - 如果该节点没有命名的属性则为 true

9.2. 基于时间/日期的表达式

使用日期表达式控制基于当前日期/时间的资源或集群选项，可包含自选日期规格。

表 9.3. 日期表达式属性

项	描述
start	遵循 ISO8601 规格的日期/时间。
end	遵循 ISO8601 规格的日期/时间。
operation	将当前日期/时间与 start 和/或 end 日期进行对比，具体要看使用的上下文。允许值为： * **gt** - 如果当前日期/时间晚于 **start** 则为 true * **lt** - 如果当前日期/时间早于 **end** 则为 true。 * **in-range** - 如果当前日期/时间晚于 **start** 并早于 **end** 则为 ture。 * **date-spec** - 执行与当前日期/时间类似 cron 的比较

9.3. 日期规格

使用日期规格创建与时间有关的类似 cron 的表达式。每个字段可包含一个数字或范围。除默认为 0 外，会忽略所有未提供的字段。

例如：**monthdays="1"** 代表每个月的第一天，**hours="09-17"** 代表 9am 到 5pm（包括 9am 和 5pm）。但不能指定为 **weekdays="1,2"** 或 **weekdays="1-2,5-6"**，因为它们包含多个范围。

表 9.4. 日期规格属性

项	描述
id	日期的特定名称
hours	允许值：0-23
monthdays	允许值：0-31（要看具体的月份和年份）
weekdays	允许值：1-7（1=周一, 7=周日）
yeardays	允许值：1-366（根据具体年份）
months	允许值：1-12
weeks	允许值：1-53（根据具体的年份）
years	公历年份
weekyears	各个公历年份可能有所不同，例如：**2005-001 Ordinal** 还可以是 **2005-01-01 Gregorian**，或者 **2004-W53-6 Weekly**
moon	允许值为：0-7（0 代表新月，4 代表满月）。

9.4. 持续时间

可使用持续时间在没有向 in_range 操作提供 end 值时用来计算 **end** 值。它们可包含与 **date_spec** 对象相同的字段，但没有限制（例如：可将持续时间设定为 19 个月）。与 **date_specs** 一样，忽略所有未提供数值的字段。

9.5. 使用 pcs 配置规则

请使用以下命令配置规则。如果省略 **score**，则会默认为 INFINITY。如果省略 **id**，则会使用 *constraint_id* 生成一个。*rule_type* 应为 **expression** 或 **date_expression**。

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

请使用以下命令删除规则。如果要删除的规则是其限制中的最后一个规则，则会删除该限制。

```
pcs constraint rule remove rule_id
```

9.6. 基于时间的表达式示例

如果现在是 2005 年的任何一天，则以下命令会配置一个值为 true 的表达式。

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

以下命令配置一个在周一到周五， 9pm 到 5am 间值为 true 的表达式。注：小时值 16 最大可匹配为 16:59:59，因为（小时）数仍匹配。

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

以下命令配置一个 13 号，周五，满月时值为 true 的表达式。

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

9.7. 使用规则决定资源位置

可使用以下命令根据规则决定资源的位置。

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

expression 可为以下值之一：

- **defined|not_defined *attribute***
- ***attribute* lt|gt|lte|gte|eq|ne *value***
- **date [start=*start*] [end=*end*] operation=gt|lt|in-range**
- **date-spec *date_spec_options***

第 10 章 Pacemaker 集群属性

使用集群属性控制在集群操作过程中遇到问题时的集群行为。

- 表 10.1 “集群属性” 论述集群属性选项。
- 第 10.2 节 “设定和删除集群属性” 论述如何设定集群属性。
- 第 10.3 节 “查询集群属性设置” 论述如何列出当前配置的集群属性。

10.1. 集群属性及选项概述

表 10.1 “集群属性” 总结了 Pacemaker 集群属性，显示这些属性的默认值，以及可以为那些属性设定的值。

注意

除本表格中论述的属性外，还有可用于集群软件的其他属性。建议不要更改这些属性的默认值。

表 10.1. 集群属性

选项	默认值	描述
batch-limit	30	允许平行运行的转移引擎（transition engine，TE）数。所谓“正确”值要具体看网络和集群节点速度和负载。
migration-limit	-1（无限）	允许在某个节点中平行运行 TE 的迁移任务数。
no-quorum-policy	stop	集群没有仲裁时该做什么？允许值为： * ignore - 继续进行所有资源的管理 * freeze - 继续资源管理，但不要恢复不在受影响分区中的资源 * stop - 停止受影响集群分区中的所有资源 * suicide - 隔离受影响集群分区中的所有资源
symmetric-cluster	true	表示资源是否默认可在任意节点中运行。
stonith-enabled	true	表示应隔离失败的节点及包含无法停止资源的节点。要保护数据则需要将这个选项设定为 **true**。 如果设定为 **true**，或未设定，则在配置更多 STONITH 前，该集群会拒绝启动资源。
stonith-action	reboot	发送到 STONITH 设备的动作。允许值为：**reboot**、**off**。还允许使用 **poweroff**，但只能用于旧有设备。
cluster-delay	60s	网络轮询延迟（动作执行除外）。所谓“正确”值要具体看网络和集群节点速度和负载。
stop-orphan-resources	true	表示是否应该停止删除的资源。
stop-orphan-actions	true	表示是否应该取消删除的动作。
start-failure-is-fatal	true	将其设定为 **false** 时，该集群会使用资源的 **resource-failure-stickiness** 值，而不是 **failcount**。
pe-error-series-max	-1 (all)	导致要保存 ERROR 的 PE 输入数。报告问题时使用。
pe-warn-series-max	-1 (all)	导致要保存 WARNING 的 PE 输入数。报告问题时使用。

选项	默认值	描述
pe-input-series-max	-1 (all)	要保存的“正常” PE 输入数。报告问题时使用。
cluster-infrastructure		目前正在运行的 Pacemaker 中堆积的信息。可用于信息及诊断目的，但用户无法进行配置。
dc-version		集群指定控制器（Designated Controller，DC）中的 Pacemaker 版本。可用于诊断目的，但用户无法进行配置。
last-lrm-refresh		本地资源管理器的最后一次刷新，新世纪后采用秒为单位。可用于诊断目的，但用户无法进行配置。
cluster-recheck-interval	15min	对基于时间的选项变更进行轮询的时间间隔。允许值为：0 代表禁用轮询；正数值代表间隔秒数（除非指定其他 SI 单位，比如 5 分钟）。
default-action-timeout	20s	Pacemaker 动作的超时值。这个用于资源自身操作的设定总是优先于集群选项的默认设定值。
maintenance-mode	false	维护模式让集群进入“无操作”模式，且在未告知其启动或停止服务前不会进行任何此类操作。完成维护模式后，集群会检查所有服务的当前状态，然后根据需要停止或启动任意服务。
shutdown-escalation	20min	停止尝试正常关机并退出前要等待的时间。仅作为高级配置使用。
stonith-timeout	60s	等待 SONITH 动作完成的时间。
stop-all-resources	false	集群是否应停止所有资源。
default-resource-stickiness	5000	代表有多少资源首选留在其所在位置。建议将这个值作为资源/操作默认值，而不是集群选项进行设定。
is-managed-default	true	代表是否允许集群启动或停止某个资源。建议将这个值作为资源/操作默认值，而不是集群选项进行设定。
enable-acl	false	（Red Hat Enterprise Linux 7.1 及之后的版本）代表集群是否可使用 **pcs acl** 设定的访问控制列表。

10.2. 设定和删除集群属性

请使用以下 **pcs** 命令设定集群属性。

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

例如：使用以下命令将 **symmetric-cluster** 值设定为 **false**。

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

可使用以下命令从配置中删除集群属性。

```
pcs property unset property
```

另外，也可以通过将 **pcs property set** 命令值字段留为空白，从配置中删除集群属性。这样可将该属性恢复成其默认值。例如：如果之前将 **symmetric-cluster** 属性设定为 **false**，以下命令可将已设定的值从配置中删除，并将 **symmetric-cluster** 恢复成默认值 **true**。

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

10.3. 查询集群属性设置

在大多数情况下，使用 **pcs** 命令显示各种集群组件值时，可交互使用 **pcs list** 或 **pcs show**。在下面的示例中，采用 **pcs list** 格式显示一个以上属性的所有设定列表，同时使用 **pcs show** 格式显示具体属性值。

请使用以下 **pcs** 命令显示已为集群设定的属性值。

```
pcs property list
```

请使用以下命令为集群显示所有属性设定值，包括未明确设定的具体属性设定值。

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

请使用以下命令显示具体集群属性的当前值。

```
pcs property show property
```

例如：要显示 **cluster-infrastructure** 属性的当前值，请执行以下命令：

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

可使用以下命令显示属性的所有默认值列表（无论是否为其设定默认值以外的其他值），以便显示详细信息。

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

第 11 章 pcsd 网页用户界面

本章提供了使用 **pcsd** 网页用户界面配置 Red Hat High Availability 集群概述。

11.1. pcsd 网页用户界面设置

请按照以下步骤将您的系统设定为使用 **pcsd** 网页用户界面配置集群。

1. 如 第 1.2 节 “安装 Pacemaker 配置工具” 所述安装 Pacemaker 配置工具。
2. 在组成集群的每个节点中使用 **passwd** 命令为用户 **hacluster** 设定密码，并在每个节点中都使用相同的密码。
3. 在每个节点中启动并启用 **pcsd** 守护进程：

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

4. 在每个节点中使用以下命令认证组成该集群的节点。运行这个命令后会为您提示输入 **Username** 和 **Password**。将 **Username** 指定为 **hacluster**。

```
# pcs cluster auth node1 node2 ... nodeN
```

5. 在任意系统中打开浏览器进入以下 URL，指定认证的节点之一（注：此操作使用 **https** 协议）。此时会为您显示 **pcsd** 网页用户界面登录页面。

```
https://nodename:2224
```

6. 作为用户 **hacluster** 登录。此时会显示 **管理集群** 页面。

11.2. 使用 pcsd 网页用户界面管理集群

可在 **管理集群** 页面中新建集群，或者网页用户界面中添加现有集群。创建集群后，会在此页面中显示该集群名称。将光标移动到该集群名称上即可显示该集群信息。

要创建集群，请点击 **新建** 按钮，并输入要创建的集群名称以及组成该集群的节点。输入此信息后，点击 **创建集群**。此时会显示在 **管理集群** 页面中刚刚创建的集群及该集群中的节点。

要在网页用户界面中添加现有集群，请点击 **添加现有集群** 按钮，并输入要使用网页用户界面管理的集群节点的主机名和 IP 地址。

点击集群名称即可管理集群。点击后会出现一个页面，您可在该页面中配置该集群的节点、资源、fence 设备和集群属性。

11.3. 集群节点

在集群管理页面的顶部菜单中选择 **节点**，显示当前配置的节点及当前所选节点状态。可在此页面中添加或删除节点，同时还可以启动、停止、重启节点，或让节点处于待机状态。有关待机模式的详情，请查看 第 3.2.4 节 “待机模式”。

还可以选择 **配置 Fencing**，直接使用这个页面配置 fence 设备，如 第 11.4 节 “Fence 设备” 所述。

11.4. Fence 设备

11.4. Fence 设备

在集群管理页面的顶部菜单中选择 **Fence 设备**，显示 **Fence 设备** 页面，该页面中会显示当前配置的 fence 设备。

要在集群中添加新 fence 设备，请点击 **添加**。此时会显示 **添加 Fence 设备** 页面。从**类型** 下拉菜单中选择 fence 设备类型时，必须为该菜单中出现的 fence 设备指定参数。可点击 **自选参数** 选项显示可为 fence 设备定义的附加参数。为新的 fence 设备输入参数后，点击 **创建 Fence 实例**。

有关使用 Pacemaker 配置 fence 设备的详情，请查看 第 4 章 *Fencing：配置 STONITH*。

11.5. 集群资源

在集群管理页面的顶部菜单中选择 **资源**，此时会显示目前为集群配置的资源以及目前所选资源的配置参数。如果某个资源属于某个资源组，则会在该资源名称旁使用括号显示资源组名称。

可添加或删除资源，还可以编辑现有资源配置。

要在集群中添加新资源，请点击 **添加**。此时会显示 **添加资源** 页面。从 **类型** 下拉菜单中选择资源类型时，必须为菜单中出现的资源指定参数。点击 **自选参数** 显示要定义资源的附加参数。输入要创建资源的参数后，点击 **创建资源**。

为某个资源配置参数时，会在该菜单中出现这个参数的概述。将光标移动到该字段后，会为该参数显示更详细的帮助信息。

可将资源定义为克隆的资源，或定义为主/辅资源。有关资源类型的详情，请查看 第 8 章 *高级资源类型*。

创建至少一个资源后，可为该资源创建资源组。有关资源组的详情，请查看 第 5.5 节 “资源组”。

要创建资源组，请在 **资源** 页面中选择属于该组的资源，然后点击 **创建资源组**。此时会显示 **创建资源组** 页面。输入组名称，并点击 **创建资源组**。此时会带您返回 **资源** 页面，该页面中会显示该资源所在组名称。创建资源组后，可在创建或修改附加资源时，将该组名称作为资源参数使用。

11.6. 集群属性

在集群管理页面的顶部菜单中选择 **集群属性**，此时会显示集群属性，并允许您修改这些属性的默认值。有关 Pacemaker 集群属性的详情，请查看 第 10 章 *Pacemaker 集群属性*。

附录 A. 在 Red Hat Enterprise Linux Release 6 和 Red Hat Enterprise Linux Release 7 中创建集群

在 Red Hat Enterprise Linux 7 中使用 Pacermaker 配置 Red Hat High Availability Cluster 时要求使用的配置工具管理界面，与在 Red Hat Enterprise Linux 6 使用 **rgmanager** 配置集群时的配置工具管理界面不同。第 A.1 节 “使用 rgmanager 和 Pacemaker 创建集群” 总结了不同集群组件间的不同配置。

Red Hat Enterprise Linux 6.5 发行本支持使用 **pcs** 配置工具通过 Pacemaker 配置集群。第 A.2 节 “在 Red Hat Enterprise Linux Release 6.5 和 Red Hat Enterprise Linux Release 7 中使用 Pacemaker 创建集群” 总结了一些 Red Hat Enterprise Linux release 6.5 中的 **pcs** 支持与 Red Hat Enterprise Linux release 7.0 中 **pcs** 支持的不同之处。

A.1. 使用 rgmanager 和 Pacemaker 创建集群

表 A.1 “使用 rgmanager 及 Pacemaker 配置集群对比” 中提供了在 Red Hat Enterprise Linux 6 中使用 **rgmanager** 以及在 Red Hat Enterprise Linux 7 中使用 Pacemaker 配置集群组件的对比概述。

表 A.1. 使用 rgmanager 及 Pacemaker 配置集群对比

配置组件	rgmanager	Pacemaker
集群配置文件	每个节点中的集群配置文件为 **cluster.conf** 文件，可在需要时直接编辑。也可以使用 **luci** 或 **ccs** 界面定义集群配置。	集群和 Pacemaker 配置文件为 **corosync.conf** 和 **cib.xml**。请不要直接编辑这些文件，而是使用 **pcs** 或 **pcsd** 界面编辑这些文件。
网络设置	配置集群前请先配置 IP 地址和 SSH。	配置集群前请先配置 IP 地址和 SSH。
集群配置工具	使用 **luci** 和 **ccs** 命令可手动编辑 **cluster.conf** 文件。	**pcs** 或者 **pcsd**。
安装	安装 **rgmanager**（该程序可提取所有相依性，包括 **ricci**、**luci** 及资源和 fencing 代理）。必要时请安装 **lvm2-cluster** 和 **gfs2-utils**。	安装所需 **pcs** 和 fencing 代理。必要时请安装 **lvm2-cluster** 和 **gfs2-utils**。
启动集群服务	按照以下步骤启动并启用集群服务： 1. 启动 **rgmanager** 和 **cman**，必要时启动 **clvmd** 和 **gfs2**。 2. 启动 **ricci**，并使用 **luci** 界面启动 **luci**。 3. 为所需服务运行 **chkconfig on** 以便每次在引导时启用它们。 也可以运行 **ccs --start** 启动并启用这些集群服务。	按照以下步骤启动并启用集群服务： 1. 在每个节点中执行 **systemctl start pcsd.service**，然后执行 **systemctl enable pcsd.service**，以便在运行时启动 **pcsd**。 2. 在集群的一个节点中运行 **pcs cluster start --all**，启动 **corosync** 和 **pacemaker**。
控制对配置工具的访问	对于 **luci**，只有 root 用户或拥有 **luci** 权限的普通用户可访问 **luci**。所有访问都需要该节点的 **ricci** 密码。	**pcsd** 图形用户界面要求您作为用户 **hacluster**（即通用系统用户）进行认证。root 用户可为 **hacluster** 设定密码。

配置组件	rgmanager	Pacemaker
创建集群	使用 **luci** 或 **ccs** 为集群命名，并定义所包含的节点，也可以直接编辑 **cluster.conf** 文件。	请使用 **pcs cluster setup** 命令或 **pcsd** 网页用户界面为集群命名，并在其中包含节点。还可以使用 **pcs cluster node add** 命令或 **pcsd** 网页用户界面在现有集群中添加节点。
将集群配置填充至所有节点	使用 **luci** 配置集群时会自动填充。若使用 **ccs** 命令，则需附加 **--sync** 选项。也可以使用 **cman_tool version -r** 命令填充。	添加节点或资源时，可自动填充集群和 Pacemaker 配置文件 **corosync.conf** 和 **cib.xml**。
全局集群属性	Red Hat Enterprise Linux 6 的 **rgmanager** 支持以下功能： * 可以配置系统，以便其选择使用哪个多播地址在集群网络中进行 IP 多播。 * 若 IP 多播不可用，则可以使用 UDP 单播传输机制。 * 可将集群配置为使用 RRP 协议。	RHEL 7 中的 Pacemaker 支持以下集群功能： * 可为集群设定 **no-quorum-policy**，以便集群指定在该集群没有仲裁时系统可以做什么。 * 有关其他可设定的集群属性，请参考 表 10.1 “集群属性”。
日志	可设定全局或具体守护进程的日志配置。	有关如何手动配置日志的详情，请查看文件 **/etc/sysconfig/pacemaker**。
验证集群	使用集群方案时，可自动使用 **luci** 和 **ccs** 进行集群确认。该集群在启动时自动确认。	该集群启动时自动验证，也可以使用 **pcs cluster verify** 命令验证。
双节点集群中的仲裁	在双节点集群中，可配置系统决定仲裁的方式： * 配置仲裁磁盘 * 使用 **ccs** 或编辑 **cluster.conf** 文件设定 **two_node=1** 和 **expected_votes=1**，以便允许单一节点维护仲裁。	**pcs** 在 **corosync** 中自动添加双节点集群所需选项。
集群状态	在 **luci** 中，可通过刷新界面的不同组件查看集群当前状态。可使用 **ccs** 命令的 **--gethost** 选项查看当前配置文件。可使用 **clustat** 命令显示集群状态。	可使用 **pcs status** 命令显示当前集群状态。
资源	请使用 **luci** 或 **ccs** 命令，或编辑 **cluster.conf** 配置文件添加定义的资源，并配置具体资源的属性。	请使用 **pcs resource create** 命令或 **pcsd** 网页界面添加定义类型的资源，并配置具体资源属性。有关使用 Pacemaker 配置集群资源的常规信息，请参考 第 5 章 *配置集群资源*。

配置组件	rgmanager	Pacemaker
资源行为、分组及启动/停止顺序	定义集群*服务*以配置资源互动方式。	在 Pacemaker 中，可使用资源组作为定义一组需处于相同位置，并按顺序启动和停止的资源的快捷方式。另外，也可根据以下方式定义资源行为及互动： * 可将资源行为的某些方面作为资源选项设定。 * 使用位置限制决定资源能够在哪些节点中运行。 * 使用顺序限制决定资源的运行顺序。 * 使用节点共置限制决定某个资源位置依赖另一资源所在位置。 有关这些专题的详情请查看 第 5 章 *配置集群资源* 和 第 6 章 *资源限制*.
资源管理：移动、启动和停止资源	可使用 **luci** 管理集群、独立集群节点以及集群服务。可使用 **ccs** 命令管理集群。可使用 **clusvadm** 管理集群服务。	可临时禁用某个节点，这样就无法使用 **pcs cluster standby** 命令托管资源，从而导致资源迁移。可使用 **pcs resource disable** 命令停止资源。
完全删除集群配置	可使用 **luci** 选择要从集群中删除的所有节点，以便完全删除集群。还可以从集群的每个节点中删除 **cluster.conf**。	可使用 **pcs cluster destroy** 命令删除集群配置。
在多个节点中活跃的资源，在多种模式的多节点中活跃的资源。	不对等。	可使用 Pacemaker 克隆那些可在多个节点中运行的资源，并将克隆的资源定义为主资源和辅资源，以便其在多个节点中运行。有关克隆的资源以及主/辅资源的详情，请查看 第 8 章 *高级资源类型*。
Fencing -- 每个节点中的单一 fence 设备	全局或本地创建 fencing 设备，并将其添加到节点中。可为集群作为整体定义 **post-fail delay** 和 **post-join delay** 值。	使用 **pcs stonith create** 命令或 **pcsd** 网页界面为每个节点创建 fencing 设备。对于可隔离多个节点的设备，则只需要定义一次即可，不需要分别在各个节点中定义。还可以定义 **pcmk_host_map**，使用单一命令为所有节点配置 fencing 设备。有关 **pcmk_host_map** 的详情请查看 表 4.1 “Fencing 设备的常规属性”。可为集群作为整体定义 **stonith-timeout** 值。
每个节点中的多（备用）fencing 设备	使用 **luci** 或 **ccs** 命令，或直接编辑 **cluster.conf** 文件定义备用设备。	配置 fencing 等级

A.2. 在 Red Hat Enterprise Linux Release 6.5 和 Red Hat Enterprise Linux Release 7 中使用 Pacemaker 创建集群

Red Hat Enterprise Linux 6.5 发行本支持使用 **pcs** 配置工具，通过 Pacemaker 进行集群配置。但在 Red Hat Enterprise Linux 6.5 和 Red Hat Enterprise Linux 7 中使用 Pacemaker 进行集群安装和创建时会稍有不同。本小节提供这两个发行本中命令的区别概述。有关在 Red Hat Enterprise Linux 7 中安装和创建集群的详情，请查看 第 1 章 *Red Hat High Availability Add-On 配置及管理参考概述* 和 第 3 章 *集群创建及管理*。

A.2.1. 在 Red Hat Enterprise Linux 6.5 和 Red Hat Enterprise Linux 7 中的 Pacemaker 安装

以下命令可安装 Pacemaker 要求在 Red Hat Enterprise Linux 6.5 中安装的 Red Hat High Availability Add-On 软件包，并防止 **corosync** 在没有安装 **cman** 的情况下启动。必须在集群的每个节点中运行这些命令。

```
[root@rhel6]# yum install pacemaker cman
[root@rhel6]# yum install pcs
[root@rhel6]# chkconfig corosync off
```

在 Red Hat Enterprise Linux 7 中，除安装 Pacemaker 需要的 Red Hat High Availability Add-On 软件包外，还要为名为 **hacluster** 的 **pcs** 管理帐户设定密码，同时要启动并启用 **pcsd** 服务。还需要为集群的这些节点认证该管理帐户。

在 Red Hat Enterprise Linux 7 中，请在集群的每个节点中运行以下命令。

```
[root@rhel7]# yum install pcs fence-agents-all
[root@rhel7]# passwd hacluster
[root@rhel7]# systemctl start pcsd.service
[root@rhel7]# systemctl enable pcsd.service
```

在 Red Hat Enterprise Linux 7 中，在集群的一个节点中运行以下命令。

```
[root@rhel7]# pcs cluster auth [node] [...] [-u username] [-p password]
```

A.2.2. 在 Red Hat Enterprise Linux Release 6.5 和 Red Hat Enterprise Linux Release 7 中使用 Pacemaker 创建集群

要在 Red Hat Enterprise Linux Release 6.5 中创建 Pacemaker 集群，则必须首先创建该集群，并在集群的每个节点中启动集群服务。例如：在 **z1.example.com** 和 **z2.example.com** 中运行以下命令，创建名为 **my_cluster**，由节点 **z1.example.com** 和 **z2.example.com** 组成的集群，并运行以下命令在那些节点中启动集群服务。

```
[root@rhel6]# pcs cluster setup --name my_cluster z1.example.com z2.example.com
[root@rhel6]# pcs cluster start
```

在 Red Hat Enterprise Linux Release 7 中，需在集群的一个节点中运行创建集群命令。只在一个节点中运行以下命令，创建名为 **my_cluster**，由节点 **z1.example.com** 和 **z2.example.com** 组成的集群，并在那些节点中启动集群服务。

```
[root@rhel7]# pcs cluster setup --start --name my_cluster z1.example.com
z2.example.com
```

附录 B. 修订历史

修订 1.1-9.2 **Sat Aug 1 2015** **Leah Liu**
完成翻译、校对

修订 1.1-9.1 **Sat Aug 1 2015** **Leah Liu**
与 XML 源 1.1-9 版本同步的翻译文件

修订 1.1-9 **Mon Feb 23 2015** **Steven Levine**
7.1 GA 发行本

修订 1.1-7 **Thu Dec 11 2014** **Steven Levine**
7.1 Beta 发行本

修订 1.1-2 **Tue Dec 2 2014** **Steven Levine**
解决 #1129854
更新 pcs 资源清除命令文档。

解决 #1129837
编写 pcs rescource move 命令的 lifetime 参数文档。

解决 #1129862, #1129461
编写访问控制列表支持文档。

解决 #1069385
添加为 DLM 启用的端口号。

解决 #1121128, #1129738
更新并澄清设置节点共置和排序组的说明。

解决 #1129713
编写集群仲裁 unblock 功能文档。

解决 #1129722
更新 auto_tie_breaker 仲裁选项描述。

解决 #1129737
编写为 pcs resource create 命令禁用参数的文档。

解决 #1129859
编写 pcs 集群配置备份和恢复文档。

解决 #1098289
说明端口要求描述。

修订 1.1-1 **Tue Nov 18 2014** **Steven Levine**
起草 7.1 发行本更新

修订 0.1-41 **Mon Jun 2 2014** **Steven Levine**
7.0 GA 发行版本

修订 0.1-39 **Thu May 29 2014** **Steven Levine**

解决：#794494
编写仲裁支持文档

解决: #1088465
编写 unfencing 文档

解决: #987087
编写 pcs 更新文档

修订 0.1-23	Wed Apr 9 2014	Steven Levine

更新 7.0 Beta 版草稿

修订 0.1-15	Fri Dec 6 2013	Steven Levine

Beta 文档

修订 0.1-2	Thu May 16 2013	Steven Levine

首次打印第一稿

索引

符号

设定属性，设定和删除集群属性

资源，资源属性，在集群中手动移动资源

- 与其他资源的相对位置，资源节点共置（Colocation）
- 位置
  - 由规则决定，使用规则决定资源位置
- 克隆，资源克隆
- 启动顺序，顺序限制
- 启用，启用和禁用集群资源
- 多状态，多状态资源：有多个模式的资源
- 属性
  - id，资源属性
  - 提供程序，资源属性
  - 标准，资源属性
- 清除，集群资源清除
- 禁用，启用和禁用集群资源
- 移动，在集群中手动移动资源
- 组，资源组
- 资源
  - 类型，资源属性
- 选项
  - failure-timeout，资源元数据选项
  - is-managed，资源元数据选项
  - migration-threshold，资源元数据选项
  - multiple-active，资源元数据选项
  - requires，资源元数据选项
  - resource-stickiness，资源元数据选项
  - target-role，资源元数据选项
  - 优先权，资源元数据选项
- 限制
  - Attribute Expression，节点属性表达式
  - Date Specification，日期规格
  - Duration，持续时间
  - 共置，资源节点共置（Colocation）
  - 日期/时间表达式，基于时间/日期的表达式
  - 规则，Pacemaker 规则
  - 顺序，顺序限制

资源选项，资源元数据选项

选项

- clone-max，创建及删除克隆的资源
- clone-node-max，创建及删除克隆的资源
- cluster-delay，集群属性及选项概述
- cluster-infrastructure，集群属性及选项概述
- cluster-recheck-interval，集群属性及选项概述
- dampen，由于连接更改而移动资源
- dc-version，集群属性及选项概述
- default-action-timeout，集群属性及选项概述
- default-resource-stickiness，集群属性及选项概述
- enable-acl，集群属性及选项概述
- failure-timeout，资源元数据选项
- globally-unique，创建及删除克隆的资源
- host_list，由于连接更改而移动资源

- start，基于时间/日期的表达式

- 日期规格，日期规格
  - hours，日期规格
  - id，日期规格
  - monthdays，日期规格
  - months，日期规格
  - moon，日期规格
  - weekdays，日期规格
  - weeks，日期规格
  - weekyears，日期规格
  - yeardays，日期规格
  - years，日期规格

- 规则，Pacemaker 规则
  - boolean-op，Pacemaker 规则
  - role，Pacemaker 规则
  - score-attribute，Pacemaker 规则

- 顺序，顺序限制
  - kind，顺序限制

限制表达式，节点属性表达式，基于时间/日期的表达式

限制规则，Pacemaker 规则

集群

- 删除属性，设定和删除集群属性
- 属性
  - batch-limit，集群属性及选项概述

- 查询属性，查询集群属性设置
- 设定属性，设定和删除集群属性
- 选项
  - cluster-delay，集群属性及选项概述
  - cluster-infrastructure，集群属性及选项概述
  - cluster-recheck-interval，集群属性及选项概述
  - dc-version，集群属性及选项概述
  - default-action-timeout，集群属性及选项概述
  - default-resource-stickiness，集群属性及选项概述
  - enable-acl，集群属性及选项概述
  - is-managed-default，集群属性及选项概述
  - last-lrm-refresh，集群属性及选项概述
  - maintenance-mode，集群属性及选项概述
  - migration-limit，集群属性及选项概述
  - no-quorum-policy，集群属性及选项概述
  - pe-error-series-max，集群属性及选项概述
  - pe-input-series-max，集群属性及选项概述
  - pe-warn-series-max，集群属性及选项概述
  - shutdown-escalation，集群属性及选项概述
  - start-failure-is-fatal，集群属性及选项概述
  - stonith-action，集群属性及选项概述
  - stonith-enabled，集群属性及选项概述
  - stonith-timeout，集群属性及选项概述
  - stop-all-resources，集群属性及选项概述
  - stop-orphan-actions，集群属性及选项概述
  - stop-orphan-resources，集群属性及选项概述
  - symmetric-cluster，集群属性及选项概述

M

N

O

P