Red Hat Enterprise Linux 9

在 Red Hat OpenStack Platform 上配置 Red Hat High Availability 集群

在 RHOSP 实例上安装并配置 HA 集群和集群资源

Last Updated: 2024-11-28

Red Hat Enterprise Linux 9 在 Red Hat OpenStack Platform 上配置 Red Hat High Availability 集群

在 RHOSP 实例上安装并配置 HA 集群和集群资源

法律通告

摘要

您可以使用红帽高可用性附加组件在 Red Hat OpenStack Platform (RHOSP)实例上配置高可用性 (HA)集群。此标题提供了安装所需软件包和代理的说明，以及配置基本集群、隔离资源和 HA 集群资源的示例。

目录

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 点顶部导航栏中的 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您的改进建议。包括到文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 前言

您可以使用 Red Hat High Availability Add-On 在 Red Hat OpenStack Platform (RHOSP)实例上配置 Red Hat High Availability (HA)集群。这要求您安装所需的软件包和代理，配置基本集群、配置隔离资源并配置 HA 集群资源。

有关 RHOSP 文档，请参阅 Red Hat Openstack Platform 的产品文档。

有关适用于在 RHEL High Availability 集群中使用 RHOSP 实例的红帽策略、要求和限制，请参阅 RHEL 高可用性集群的支持策略 - OpenStack 虚拟机作为集群成员 - 红帽客户门户网站。

第 2 章 HA 实例的 RHOSP 服务器组配置

在创建 RHOSP HA 集群节点实例前，创建一个实例服务器组。按关联性策略对实例进行分组。如果配置多个集群，请确保每个集群只有一个服务器组。

您为服务器组设置的关联性策略可以决定集群是否在 hypervisor 失败时保持正常运行。

默认关联性策略是 **affinity**。使用这个关联性策略，所有集群节点都可以在同一 RHOSP hypervisor 上创建。在这种情况下，如果虚拟机监控程序失败，整个集群会失败。因此，为 **anti-affinity** 或 **soft-anti-affinity** 的服务器组设置关联性策略。

- 通过具有 **anti-affinity** 关联性的策略，服务器组每个 Compute 节点只允许一个集群节点。尝试创建比 Compute 节点更多的集群节点会生成错误。虽然此配置根据 RHOSP hypervisor 失败提供最高的保护，但可能需要更多资源来部署大型集群，超过您的可用集群。

- 使用 **soft-anti-affinity** 关联性策略，服务器组在所有计算节点上平均分配集群节点。和 **anti-affinity** 策略相比，这个策略对 hypervisor 失败的保护性要更弱，但它比 **affinity** 关联性策略提供了更高的高可用性。

在确定部署的服务器组关联性策略时，需要在集群的需要与您所拥有的资源间进行平衡。您应考虑以下集群组件：

- 集群中的节点数

- 可用的 RHOSP Compute 节点数量

- 用于使集群可以正常工作的集群仲裁所需的节点数量

有关关联性和创建实例服务器组的信息，请参阅 计算调度程序过滤器 和 命令行界面参考。

第 3 章 安装高可用性和 RHOSP 软件包和代理

在 Red Hat OpenStack Platform (RHOSP)上安装配置 Red Hat High Availability 集群所需的软件包。您必须在要用作集群成员的每个节点上安装软件包。

先决条件

- RHOSP 实例的服务器组用作 HA 集群节点，如 HA 实例的 RHOSP 服务器组配置 中所述
- 每个 HA 集群节点的 RHOSP 实例
  - 实例是服务器组的成员
  - 实例配置为运行 RHEL 9.1 或更高版本的节点

流程

1. 启用 RHEL HA 软件仓库和 RHOSP 工具频道。

```
# subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
# subscription-manager repos --enable=openstack-17-tools-for-rhel-9-x86_64-rpms
```

2. 安装 Red Hat High Availability Add-On 软件包以及 RHOSP 集群资源代理和 RHOSP 隔离代理所需的软件包。

```
# dnf install pcs pacemaker python3-openstackclient python3-novaclient fence-agents-openstack
```

3. 在每个节点上安装 **pcs** 和 **pacemaker** 软件包会创建用户 **hacluster**，即 **pcs** 管理帐户。在所有集群节点上为用户 **hacluster** 创建密码。对所有节点使用相同的密码简化了集群管理。

```
# passwd hacluster
```

4. 如果安装了 **firewalld.service**，在 RHEL 防火墙中添加高可用性服务。

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

5. 启动 **pcs** 服务，并使其在引导时启动。

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

6. 验证 **pcs** 服务正在运行。

```
# systemctl status pcsd.service
pcsd.service - PCS GUI and remote configuration interface
Loaded: loaded (/usr/lib/systemd/system/pcsd.service; enabled; vendor preset: disabled)
Active: active (running) since Thu 2018-03-01 14:53:28 UTC; 28min ago
Docs: man:pcsd(8)
man:pcs(8)
Main PID: 5437 (pcsd)
CGroup: /system.slice/pcsd.service
```

```
    └─5437 /usr/bin/ruby /usr/lib/pcsd/pcsd > /dev/null &
Mar 01 14:53:27 ip-10-0-0-48.ec2.internal systemd[1]: Starting PCS GUI and remote
configuration interface…
Mar 01 14:53:28 ip-10-0-0-48.ec2.internal systemd[1]: Started PCS GUI and remote
configuration interface.
```

7. 编辑 **/etc/hosts** 文件并添加 RHEL 主机名和内部 IP 地址。有关 **/etc/hosts** 的详情，请查看红帽知识库解决方案：如何在 RHEL 集群节点上设置 /etc/hosts 文件？

其他资源

- 有关配置和管理红帽高可用性集群的更多信息，请参阅 配置和管理高可用性集群。

第 4 章 为 RHOSP 设置一个验证方法

高可用性隔离代理和资源代理支持与 RHOSP 通信的三种身份验证方法：

- 使用 **clouds.yaml** 配置文件进行身份验证
- 使用 OpenRC 环境脚本进行身份验证
- 通过 Pacemaker 使用 **username** 和 password 进行身份验证

确定用于集群的身份验证方法后，在创建隔离或集群资源时指定适当的身份验证参数。

4.1. 使用 CLOUDS.YAML 文件，使用 RHOSP 进行身份验证

本文档中使用 **clouds.yaml** 文件进行身份验证的流程，使用此流程中显示的 **clouds.yaml** 文件。这些过程为 **cloud= parameter** 指定 **ha-example**，具体如此文件中所定义。

流程

1. 在属于集群的一部分的每个节点上，创建一个 **clouds.yaml** 文件，如下例所示。有关创建 **clouds.yaml** 文件的详情，请参考 用户和身份管理指南。

```
$ cat .config/openstack/clouds.yaml
clouds:
  ha-example:
    auth:
      auth_url: https://<ip_address>:13000/
      project_name: rainbow
      username: unicorns
      password: <password>
      user_domain_name: Default
      project_domain_name: Default
  <. . . additional options . . .>
  region_name: regionOne
  verify: False
```

2. 使用以下基本的 RHOSP 命令测试身份验证是否成功，以及是否可以访问 RHOSP API，替换您在为 **ha-example** 创建的 **cloud.yaml** 文件中指定的云的名称。如果这个命令没有显示服务器列表，请联络您的 RHOSP 管理员。

```
$ openstack --os-cloud=ha-example server list
```

3. 指定在创建集群资源或隔离资源时的 cloud 参数。

4.2. 使用 OPENRC 环境脚本，使用 RHOSP 进行身份验证

要使用 OpenRC 环境脚本通过 RHOSP 进行身份验证，请执行以下步骤。

流程

1. 在作为集群一部分的每个节点中，配置 OpenRC 环境脚本。有关创建 OpenRC 环境变量的详情，请参考 使用 OpenStack RC 文件设置环境变量。

2. 测试身份验证是否成功，并可使用以下基本 RHOSP 命令访问 RHOSP API。如果这个命令没有显示服务器列表，请联络您的 RHOSP 管理员。

```
$ openstack server list
```

3. 指定在创建集群资源或隔离资源时的 **openrc** 参数。

4.3. 通过 USERNAME 和 PASSWORD ， 使用 RHOSP 进行身份验证

要通过 **username** 和 password ， 使用 RHOSP 进行身份验证，请在创建资源时，为集群资源指定 **username**、**password** 和 **auth_url** 参数，或为隔离资源指定隔离资源。根据 RHOSP 配置，可能需要其他身份验证参数。RHOSP 管理员提供要使用的身份验证参数。

第 5 章 在 RED HAT OPENSTACK PLATFORM 上创建基本集群

此流程在没有配置隔离或资源的 RHOSP 平台上创建高可用性集群。

先决条件

- 为每个 HA 集群节点配置一个 RHOSP 实例
- HAcluster 节点正在运行 RHEL 9.1 或更高版本
- 在每个节点上安装高可用性和 RHOSP 软件包，如 安装高可用性和 RHOSP 软件包和代理中所述。

流程

1. 在其中一个群集节点上，输入以下命令验证 **pcs** 用户 **hacluster**。指定集群中的每个节点的名称。在本例中，集群的节点为 **node01**、**node02** 和 **node03**。

```
[root@node01 ~]# pcs host auth node01 node02 node03
Username: hacluster
Password:
node01: Authorized
node02: Authorized
node03: Authorized
```

2. 创建集群。在本例中，集群名为 **newcluster**。

```
[root@node01 ~]# pcs cluster setup newcluster node01 node02 node03

...

Synchronizing pcsd certificates on nodes node01, node02, node03…
node02: Success
node03: Success
node01: Success
Restarting pcsd on the nodes in order to reload the certificates…
node02: Success
node03: Success
node01: Success
```

验证

1. 启用集群。

```
[root@node01 ~]# pcs cluster enable --all
node01: Cluster Enabled
node02: Cluster Enabled
node03: Cluster Enabled
```

2. 启动集群。命令的输出指示集群是否在每个节点上已启动。

```
[root@node01 ~]# pcs cluster start --all
node02: Starting Cluster…
node03: Starting Cluster…
```

```
node01: Starting Cluster...
```

第 6 章 为 RED HAT OPENSTACK PLATFORM 上的 HA 集群配置隔离

隔离配置确保 HA 集群上的故障节点被自动隔离。这可防止节点消耗集群的资源或影响集群的功能。

使用 **fence_openstack** 隔离代理为 RHOSP 上的 HA 集群配置隔离设备。您可以使用以下命令查看 RHOSP 隔离代理的选项。

```
# pcs stonith describe fence_openstack
```

先决条件

- 在 RHOSP 上运行配置的 HA 集群
- 使用您要用于集群配置的 RHOSP 验证方法访问 RHOSP API，如 为 RHOSP 设置身份验证方法 中所述
- 集群属性 **stonith-enabled** 设置为 **true**，这是默认值。红帽不支持禁用了隔离功能的集群，因为它不适用于生产环境。运行以下命令确保隔离已启用。

```
# pcs property config --all
Cluster Properties:
. . .
stonith-enabled: true
```

流程

从集群中的任何节点完成以下步骤。

1. 确定集群中每个节点的 UUID。
   以下命令显示 **ha-example** 项目中所有 RHOSP 实例名称的完整列表，以及标题 **ID** 下与该 RHOSP 实例关联的集群节点的 UUID。节点主机名可能与 RHOSP 实例名称不匹配。

```
# openstack --os-cloud="ha-example" server list
…
| ID                                   | Name            |...
| 6d86fa7d-b31f-4f8a-895e-b3558df9decb|testnode-node03-vm|...
| 43ed5fe8-6cc7-4af0-8acd-a4fea293bc62|testnode-node02-vm|...
| 4df08e9d-2fa6-4c04-9e66-36a6f002250e|testnode-node01-vm|...
```

2. 创建隔离设备，使用 **pcmk_host_map 参数**将集群中的每个节点映射到该节点的 UUID。以下么个示例隔离设备创建命令都使用不同的身份验证方法。

   a. 以下命令使用 **clouds.yaml** 配置文件为 3 节点集群创建 **fence_openstack** 隔离设备。对于 **cloud= parameter**，在 clouds.yaml' 文件中指定云名称。

```
# pcs stonith create fenceopenstack fence_openstack
pcmk_host_map="node01:4df08e9d-2fa6-4c04-9e66-
36a6f002250e;node02:43ed5fe8-6cc7-4af0-8acd-a4fea293bc62;node03:6d86fa7d-
b31f-4f8a-895e-b3558df9decb" power_timeout="240" pcmk_reboot_timeout="480"
pcmk_reboot_retries="4" cloud="ha-example"
```

   b. 以下命令使用 OpenRC 环境脚本创建一个 **fence_openstack** 隔离设备。

```
# pcs stonith create fenceopenstack fence_openstack
pcmk_host_map="node01:4df08e9d-2fa6-4c04-9e66-
36a6f002250e;node02:43ed5fe8-6cc7-4af0-8acd-a4fea293bc62;node03:6d86fa7d-
b31f-4f8a-895e-b3558df9decb" power_timeout="240" pcmk_reboot_timeout="480"
pcmk_reboot_retries="4" openrc="/root/openrc"
```

c. 以下命令使用用户名和密码进行身份验证创建 **fence_openstack** 隔离设备。身份验证参数（包括 **username**, **password**, **project_name**, 和 **auth_url**）由 RHOSP 管理员提供。

```
# pcs stonith create fenceopenstack fence_openstack
pcmk_host_map="node01:4df08e9d-2fa6-4c04-9e66-
36a6f002250e;node02:43ed5fe8-6cc7-4af0-8acd-a4fea293bc62;node03:6d86fa7d-
b31f-4f8a-895e-b3558df9decb" power_timeout="240" pcmk_reboot_timeout="480"
pcmk_reboot_retries="4" username="XXX" password="XXX"
project_name="rhelha" auth_url="XXX" user_domain_name="Default"
```

验证

1. 从集群中的一个节点，隔离集群中的不同节点，并检查集群状态。如果隔离的节点离线，则隔离操作成功。

```
[root@node01 ~] # pcs stonith fence node02
[root@node01 ~] # pcs status
```

2. 重启您隔离的节点并检查状态以验证节点是否已启动。

```
[root@node01 ~] # pcs cluster start node02
[root@node01 ~] # pcs status
```

第 7 章 在 RED HAT OPENSTACK PLATFORM 中配置 HA 集群资源

下表列出了用来为 RHOSP 上的 HA 集群配置资源的特定于 RHOSP 资源代理。

openstack-info （必需）	提供对特定于 RHOSP 资源代理的支持。您必须将 **openstack-info** 资源配置为集群的克隆资源，以便运行 **fence_openstack** 隔离代理以外的任何其他特定于 RHOSP 的资源代理。有关配置 **openstack-info** 资源的详情，请参考 在 Red Hat OpenStack Platform 上的 HA 集群中配置 **openstack-info** 资源。
openstack-virtual-ip	配置虚拟 IP 地址资源。有关配置 **openstack-virtual-ip** 资源的详情，请参考 在 Red Hat Openstack Platform 上的 HA 集群中配置虚拟 IP 地址。
openstack-floating-ip	配置浮动 IP 地址资源。有关配置 **openstack-floating-ip** 资源的详情，请参考 在 Red Hat OpenStack Platform 上的 HA 集群中配置浮动 IP 地址。
openstack-cinder-volume	配置块存储资源。有关配置 **openstack-cinder-volume** 资源的详情，请参考 在 Red Hat OpenStack Platform 上的 HA 集群中配置块存储资源。

在配置其他集群资源时，请使用标准 Pacemaker 资源代理。

7.1. 在 RED HAT OPENSTACK PLATFORM 上的 HA 集群中配置 OPENSTACK-INFO 资源（必需）

您必须配置 **openstack-info** 资源，以便运行除 **fence_openstack** 隔离代理之外的任何其他特定于 RHOSP 的资源代理。

这个创建 **openstack-info** 资源的流程使用 **clouds.yaml** 文件进行 RHOSP 身份验证。

先决条件

- 在 RHOSP 上运行配置的 HA 集群
- 使用您要用于集群配置的 RHOSP 验证方法访问 RHOSP API，如 为 RHOSP 设置身份验证方法 中所述

流程

从集群中的任何节点完成以下步骤。

1. 要查看 **openstack-info** 资源代理的选项，请运行以下命令。

```
# pcs resource describe openstack-info
```

2. 创建 **openstack-info** 资源作为克隆资源。在本例中，该资源也称为 **openstack-info**。本例使用 **clouds.yaml** 配置文件，**cloud=** 参数设为 **clouds.yaml** 文件中的云名称。

```
# pcs resource create openstack-info openstack-info cloud="ha-example" clone
```

3. 检查集群状态，以验证资源是否正在运行。

```
# pcs status

Full List of Resources:

    * Clone Set: openstack-info-clone [openstack-info]:
      * Started: [ node01 node02 node03 ]
```

7.2. 在 RED HAT OPENSTACK PLATFORM 上的 HA 集群中配置虚拟 IP 地址

这个为 RHOSP 平台上的 HA 集群创建 RHOSP 虚拟 IP 地址资源的流程，使用 **clouds.yaml** 文件进行 RHOSP 身份验证。

RHOSP 虚拟 IP 资源与 **IPaddr2** 集群资源一起使用。当您配置 RHOSP 虚拟 IP 地址资源时，资源代理可确保 RHOSP 基础架构将虚拟 IP 地址与网络中的集群节点关联。这允许 **IPaddr2** 资源在该节点上正常工作。

先决条件

- 在 RHOSP 上运行配置的 HA 集群
- 用作虚拟 IP 地址的分配的 IP 地址
- 使用您要用于集群配置的 RHOSP 验证方法访问 RHOSP API，如 为 RHOSP 设置身份验证方法 中所述

流程

从集群中的任何节点完成以下步骤。

1. 要查看 **openstack-virtual-ip** 资源代理的选项，请运行以下命令：

```
# pcs resource describe openstack-virtual-ip
```

2. 运行以下命令，以确定您使用的虚拟 IP 地址的子网 ID。在本例中，虚拟 IP 地址为 172.16.0.119。

```
# openstack --os-cloud=ha-example subnet list
+--------------------------------------+ ...  +----------------+
| ID                                   | ...  | Subnet         |
+--------------------------------------+ ...  +----------------+
| 723c5a77-156d-4c3b-b53c-ee73a4f75185 | ...  | 172.16.0.0/24  |
+--------------------------------------+ ...  +----------------+
```

3. 创建 RHOSP 虚拟 IP 地址资源。
以下命令为 IP 地址 172.16.0.119 创建 RHOSP 虚拟 IP 地址资源，指定您在上一步中确定的子网 ID。

```
# pcs resource create ClusterIP-osp ocf:heartbeat:openstack-virtual-ip cloud=ha-example ip=172.16.0.119 subnet_id=723c5a77-156d-4c3b-b53c-ee73a4f75185
```

4. 配置排序和位置约束：

- 确保 **openstack-info** 资源在虚拟 IP 地址资源之前启动

- 确保 **openstack-info** 资源在虚拟 IP 地址资源之前启动。
- 确保虚拟 IP 地址资源运行在与 **openstack-info** 资源相同的节点上。

```
# pcs constraint order start openstack-info-clone then ClusterIP-osp
Adding openstack-info-clone ClusterIP-osp (kind: Mandatory) (Options: first-action=start
then-action=start)
# pcs constraint colocation add ClusterIP-osp with openstack-info-clone
score=INFINITY
```

5. 为虚拟 IP 地址创建 **IPaddr2** 资源。

```
# pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=172.16.0.119
```

6. 配置排序和位置限制，以确保 **openstack-virtual-ip** 资源在 **IPaddr2** 资源之前启动，**IPaddr2** 资源在与 **openstack-virtual-ip** 资源在同一节点上运行。

```
# pcs constraint order start ClusterIP-osp then ClusterIP
Adding ClusterIP-osp ClusterIP (kind: Mandatory) (Options: first-action=start then-
action=start)
# pcs constraint colocation add ClusterIP with ClusterIP-osp
```

验证

1. 验证资源限制配置。

```
# pcs constraint config
Location Constraints:
Ordering Constraints:
  start ClusterIP-osp then start ClusterIP (kind:Mandatory)
  start openstack-info-clone then start ClusterIP-osp (kind:Mandatory)
Colocation Constraints:
  ClusterIP with ClusterIP-osp (score:INFINITY)
  ClusterIP-osp with openstack-info-clone (score:INFINITY)
```

2. 检查集群状态，以验证资源是否正在运行。

```
# pcs status
. . .

Full List of Resources:
  * fenceopenstack     (stonith:fence_openstack):      Started node01
  * Clone Set: openstack-info-clone [openstack-info]:
    * Started: [ node01 node02 node03 ]
  * ClusterIP-osp      (ocf::heartbeat:openstack-virtual-ip):  Started node03
  * ClusterIP  (ocf::heartbeat:IPaddr2):       Started node03
```

7.3. 在 RED HAT OPENSTACK PLATFORM 上的 HA 集群中配置浮动 IP 地址

以下流程为 RHOSP 上的 HA 集群创建浮动 IP 地址资源。此流程使用 **clouds.yaml** 文件进行 RHOSP 身份验证。

先决条件

先决条件

- 在 RHOSP 上运行配置的 HA 集群
- 用作浮动 IP 地址的公共网络上的 IP 地址，由 RHOSP 管理员分配
- 使用您要用于集群配置的 RHOSP 验证方法访问 RHOSP API，如 为 RHOSP 设置身份验证方法 中所述

流程

从集群中的任何节点完成以下步骤。

1. 要查看 **openstack-floating-ip** 资源代理的选项，请运行以下命令。

```
# pcs resource describe openstack-floating-ip
```

2. 在您要用来创建浮动 IP 地址资源的公共网络上查找地址的子网 ID。

   a. 公共网络通常是使用默认网关的网络。运行以下命令以显示默认网关地址。

```
# route -n | grep ^0.0.0.0 | awk '{print $2}'
172.16.0.1
```

   b. 运行以下命令来查找 public 网络的子网 ID。这个命令生成一个 ID 和子网标题的表。

```
# openstack --os-cloud=ha-example subnet list
+--------------------------------------+---+---------------+
| ID                                   |   | Subnet
+--------------------------------------+---+---------------+
| 723c5a77-156d-4c3b-b53c-ee73a4f75185 |   | 172.16.0.0/24 |
+--------------------------------------+-----------------+
```

3. 创建浮动 IP 地址资源，为该地址指定资源和子网 ID 的公共 IP 地址。当您配置浮动 IP 地址资源时，资源代理在公共网络上配置虚拟 IP 地址，并将其与集群节点关联。

```
# pcs resource create float-ip openstack-floating-ip cloud="ha-example"
ip_id="10.19.227.211" subnet_id="723c5a77-156d-4c3b-b53c-ee73a4f75185"
```

4. 配置排序约束，以确保 **openstack-info** 资源在浮动 IP 地址资源之前启动。

```
# pcs constraint order start openstack-info-clone then float-ip
Adding openstack-info-clone float-ip (kind: Mandatory) (Options: first-action=start then-
action=start
```

5. 配置位置约束，以确保浮动 IP 地址资源与 **openstack-info** 资源在同一节点上运行。

```
# pcs constraint colocation add float-ip with openstack-info-clone score=INFINITY
```

验证

1. 验证资源限制配置。

```
# pcs constraint config
Location Constraints:
```

```
Ordering Constraints:
  start openstack-info-clone then start float-ip (kind:Mandatory)
Colocation Constraints:
  float-ip with openstack-info-clone (score:INFINITY)
```

2. 检查集群状态，以验证资源是否正在运行。

```
# pcs status
. . .
Full List of Resources:
  * fenceopenstack      (stonith:fence_openstack):      Started node01
  * Clone Set: openstack-info-clone [openstack-info]:
    * Started: [ node01 node02 node03 ]
  * float-ip    (ocf::heartbeat:openstack-floating-ip):  Started node02
```

7.4. 在 RED HAT OPENSTACK PLATFORM 上的 HA 集群中配置块存储资源

以下流程为 RHOSP 上的 HA 集群创建块存储资源。此流程使用 **clouds.yaml** 文件进行 RHOSP 身份验证。

先决条件

- 在 RHOSP 上运行配置的 HA 集群
- 由 RHOSP 管理员创建的块存储卷
- 使用您要用于集群配置的 RHOSP 验证方法访问 RHOSP API，如 为 RHOSP 设置身份验证方法 中所述

流程

从集群中的任何节点完成以下步骤。

1. 要查看 **openstack-cinder-volume** 资源代理的选项，请运行以下命令：

```
# pcs resource describe openstack-cinder-volume
```

2. 确定您要配置为集群资源的块存储卷卷 ID。
运行以下命令以显示包括每个卷的 UUID 和名称的可用卷表。

```
# openstack --os-cloud=ha-example volume list
| ID                                   | Name                    |
| 23f67c9f-b530-4d44-8ce5-ad5d056ba926| testvolume-cinder-data-disk |
```

如果您已经知道卷名称，您可以运行以下命令，指定您要配置的卷。这将显示一个带有 ID 字段的表。

```
# openstack --os-cloud=ha-example volume show testvolume-cinder-data-disk
```

3. 创建块存储资源，为卷指定 ID。

```
# pcs resource create cinder-vol openstack-cinder-volume volume_id="23f67c9f-b530-4d44-8ce5-ad5d056ba926" cloud="ha-example"
```

4. 配置排序约束，以确保 **openstack-info** 资源在块存储资源前启动。

```
# pcs constraint order start openstack-info-clone then cinder-vol
Adding openstack-info-clone cinder-vol (kind: Mandatory) (Options: first-action=start then-action=start
```

5. 配置位置约束，以确保块存储资源与 **openstack-info** 资源在同一节点上运行。

```
# pcs constraint colocation add cinder-vol with openstack-info-clone score=INFINITY
```

验证

1. 验证资源限制配置。

```
# pcs constraint config
Location Constraints:
Ordering Constraints:
  start openstack-info-clone then start cinder-vol (kind:Mandatory)
Colocation Constraints:
  cinder-vol with openstack-info-clone (score:INFINITY)
```

2. 检查集群状态，以验证资源是否正在运行。

```
# pcs status
. . .
Full List of Resources:
  * Clone Set: openstack-info-clone [openstack-info]:
    * Started: [ node01 node02 node03 ]
  * cinder-vol  (ocf::heartbeat:openstack-cinder-volume):        Started node03
  * fenceopenstack      (stonith:fence_openstack):       Started node01
```