

第 12 章 高可用性

现代社会严重依赖于计算机网络提供的信息，移动终端的普及进一步加重了这种依赖，人们无时无刻都需要能够访问网络。如果你在提供这类服务，那么确保服务始终在线就变得非常重要。

我们可以通过计算服务在线时间（A）和总时间段（B）的比值来定义服务可用性。通常都表示为在一年内的在线时间比率。

表 12.1：可用性，一年内的当机时间

可用性 %	一年内的停机时间
99	3.65 天
99.9	8.76 小时
99.99	52.56 分钟
99.999	5.26 分钟
99.9999	31.5 秒
99.99999	3.15 秒

提高可用性的方法有很多。最具逼格的是重写软件，以便软件能够同时在多个主机并发运行。这要求软件本身具备错误检测和故障转移能力。对于只包含静态页面的 Web 网站来说，这种方法还能凑合用用。但更多情况下，这种方式都非常复杂，经常因为你无法修改软件而完全没有可行性。以下是一些不修改软件的提高可用性的办法：

- 使用可靠的服务器硬件

➤ 注意

由于质量的不同，相同功能的计算机硬件往往具有不同的可用性指标。大部分厂商将可靠性较高的硬件作为“服务器级”产品出售，当然其价格也更高。

- 消除单点故障（冗余硬件）
 - 使用不间断电源（UPS）

- 为主板配备多路电源
- 使用 ECC 内存
- 使用多路网卡
- 使用 RAID 技术管理本地存储
- 使用分布式多副本存储技术保存虚拟机镜像
- 减少停机时间
 - 可快速访问的管理界面（24/7）
 - 可用的空闲节点（Proxmox VE 集群中的其他节点）
 - 自动化故障检测（ha-manager 提供）
 - 自动化故障转移（ha-manager 提供）

由于彻底消除了对硬件的依赖，Proxmox VE 这样的虚拟化技术能够轻松实现服务的高可用性。在配置了冗余存储和网络资源的情况下，遭遇个别服务器节点故障时，可以很容易在集群中其他服务器节点恢复服务运行。

Proxmox VE 进一步提供了 ha-manager 组件，能够自动完成包括故障检测和故障转移在内的一切高可用管理任务。

Proxmox VE 的 ha-manager 组件就像一个“全自动”的管理员。你只需将资源（虚拟机，容器等）配置交给它管理，ha-manager 就会连续监测服务运行状态，并在发生故障时将服务转移到其他节点运行。当然，ha-manager 也可以处理日常的管理操作请求，例如开机、停止、重新部署和迁移虚拟机。

但高可用性不是免费的午餐。实现高可用性需要投入更多资源，预备空闲节点等都会增加成本，因此你应该认真计算评估高可用性的收益和所需成本。

➤ 注意

将可用性从 99% 提高到 99.9% 还是比较容易的。但从 99.9999% 提高到 99.99999% 则难的多也贵的多。ha-manager 的故障检测和故障转移时间大概为 2 分钟，因此能实现的可用性最多不超过 99.999%。

12.1 部署条件

在开始部署 HA 之前，需要满足以下条件：

- 集群最少有 3 个节点（以得到稳定的 quorum）
- 为虚拟机和容器配置共享存储
- 硬件冗余（各个层面）
- 使用可靠的“服务器”硬件
- 硬件看门狗 - 如不具备，也可以退而求其次使用 Linux 内核的软件看门狗（softdog）
- 可选的硬件隔离设备

12.2 资源

我们将 ha-manager 管理的对象称为资源。一个资源（也称为“服务”）由一个唯一的服务 ID 标识（SID）。服务 ID 由资源类型和类型内的 ID 两部分组成，例如 vm:100，是指一个 vm 类型（虚拟机）的资源，而资源 ID 为 100。

目前主要有两类资源，虚拟机和容器。一个资源对应一个虚拟机或容器，资源的所有相关软件需要安装到这个虚拟机或容器中，而不是像 rgmanager 那样把多个资源捆绑成一个大资源。通常来说，HA 管理的资源不应再依赖其他资源。

12.3 管理任务

本节将简单介绍常见管理任务。首先是在资源上激活 HA，也就是把资源添加到 HA 的资源配置中，可以通过 WebGUI 进行，也可以使用命令行工具完成该操作，如下：

```
# ha-manager add vm:100
```

之后，HA 组件将启动该资源并全力确保它连续运行。当然，你也可以配置该资源的“指定”工作状态，例如可以要求 HA 组件停止该资源的运行：

```
# ha-manager set vm:100 --state stopped
```

然后再启动运行

```
# ha-manager set vm:100 --state started
```

你也可以使用常用的虚拟机和容器管理工具来改变资源运行状态，而常用工具会自动调用 HA 组件完成操作指令。因此

```
# qm start 100
```

将资源状态设置为 started。命令 qm stop 的原理类似，只是将资源状态设置为 stopped。

➤ 注意

HA 组件以异步方式工作，并需要和集群其他成员进行通讯，因此从发出指令到观察到操作完成需要一些时间。

可以用如下命令查看 HA 的资源配置情况：

```
# ha-manager config
```

```
vm:100
```

```
state stopped
```

可以用如下命令查看 HA 管理器和资源状态：

```
# ha-manager status
```

```
quorum OK
```

```
master node1 (active, Wed Nov 23 11:07:23 2016)
```

```
lrm elsa (active, Wed Nov 23 11:07:19 2016)
```

```
service vm:100 (node1, started)
```

可以用如下命令将资源迁移到其他节点：

```
# ha-manager migrate vm:100 node2
```

上面的命令采用在线迁移方式，虚拟机在迁移过程中将保持运行。在线迁移需要通过网络将虚拟机内存数据传输到目标节点，因此在某些情况下关闭虚拟机然后在目标节点重新启动可能更快，具体可使用 relocate 命令进行：

```
# ha-manager relocate vm:100 node2
```

最后，可用如下命令将资源从 HA 的资源配置中删除：

```
# ha-manager remove vm:100
```

➤ 注意

该操作并不需要启停虚拟机。

所有的 HA 管理操作都可通过 WebGUI 进行，一般情况下无须使用命令行。

12.4 工作原理

本节将详细描述 HA 管理器的内部工作原理，包括所有服务进程及其协同工作过程。
HA 在每个节点上都有两个服务进程：

pve-ha-lrm

该服务称为本地资源管理器（LRM），其主要任务是控制本地节点的资源运行状态，首先从当前管理器状态文件读取资源的指定工作状态，然后执行相应的操作命令。

pve-ha-crm

该服务称为集群资源管理器（LRM），其主要任务是负责集群节点之间的协同决策工作，具体包括向 LRM 发送命令，处理命令执行结果，在出现故障时将资源转移到其他节点运行，此外还负责故障节点隔离。

➤ 注意

HA 服务利用了集群文件系统提供的锁机制。通过锁机制，确保了每次只有一个 LRM 被激活并处于工作状态。由于 LRM 只在获取锁之后才能执行 HA 任务，我们可以在获取锁之后将故障节点标记为隔离，然后可以在其他节点安全地恢复原来在故障节点运行的 HA 资源，而无须担心故障节点的干扰。整个过程都在拥有 HA 管理器主锁的 CRM 监督下进行。

12.4.1 资源状态

CRM 使用一个枚举变量来记录当前资源的状态。不仅 WebGUI 界面有显示当前资源状态，并且你可以运行 ha-manager 命令行工具获取该状态。

```
# ha-manager status  
quorum OK  
master elsa (active, Mon Nov 21 07:23:29 2016)  
lrm elsa (active, Mon Nov 21 07:23:22 2016)  
service ct:100 (elsa, stopped)  
service ct:102 (elsa, started)  
service vm:501 (elsa, started)
```

以下是可能的状态

stopped

资源已停止（LRM 确认）。如果 LRM 检测到应处于停止状态的资源仍然在运行，它将再次停止该资源。

request_stop

资源应被停止。该状态下，CRM 将等待 LRM 确认资源已停止。

stopping

正在挂起的停止请求。表示 CRM 仍未接到该停止请求。

started

资源处于运行状态，并且 LRM 应该在发现资源未运行时立刻启动该资源。如果资源因故停止运行，LRM 会在检测到后立刻重启它（查看 12.7 节启动失败策略）。

starting

正在挂起的启动请求。表示 CRM 未得到 LRM 对该资源正在运行的确认。

fence

等待节点完成隔离（将节点从集群投票范围内隔离出去）。一旦完成隔离，资源将在其他节点恢复（查看 12.6 节隔离）。

freeze

表示禁止访问资源。该状态用于节点重启过程，或 LRM 重启过程（查看 12.9 节软件包升级）。

migrate

将资源迁移（在线）到其他节点。

error

因 LRM 错误，资源被禁用。该状态往往意味着需要手工干预（查看 12.8 节错误恢复）。

queued

表示资源刚被添加到 HA，而 CRM 尚未确认已看到该资源。

disabled

资源被停止运行，并被标记为 disabled。

12.4.2 本地资源管理器

本地资源管理器（pve-ha-lrm）以系统服务形式启动。启动后，该服务将等待集群进入多数票状态，以确保集群锁机制正常工作。

该服务有 3 种状态：

wait for agent lock

表示 LRM 在等待获取的独占锁。如果未配置任何 HA 资源，该状态就相当于空闲状态。

active

表示配置了 HA 资源，并且 LRM 获得了独占锁。

lost agent lock

表示 LRM 失去了独占锁，一般意味着有错误发生，并且节点失去了多数票。

LRM 进入 active 状态后，将读取配置文件/etc/pve/ha/manager_status，并根据它所管理的资源判断应该执行的管理命令。每条命令都由一个独立工作进程执行，因此可以并发执行多条命令，但默认最多同时并发执行 4 条命令。可以修改数据中心配置项 max_worker 来调整默认并发数。当命令执行完后，工作进程将被回收，执行结果也会被 CRM 记录保存。

➤ 注意

默认的并发数 4 不一定适用于所有环境。例如，同时执行 4 个在线迁移操作可能会导致对网络的竞争使用，特别在物理网络速度较慢或配置了大内存资源时。在任何情况下必须确保避免发生竞争的情况，必要时可以降低 max_worker 的值。相反，如果你的硬件配置极端牛逼，也可以考虑增加 max_worker 的值。

CRM 发出的每条命令都由一个 UID 标识，当工作进程完成命令执行后，执行结果将被写入 LRM 状态文件/etc/pve/nodes/<nodename>/lrm_status，而 CRM 可能会收集该结果并用它自己的状态机进一步处理该结果。

通常，CRM 和 LRM 对每一个资源的操作都是同步进行的。也就是说，CRM 发出一个唯一 UID 标识的命令，LRM 则执行一次该命令并将执行结果写回文件，而执行结果用同一个 UID 标识。这确保了 LRM 不会执行过期的命令。但 stop 命令和 error 命令是两个例外，这两个命令不依赖于处理结果，并总是在 stopped 或 error 状态执行。

➤ 注意

HA 组件会记录每个操作的日志。这有助于理解集群中发生的事以及发生的原因。这对于了解两个服务进程 LRM 和 CRM 干了什么尤为重要。你可以用命令 journalctl -u pve-ha-lrm 查看资源所在节点的本地资源管理器日志，并用同样命令查看当前主节点的 pve-ha-crm 服务日志。

12.4.3 集群资源管理器

集群资源管理器（pve-ha-crm）在每个节点启动后，将进入等待状态直到获取管理器锁。管理器锁每次只能由一个节点获取，而成功获取该锁的节点将被提升为 CRM 主节点。

该服务有 3 种状态：

wait for agent lock

表示 CRM 在等待获取的独占锁。如果未配置任何 HA 资源，该状态就相当于空闲状态。

active

表示配置了 HA 资源，并且 CRM 获得了独占锁。

lost agent lock

表示 CRM 失去了独占锁，一般意味着有错误发生，并且节点失去了多数票。

CRM 的主要任务是管理那些纳入 HA 管理的资源，并尽力确保资源处于指定的状态。例如，对于一个指定状态为 started 的资源，一旦被发现未运行就会立刻被启动，如果资源意外崩溃，也会被自动重启。而 CRM 将负责告诉 LRM 具体进行哪些操作。

一个节点失去集群多数票后，会进入 unknown 状态。此时，如果 CRM 能够安全释放故障节点的锁，相关资源将会被转移到其他节点重新启动。

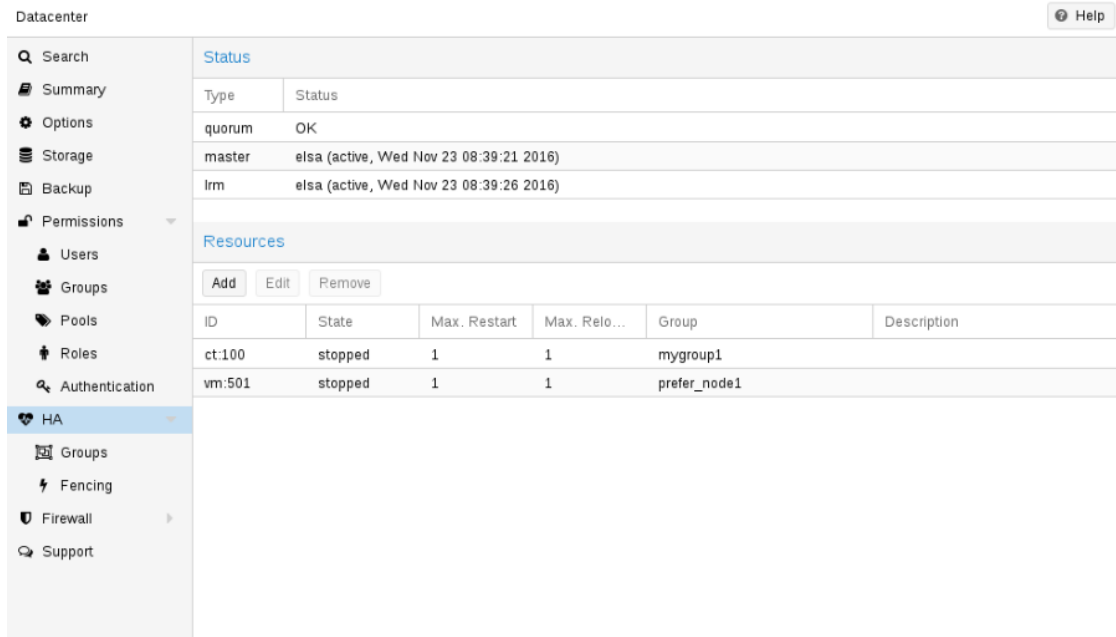
当集群节点判定自己不再拥有集群多数票后，LRM 将等待新的多数票形成。只要没能形成多数票，节点就无法重置看门狗，最终看门狗超时会触发节点重启，看门狗默认超时时间为 60 秒。

12.5 配置步骤

HA 组件被紧密集成到了 Proxmox VE API 中。因此，你既能够通过 ha-manager 命令行配置 HA，也可以通过 WebGUI 配置 HA，两种方式都很简便，更进一步，还可以用自动化工具直接调用 API 配置 HA。

HA 配置文件全部保存在 /etc/pve/ha/ 目录中，因此可以被自动复制到集群所有节点，所有节点都共享使用相同的 HA 配置。

12.5.1 资源



The screenshot shows the HA Manager web interface. On the left is a sidebar menu with options: Search, Summary, Options, Storage, Backup, Permissions, Users, Groups, Pools, Roles, Authentication, HA (selected), Groups, Fencing, Firewall, and Support. The main content area is titled 'Status' and shows a table with columns 'Type' and 'Status'. It lists 'quorum' as 'OK', 'master' as 'elsa (active, Wed Nov 23 08:39:21 2016)', and 'lrm' as 'elsa (active, Wed Nov 23 08:39:26 2016)'. Below this is a 'Resources' section with 'Add', 'Edit', and 'Remove' buttons. A table lists resources with columns: ID, State, Max. Restart, Max. Relo..., Group, and Description. It shows two resources: 'ct:100' (stopped, 1 restart, 1 relocate, mygroup1) and 'vm:501' (stopped, 1 restart, 1 relocate, prefer_node1).

Type	Status
quorum	OK
master	elsa (active, Wed Nov 23 08:39:21 2016)
lrm	elsa (active, Wed Nov 23 08:39:26 2016)

ID	State	Max. Restart	Max. Relo...	Group	Description
ct:100	stopped	1	1	mygroup1	
vm:501	stopped	1	1	prefer_node1	

资源配置文件/etc/pve/ha/resources.cfg 保存了 ha-manager 管理的所有资源列表。资源列表中的资源配置格式如下：

```
<type>: <name>
    <property> <value>
    ...
```

每条资源配置信息都以冒号分隔的资源类型和资源名称开始，这也是 ha-manager 命令用于标识 HA 资源的 ID（例如 vm:100 或 ct:101），而后续配置行包含了附加属性：

comment: <string>

描述信息。

group: <string>

HA 组标识符。

max_relocate: <integer> (0 -N) (default = 1)

资源启动失败后尝试重新部署最大次数。

max_restart: <integer> (0 -N) (default = 1)

资源启动失败后尝试重新启动最大次数。

state: <disabled | enabled | started | stopped> (default = started)

资源的指定状态。CRM 将根据该状态值管理相关资源。请注意 enabled 是 started 的别名。

started

CRM 将尝试启动资源，并在成功启动后将状态设置为 started。如果遭遇节点故障或启动失败，CRM 将尝试恢复资源。如果所有尝试均告失败，状态将被设为 error。

stopped

CRM 将努力确保资源处于停止状态。但在遭遇节点时，CRM 还是会尝试将资源重新部署到其他节点。

disabled

CRM 将努力确保资源处于停止状态。但在遭遇节点时，CRM 不会将资源重新部署到其他节点。设置该状态的主要目的是将资源从 error 状态恢复出来，因为这是 error 状态的资源唯一可以被设置的状态。

以下是一个实际生产中的例子，其中包含了一个虚拟机和一个容器。可以看到，配置文件的语法其实非常简单，所以你可以用文本编辑器直接读取或修改这些配置文件：

配置示例 (/etc/pve/ha/resources.cfg)

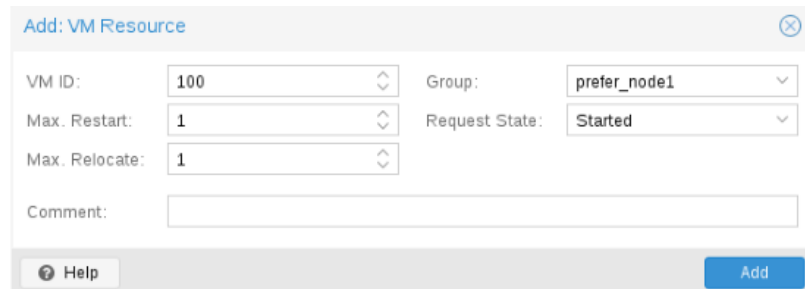
vm: 501

state started

max_relocate 2

ct: 102

Note: use default settings for everything



The screenshot shows a dialog box titled "Add: VM Resource". It contains several input fields and dropdown menus. The "VM ID" field is set to "100". The "Group" dropdown is set to "prefer_node1". The "Max. Restart" field is set to "1". The "Max. Relocate" field is set to "1". The "Request State" dropdown is set to "Started". There is a "Comment" text area at the bottom. At the bottom left is a "Help" button, and at the bottom right is an "Add" button.

以上配置示例是由命令行工具 ha-manager 生成的：

```
# ha-manager add vm:501 --state started --max_relocate 2
```

```
# ha-manager add ct:102
```

12.5.2 组

Datacenter

Help

Search

Summary

Options

Storage

Backup

Permissions

Users

Groups

Pools

Roles

Authentication

HA

Groups

Fencing

Firewall

Support

CreateEditRemove

Group ↑	restricted	nofailback	Nodes	Comment
mygroup1	No	No	node1:2,node3:1,node4,node2:1	complex group
mygroup2	Yes	No	node1,node2	simple restricted group
prefer_node1	No	No	node1	prefer node1

HA 的组配置文件/etc/pve/ha/groups.cfg 用于定义集群节点服务器组。一个资源可以被指定只能在一个组内的节点上运行。组配置示例如下：

```
group: <group>
    nodes <node_list>
    <property> <value>
    ...
```

comment: <string>

描述信息。

nodes: <node>[:<pri>]{,<node>[:<pri>]}*

节点组成员列表，其中每个节点都可以被赋予一个优先级。绑定在一个组上的资源会优先选择在最高优先级的节点上运行。如果有多个节点都被赋予最高优先级，资源将会被平均分配到这些节点上运行。优先级的值只有相对大小意义。

nofailback: <boolean> (default = 0)

CRM 会尝试在最高优先级的节点运行资源。当有更高优先级的节点上线后，CRM 将把资源迁移到更高优先级节点。设置 `nofailback` 后，CRM 将继续保持资源在原节点运行。

restricted: <boolean> (default = 0)

绑定到 `restricted` 组的资源将只能够在该组的节点运行。如果该组的节点全部关机，则相关资源将停止运行。而对于非 `restricted` 组而言，如果该组的节点全部关机，相关资源可以转移到集群内的任何节点运行，一旦该组节点重新上线，相关资源会立刻迁移回到该组节点上运行。可以通过设置只有一个成员的非 `restricted` 组实现更好表现。

Create: HA Group

ID: restricted: ☐
nofailback: ☐

Comment:

<input type="checkbox"/> Node ↑	Memory usage %	CPU usage	Priority
<input type="checkbox"/> elsa	7.3 %	0.7% of 8CPUs	<input type="text" value=""/>

指定资源在固定节点运行是很常见的做法，但通常也会允许资源在其他节点运行。为此，你可以设置一个只有一个节点的非 `restricted` 组：

```
# ha-manager groupadd prefer_node1 --nodes node1
```

对于节点较多的集群而言，可以考虑制定更加详尽的故障转移策略。例如，你可以指定一组资源固定在 `node1` 节点运行。一旦 `node1` 节点不可用，你可以将相关资源平均分配到 `node2` 和 `node3` 节点运行。如果 `node2` 和 `node3` 也遭遇故障，则可以进一步转移到 `node4` 运行。为达到该效果，你可以设置节点列表如下：

```
# ha-manager groupadd mygroup1 -nodes "node1:2,node2:1,node3:1,node4"
```

另一个例子是，如果某个资源需要用到只有特定节点，比如 node1 和 node2，才具有的硬件或其他资源。我们就需要确保 HA 管理器不在其他节点运行该资源。为此，我们需要创建一个由指定节点构成的 restricted 组：

```
# ha-manager groupadd mygroup2 -nodes "node1,node2" -restricted
```

以上命令创建的配置文件如下：

配置文件示例 (/etc/pve/ha/groups.cfg)

```
group: prefer_node1
    nodes node1

group: mygroup1
    nodes node2:1,node4,node1:2,node3:1

group: mygroup2
    nodes node2,node1
    restricted 1
```

选项 nofailback 主要用于在管理操作中避免意外的资源迁移。例如，如果你需要将一个资源迁移到一个优先级较低的节点运行，就需要设置 nofailback 选项来告诉 HA 管理器不要立刻把资源迁移回原来的高优先级节点。

另一种可能场景是，在节点因故障被隔离后，相关资源会自动迁移到其他节点运行，而管理员在把故障节点重新恢复加入集群后，可能会希望先查明故障原因并检测该节点是否能稳定运行。这时可以设置 nofailback 选项组织 HA 管理器立刻把相关资源迁移故障节点运行。

12.6 隔离

在节点发生故障后，隔离能够确保故障节点彻底离线。这样做主要是为了避免在其他节点恢复资源运行时重复运行同一个资源。这是非常重要的，如果不能确保隔离故障节点，就不可能在其他节点安全恢复资源运行。

如果节点没有被隔离，该节点就可能处于一种不可知的状态，并仍然能够访问集群的共享资源。而这是非常危险的！想象一下这种情形，如果隔离切断了故障节点的所有网络连接，但没有切断对存储的访问，现在尽管故障节点不能再访问网络，但其上的虚拟机仍在运行，并能够向共享存储写入数据。

如果我们现在在其他节点再次启动该虚拟机，我们就可能引发危险的竞争条件，因为现在两个节点上的两个虚拟机在同时向同一个镜像写入数据。这样的情况下，很可能会损坏虚拟机的所有数据，并导致整个虚拟机不可用。当然，我们再启动同一个虚拟机的操作很可能会因为存储禁止多次挂载的保护措施而失败。

12.6.1 Proxmox VE 的隔离措施

隔离节点的方法有很多种，例如隔离设备可以切断节点电源或禁止和外部通信。但这种方法往往过于昂贵，并可能导致其他的问题，例如在隔离设备失效时就无法恢复任何服务。

因此我们采用了一种较简便的隔离方法，而没有采用任何外部隔离设备硬件。具体是采用看门狗计时器来实现。

可能的隔离措施

- 外部电源开关
- 通过在交换机禁止外部网络通信来隔离节点
- 基于看门狗的自隔离

自从微控制器诞生以来，看门狗就广泛用于重要系统和具有高可靠性要求的系统中。看门狗通常都是一块独立的简单集成电路，用于检测计算机故障并帮助从故障中恢复。

在正常情况下，ha-manager 会定期重置看门狗计时器，以防止超时。如果发生硬件故障或程序错误，计算机未能重置看门狗，计时器就会超时并触发主机重启（reboot）。

最新的服务器主板一般集成了硬件看门狗，但需要配置后才能使用。如果服务器没有配置硬件看门狗，可以退而求其次使用 Linux 内核的 softdog。软件看门狗不仅可靠，但并不独立于服务器硬件，因此可靠性较硬件看门狗低一些。

12.6.2 硬件看门狗配置

出于安全考虑，所有的硬件看门狗模块默认都是被禁止的。如果不能正确初始化，硬件看门狗就和一枝上了膛的枪一样危险。你可以在/etc/default/pve-ha-namager 中指定硬件看门狗驱动模块来启用硬件看门狗，示例如下：

```
# select watchdog module (default is softdog)
WATCHDOG_MODULE=iTCO_wdt
```

该配置将被 watchdog-mux 服务读取，并在开机时加载指定的模块。

12.6.3 恢复被隔离的服务

当节点发生故障并被成功隔离后，CRM 服务将尝试把资源从故障节点转移到其他节点运行。

资源迁移目标节点的选择，由 group 资源参数配置，当前可用节点列表，各节点当前的运行负载情况共同决定。

CRM 服务首先在用户设定的节点列表（从 group 配置）和当前可用节点列表之间进行交叉比对选出可用节点列表，然后从中选择具有最高优先级的节点，最后再从中选出负载最低的节点作为目标节点。这可以资源迁移导致节点超载的可能性降到最低。

☒ 重要

发生节点故障后，CRM 会将相关资源分配给其他节点继续运行，从而使得这些节点承担更多资源的运行，有可能导致负载过高。特别在小规模集群中有可能发生这种情况。因此，请认真设计你的集群，以确保能处理这种最坏的情况。

12.7 启动失败策略

当一个服务在某节点上启动失败一次或若干次后，将按照启动失败策略进行处置。启动失败策略包括设置在同一节点的重启次数，以及转移到其他节点继续启动之前的重启次数。该策略的目标是避免共享资源临时不可用导致的启动失败。例如，由于网络问题，共享存储在某个节点上暂时不可用，但在其他节点仍然可以正常访问，转移到其他节点运行的策略将允许该资源继续运行。

对每一个服务，都有两个服务启动恢复策略参数可以配置：

`max_restart`

当前节点上重启失败服务的最大尝试次数。默认为 1。

`max_relocate`

在把服务转移到其他节点继续运行之前尝试重启失败服务的最大次数。只有在当前节点尝试重启次数超过 `max_relocate` 后，才会把服务转移到其他节点。默认为 1。

➤ 注意

当服务启动成功后，转移计数器会被重置为 0。也就是说，如果未能排除故障，服务继续重启，只有重启策略反复生效。

12.8 错误恢复

如果经过各种尝试都不能恢复，服务将进入 error 状态。该状态下 HA 组件将不再操作该服务。改变 error 状态的唯一方法就是手工禁用服务：

```
# ha-manager set vm:100 --state disabled
```

该操作也可以通过 WebGUI 界面进行。

从 error 状态恢复的步骤如下：

- 确保资源处于安全并一致的状态（例如：在服务不能停止时强行杀死进程）
- 禁用资源以移除 error 标识
- 修复导致错误的故障
- 排除故障后，重新启动资源。

12.9 软件包升级

升级 ha-manager 时，你应该一个节点一个节点的进行。出于多种原因，永远不要同时升级所有节点。首先，尽管我们会彻底测试 Proxmox VE，但不能确保消除一切 bug，特别在你个性化的安装环境中。逐个节点进行升级，并在升级后检查每个节点的运行情况有助于在发生意外时恢复集群。同时升级所有节点可能导致集群崩溃，并非最佳实践。

此外，Proxmox VE 的 HA 组件在集群节点和本地资源管理器之间采用了请求确认协议来传递命令。在重启时，LRM 将向 CRM 发出请求，冻结其所有服务。这将防止 LRM 重启时避免相关资源被集群访问。这样 LRM 就可以在重启时安全地关闭看门狗。LRM 重启通常发生在软件升级时，当前的主 CRM 需要确认 LRM 的请求，如果不这样做，升级过程持续的时间可能过长，并可能触发看门狗重启服务器。

12.10 节点维护

在维护节点时，例如更换硬件或安装新内核时，可以将节点关机或重启。

12.10.1 关机

关机（断电）通常在需要停止节点一段时间时使用。此时，LRM 将停止其管理的所有服务。也就是说，其他节点将接手继续运行这些服务。

➤ 注意

最新的服务器往往配置了大容量内存。所以我们先停止所有资源运行，然后在其他节点启动，以避免大量内存数据的在线迁移。如果你希望使用在线迁移，你需要在关闭节点前手工执行。

12.10.2 重启

重启节点可使用 `reboot` 命令。这通常在安装新内核后执行。请注意重启和“关机”的区别，重启后节点会很快恢复运行。

重启前，LRM 告诉 CRM 它希望重启，并等待 CRM 将所有资源置于 `freeze` 状态（也就是在[软件包升级](#)时所处于的状态，见 [12.9](#) 节）。这样相关资源就不会迁移到其他节点。想法，重启后 CRM 将在当前节点重启相关资源。

12.10.3 手工迁移资源

最后但不是唯一，你可以在关机或重启前手工把资源迁移到其他节点运行。该方式的好处是你将全程掌控资源运行状态，并且可以决定使用在线迁移或离线迁移。

➤ 注意

请不要杀死 `pve-ha-crm`，`pve-ha-lrm` 或 `watchdog-mux` 等服务。由于它们是基于看门狗的管理服务，这样做可能会导致服务器重启。
