



Red Hat Enterprise Linux for SAP Solutions 9

使用 RHEL HA 附加组件自动化 SAP HANA 扩展
系统复制

法律通告

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本指南论述了如何在支持的 RHEL 版本的 Pacemaker 集群中在 Scale-Up 中配置 Automated HANA 系统复制。

目录

使开源包含更多	3
对红帽文档提供反馈	4
第 1 章 概述	5
1.1. 支持政策	5
1.2. 所需的订阅和软件仓库	5
第 2 章 配置 SAP HANA 系统复制	7
2.1. 先决条件	7
2.2. 执行初始 SAP HANA 数据库备份	8
2.3. 配置 SAP HANA 主复制实例	8
2.4. 配置 SAP HANA 二级复制实例	8
2.5. 检查 SAP HANA 系统复制状态	9
2.6. 测试 SAP HANA 系统复制	11
第 3 章 配置 HA 集群以管理 SAP HANA SCALE-UP 系统复制设置	12
3.1. 使用 RHEL HA 附加组件安装管理 SAP HANA 扩展系统复制所需的资源代理和其他组件	12
3.2. 启用 SAP HANA SRCONNECTIONCHANGED () HOOK	12
3.3. 配置常规 HA 集群属性	14
3.4. 创建克隆的 SAPHANATOPOLOGY 资源	14
3.5. 创建 PROMOTABLE SAPHANA 资源	16
3.6. 创建虚拟 IP 地址资源	18
3.7. 创建限制	19
3.8. 为 ACTIVE/ACTIVE (READ-ENABLED) SAP HANA 系统复制设置添加辅助虚拟 IP 地址 (可选)	20
3.9. 为 HDBINDEXSERVER 进程失败操作启用 SAP HANA SRSERVICESTATECHANGED () HOOK (可选)	21
第 4 章 测试设置	23
第 5 章 维护步骤	24
5.1. 更新操作系统和 HA 集群组件	24
5.2. 更新 SAP HANA 实例	24
5.3. 手动将 SAPHANA 资源移动到另一节点(SAP HANA 系统复制由 HA 集群接管)	24
第 6 章 参考	25
6.1. RED HAT	25
6.2. SAP	25
6.3. 其他	25

使开源包含更多

红帽承诺替换我们的代码和文档中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于这一努力的精力，这些更改将在即将发布的版本中逐渐实施。[有关让我们的语言更加包含的更多详情，请参阅我们的CTO Chris Wright 信息。](#)

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）

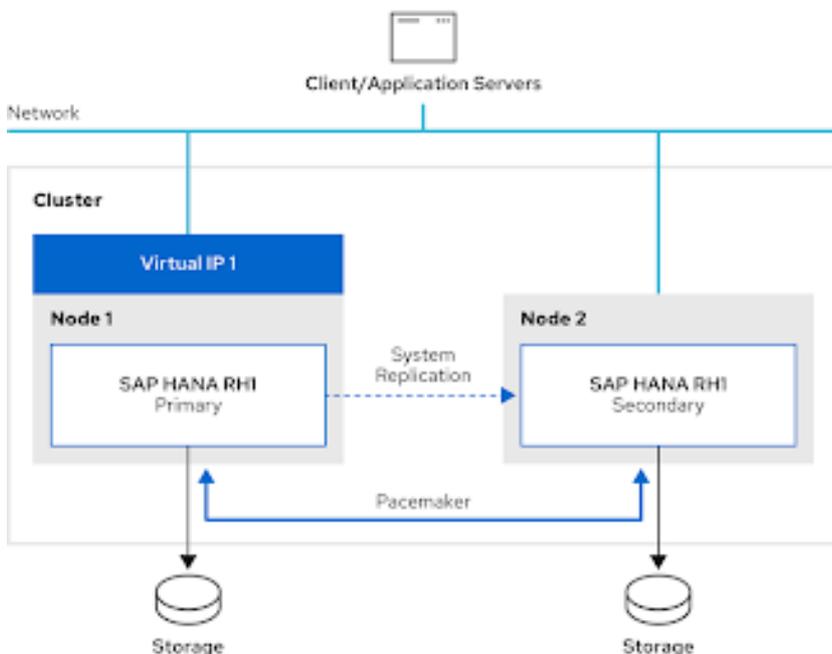
1. 确保您已登录到 [JIRA](#) 网站。
2. 通过单击此[链接](https://issues.redhat.com/secure/CreateInfoDetails!init.jspx?pid=12330720&issuetype=3&components=12387093&priority=10200&summary=Doc&description=775&assignee=rh-ee-pmohta) 来提供反馈。
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括到文档相关部分的链接。
5. 如果要通知将来的更新，请确保已分配为 **Reporter**。
6. 点对话框底部的 **Create**。

第 1 章 概述

本文档论述了如何 [在 RHEL 9 上使用 RHEL HA 附加组件](#) 来设置 HA 集群，以自动执行"性能优化的" SAP HANA Scale-Up 系统复制设置。

"性能优化"意味着每个节点上运行的单个 SAP HANA 实例都控制每个节点上的大多数资源(CPU、RAM)，这意味着 SAP HANA 实例可以尽可能多地运行。由于二级 SAP HANA 实例配置为预加载所有情况，因此当主 SAP HANA 实例故障时应该很快发生。

下图显示了设置类似如下的概述：



206_RHEL_1022

通过"性能优化的" SAP HANA 系统复制设置，也可以使用 [Active/Active \(Read Enabled\) SAP HANA System Replication](#) 配置，这将允许对次要 SAP HANA 实例上的客户端进行只读访问。除了管理 'performance-optimized' SAP HANA Scale-Up System Replication 系统复制的基本设置外，还提供了管理 [Active/Active \(Read Enabled\) SAP HANA Scale-Up System Replication](#) 配置所需的额外集群配置的可选说明。

本文档中描述的资源代理和用于设置的集群配置根据 [SAP Note 2063657 - SAP HANA System Replication over Decision Guideline](#) 提供的指南进行了开发。

本文档不包括运行 SAP HANA 或 SAP HANA 安装过程的 RHEL 9 的安装和配置。请参阅 [为 SAP HANA 2 安装配置 RHEL 9](#) 以了解如何在每个 HA 集群节点上运行 SAP HANA 的信息，并参阅 [SAP HANA 安装指南](#) 和硬件供应商/云供应商的指南来安装 SAP HANA 实例。

本文档中描述的设置是使用内部的"裸机"服务器完成的。如果您计划在 AWS、Azure 或 GCP 等公共云环境中使用此类设置，请查看具体平台的文档：[HA Solutions for 'performance optimized' SAP HANA Scale-Up System Replication- Configuration Guides](#)。

1.1. 支持政策

请参阅 [RHEL 高可用性集群中的支持政策 - 在集群中管理 SAP HANA](#)。

1.2. 所需的订阅和软件仓库

如 SAP Note [3108302 - SAP HANA DB: 推荐的 OS Settings for RHEL 9](#) 所述，每个运行 SAP HANA 的 RHEL 9 系统都需要一个 [RHEL for SAP Solutions](#) 订阅。除了在 RHEL 9 上运行 SAP HANA 的标准仓库外，所有 HA 集群节点还必须启用了 [RHEL HA 附加组件的存储库](#)。启用的仓库列表应该类似如下：

```
[root]# dnf repolist
repo id                repo name                status
rhel-9-for-x86_64-appstream-rpms  Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)
8,603
rhel-9-for-x86_64-baseos-rpms     Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)
3,690
rhel-9-for-x86_64-highavailability-rpms  Red Hat Enterprise Linux 9 for x86_64 - High Availability
(RPMs) 156
rhel-9-for-x86_64-sap-solutions-rpms  Red Hat Enterprise Linux 9 for x86_64 - SAP Solutions
(RPMs) 10
```

有关如何确保每个 HA 集群节点上启用了正确的订阅 [和仓库的更多信息](#)，请参阅 [RHEL for SAP Subscriptions](#) 和 [Repositories](#)。

第 2 章 配置 SAP HANA 系统复制

在配置 HA 集群前，必须根据 SAP: SAP HANA 系统复制：配置，对 [SAP HANA 系统复制进行配置和测试](#)。

以下示例演示了如何在稍后要成为管理 SAP HANA 系统复制设置的 HA 集群的节点上启用 SAP HANA 系统复制。

有关如何确保每个 HA 集群节点上启用了正确的订阅和仓库的更多信息，请参阅 [RHEL for SAP 订阅](#) 和 [存储库](#)。

示例中使用的 SAP HANA 配置：

```
SID: RH1
Instance Number: 02
node1 FQDN: node1.example.com
node2 FQDN: node2.example.com
node1 SAP HANA site name: DC1
node2 SAP HANA site name: DC2
SAP HANA 'SYSTEM' user password: <HANA_SYSTEM_PASSWORD>
SAP HANA administrative user: rh1adm
```

2.1. 先决条件

确保两个系统都可以解决这两个系统的 FQDN，而无问题。要确保在没有 DNS 的情况下解析 FQDN，您可以将其放在 `/etc/hosts` 中，如下例所示：

```
[root]# cat /etc/hosts
...
192.168.0.11 node1.example.com node1
192.168.0.12 node2.example.com node2
```



注意

如主机名 | [SAP Help Portal SAP HANA](#) 中所述，只支持带有小写字符的主机名。

要使系统复制正常工作，SAP HANA `log_mode` 变量必须设置为 `normal`，这也是默认值。如需更多信息，请参阅 [SAP Note 3221437 - 系统复制失败，因为 "Connection refused: Primary 必须以正常方式运行日志模式！](#)”。这可以在这两个节点上使用以下命令，以 SAP HANA 管理用户身份验证。

```
[rh1adm]$ hdbsql -u system -p <HANA_SYSTEM_PASSWORD> -i 02 "select value from
'SYS'.'M_INIFILE_CONTENTS' where key='log_mode'"
VALUE "normal"
1 row selected
```

SAP HANA 管理用户对安装期间选择的 SID 执行了大量配置步骤。对于本文档中描述的示例设置，用户 id `rh1adm` 用于 SAP HANA 管理用户，因为使用的 SID 是 `RH1`。

要从 root 用户切换到 SAP HANA 管理用户，您可以使用以下命令：

```
[root]# sudo -i -u rh1adm
[rh1adm]$
```

2.2. 执行初始 SAP HANA 数据库备份

只有在 HANA 实例上执行初始备份后，SAP HANA 系统复制将作为 SAP HANA 系统复制设置的主要实例后，才会工作。

下面显示了一个在 `/tmp/foo` 目录中创建初始备份的示例。

请注意，备份的大小取决于数据库大小，可能需要一些时间才能完成。放置备份的目录必须可由 SAP HANA 管理用户写入。

在单租户 SAP HANA 设置上，以下命令可用于创建初始备份：

```
[rh1adm]$ hdbsql -i 02 -u system -p <HANA_SYSTEM_PASSWORD> "BACKUP DATA USING FILE
('/tmp/foo')"
0 rows affected (overall time xx.xxx sec; server time xx.xxx sec)
```

在多租户 SAP HANA 设置上，需要备份 **SYSTEMDB** 和所有租户数据库。以下示例演示了如何备份 **SYSTEMDB**：

```
[rh1adm]$ hdbsql -i 02 -u system -p <HANA_SYSTEM_PASSWORD> -d SYSTEMDB "BACKUP
DATA USING FILE ('/tmp/foo')"
0 rows affected (overall time xx.xxx sec; server time xx.xxx sec)
[rh1adm]# hdbsql -i 02 -u system -p <HANA_SYSTEM_PASSWORD> -d SYSTEMDB "BACKUP
DATA FOR RH1 USING FILE ('/tmp/foo-RH1')"
0 rows affected (overall time xx.xxx sec; server time xx.xxx sec)
```

请参阅 SAP HANA 文档来了解如何备份租户数据库。

2.3. 配置 SAP HANA 主复制实例

成功完成初始备份后，使用以下命令初始化 SAP HANA 系统复制：

```
[rh1adm]$ hdbnsutil -sr_enable --name=DC1
checking for active nameserver ...
nameserver is active, proceeding ...
successfully enabled system as system replication source site done.
```

在初始化 SAP HANA 系统复制状态后，验证当前节点是否显示为“主要”：

```
[rh1adm]#$ hdbnsutil -sr_state
checking for active or inactive nameserver ...
System Replication State
~~~~~
mode: primary
site id: 1
site name: DC1
Host Mappings:
```

2.4. 配置 SAP HANA 二级复制实例

在使用与 SAP HANA 主实例相同的 SID 和实例号在其他 HA 集群节点上安装二级 SAP HANA 实例后，需要将其注册到已运行 SAP HANA 主实例中。

变为二级复制实例的 SAP HANA 实例需要首先停止，然后才能注册到主实例：

```
[rh1adm]$ HDB stop
```

当次要 SAP HANA 实例停止后，将 SAP HANA 系统 PKI **SSFS_RH1.KEY** 和 **SSFS_RHDAT** 文件从主 SAP HANA 实例复制到次要 SAP HANA 实例：

```
[rh1adm]$ scp root@node1:/usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1.KEY
/usr/sap/RH1/SYS/global/security/rsecssfs/key/SSFS_RH1.KEY
...
[rh1adm]$ scp root@node1:/usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH1.DAT
/usr/sap/RH1/SYS/global/security/rsecssfs/data/SSFS_RH1.DAT
...
```

如需更多信息，请参阅 [SAP Note 2369981 - 使用 HANA System Replication 进行身份验证所需的配置步骤](#)。

现在，SAP HANA 辅助复制实例可以使用以下命令注册到 SAP HANA 主复制实例：

```
[rh1adm]$ hdbnsutil -sr_register --remoteHost=node1 --remoteInstance=${TINSTANCE} --
replicationMode=syncmem --operationMode=logreplay --name=DC2
adding site ...
checking for inactive nameserver ...
nameserver node2:30201 not responding.
collecting information ...
updating local ini files ...
done.
```

根据您的 HANA 系统复制的要求，请选择 `replicationMode` 和 `operationMode` 的值。如需更多信息，请参阅 [SAP HANA 系统复制和操作模式的复制模式](#)。

注册成功后，可以再次启动 SAP HANA 辅助复制实例：

```
[rh1adm]$ HDB start
```

验证辅助节点是否正在运行，并且 'mode' 与用于 `hdbnsutil -sr_register` 命令中的 `replicationMode` 参数的值匹配。如果注册成功，SAP HANA 辅助复制实例上的 SAP HANA 系统复制状态应类似于如下：

```
[rh1adm]$ hdbnsutil -sr_state
checking for active or inactive nameserver ...
System Replication State
~~~~~
mode: syncmem
site id: 2
site name: DC2
active primary site: 1
Host Mappings:
~~~~~
node2 -> [DC1] node1
node2 -> [DC2] node2
```

2.5. 检查 SAP HANA 系统复制状态

要检查 SAP HANA 系统复制的当前状态，您可以使用 SAP HANA 提供的 **systemReplicationStatus.py** Python 脚本作为当前主 SAP HANA 节点上的 SAP HANA 管理用户。

在单一租户 SAP HANA 设置上，输出应类似于如下：

```
[rh1adm]$ python /usr/sap/RH1/HDB02/exe/python_support/systemReplicationStatus.py
| Host | Port | Service Name | Volume ID | Site ID | Site Name | Secondary | Secondary | Secondary |
Secondary | Secondary | Replication | Replication | Replication | | | | | | | | | | Host | Port | Site ID | Site
Name | Active Status | Mode | Status | Status Details |
| ---- | ---- | ----- | ----- | ---- | ----- | ----- | ----- | ----- | ----- | ----- | -
----- | ----- | ----- | | node1 | 30201 | nameserver | 1 | 1 | DC1 | node2 | 30201 | 2 |
DC2 | YES | SYNCMEM | ACTIVE | |
| node1 | 30207 | xsengine | 2 | 1 | DC1 | node2 | 30207 | 2 | DC2 | YES | SYNCMEM | ACTIVE | |
| node1 | 30203 | indexserver | 3 | 1 | DC1 | node2 | 30203 | 2 | DC2 | YES | SYNCMEM | ACTIVE | |

status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~
mode: PRIMARY
site id: 1
site name: DC1
```

在多租户 SAP HANA 设置中，输出应类似于如下：

```
[rh1adm]$ python /usr/sap/RH1/HDB02/exe/python_support/systemReplicationStatus.py
| Database | Host | Port | Service Name | Volume ID | Site ID | Site Name | Secondary | Secondary |
Secondary | Secondary | Secondary | Replication | Replication | Replication | | | | | | | | | | Host | Port |
Site ID | Site Name | Active Status | Mode | Status | Status Details |
| ---- | ---- | ---- | ----- | ----- | ---- | ----- | ----- | ----- | ----- | ----- | -----
----- | ----- | ----- | ----- |
| SYSTEMDB | node1 | 30201 | nameserver | 1 | 1 | DC1 | node2 | 30201 | 2 | DC2 | YES |
SYNCMEM | ACTIVE | |
| RH1 | node1 | 30207 | xsengine | 2 | 1 | DC1 | node2 | 30207 | 2 | DC2 | YES | SYNCMEM | ACTIVE
| |
| RH1 | node1 | 30203 | indexserver | 3 | 1 | DC1 | node2 | 30203 | 2 | DC2 | YES | SYNCMEM |
ACTIVE | |

status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~
mode: PRIMARY
site id: 1
site name: DC1
```

在这两种情况下，还要检查返回码：

```
echo $?
15
```

返回代码 15（活跃）是正常的。14 表示同步，13 是初始化的。

2.6. 测试 SAP HANA 系统复制

测试阶段是一个非常重要的阶段，用于验证是否满足 KPI，环境是否执行配置方式。如果 SAP HANA System Replication 设置在没有 HA 集群的情况下无法正常工作，则可能会导致在稍后配置 HA 集群时造成意外行为，以管理 SAP HANA 系统复制设置。

因此，建议您根据具体要求对一些测试情况进行改进。测试应该使用实际的数据负载和大小来执行。

测试问题单	描述
完整复制	测量初始同步所需的时间，从时间开始
丢失的连接	测量到 primary 和 secondary 所需的时间
Inover	测量次要系统完全所需的时间
数据一致性	创建或更改数据，然后执行接管并检查数据是否仍然可用。
客户端连接	在接管后测试客户端访问，以检查 DNS/Virtual IP 交换机是否正常工作。
主成为辅助	测量两个系统同步所需的时间，前者主要在接管后成为次要系统。

请参阅"9 节"部分。测试" [如何为 SAP HANA 执行系统复制](#)，了解更多信息。

第 3 章 配置 HA 集群以管理 SAP HANA SCALE-UP 系统复制设置

有关在 RHEL 中设置 pacemaker-based HA 集群的一般指导，请参阅以下文档：

- [在 RHEL 9 中配置和管理高可用性集群](#)
- [RHEL 高可用性集群的支持政策](#)

本指南的其余部分将假设以下事项已配置且正常工作：

- 基本 HA 集群会根据官方的红帽文档进行配置，并拥有合适的且正常工作的隔离（请参阅 [隔离/STONITH 支持策略](#)，了解根据设置运行的平台使用哪些隔离机制的指南）。



注意

此解决方案不支持将 `fence_scsi/fence_mpath` 用作隔离/STONITH 机制，因为没有由 HA 集群管理的 SAP HANA 实例访问的共享存储。

- 已配置了 SAP HANA 系统复制，验证 SAP HANA 实例之间的手动接管正常工作。
- 在所有 HA 集群节点上都禁用 SAP HANA 实例的自动启动（SAP HANA 实例的启动和停止将由 HA 集群管理）。



注意

如果由 HA 集群管理的 SAP HANA 实例 [启用了 systemd](#) (SAP HANA 2.0 SPS07 及更高版本)，则需要额外的配置更改，以确保 systemd 不会干扰 HA 集群管理 SAP 实例。请查看 [第 2 部分。Red Hat HA Solutions for SAP 中的 Systemd-Based SAP Startup Framework for information.](#)

3.1. 使用 RHEL HA 附加组件安装管理 SAP HANA 扩展系统复制所需的资源代理和其他组件

设置管理 SAP HANA Scale-Up 系统复制设置的 HA 集群所需的资源代理和其他 SAP HANA 特定组件通过 "RHEL for SAP Solutions" repo 中的 `resource-agents-sap-hana` RPM 软件包提供。

要安装软件包，请使用以下命令：

```
[root]# dnf install resource-agents-sap-hana
```

3.2. 启用 SAP HANA SRCONNECTIONCHANGED () HOOK

如 SAP [实施 HA/DR 提供程序](#) 中所述，SAP HANA 的最新版本提供了所谓的"hooks"，允许 SAP HANA 向某些事件发送通知。`srConnectionChanged ()` hook 可用于改进 HA 集群在发生 SAP HANA 系统复制状态更改时检测的能力，这需要 HA 集群采取行动，并避免数据丢失/数据损坏，防止意外接管这种情况。



注意

当使用 SAP HANA 2.0 SPS0 或更高版本以及 `resource-agents-sap-hana` 软件包的版本时，提供用于支持 `srConnectionChanged ()` hook 的组件，在继续 HA 集群设置前，必须启用 hook。

3.2.1. 验证 `resource-agents-sap-hana` 软件包的版本

请验证是否安装了 RHEL 9 版本所需的 `resource-agents-sap-hana` 软件包的正确版本，以便在 Red Hat Pacemaker 集群中为 RHEL 9 启用 `srConnectionChanged ()` hook，如 [How can can can the srConnectionChanged \(\) hook 可用于改进需要接管的情况的检测](#)，在 Red Hat Pacemaker 集群中，管理 HANA Scale-up 或 Scale-out 系统复制？

3.2.2. 在所有 SAP HANA 实例上激活 `srConnectionChanged ()` hook



注意

需要为所有 HA 集群节点上的每个 SAP HANA 实例执行 `srConnectionChanged ()` hook 的步骤。

1. 停止两个节点上的 HA 集群（只需要在一个 HA 集群节点上运行）：

```
[root]# pcs cluster stop --all
```

验证所有 SAP HANA 实例都已完全停止。

2. 更新每个节点上的 SAP HANA `global.ini` 文件，以启用 SAP HANA 实例（例如，在 `/hana/shared/RH1/global/hdb/custom/config/global.ini` 文件中）使用 hook 脚本：

```
[ha_dr_provider_SAPHanaSR]
provider = SAPHanaSR
path = /usr/share/SAPHanaSR/srHook
execution_order = 1
```

```
[trace]
ha_dr_saphanasr = info
```

3. 在每个 HA 集群节点上，运行以下命令来创建文件 `/etc/sudoers.d/20-saphana`，并添加以下内容以允许 hook 脚本在调用 `srConnectionChanged ()` hook 时更新节点属性。

```
[root]# visudo -f /etc/sudoers.d/20-saphana
```

```
Cmdnd_Alias DC1_SOK = /usr/sbin/crm_attribute -n hana_rh1_site_srHook_DC1 -v SOK -t
crm_config -s SAPHanaSR
Cmdnd_Alias DC1_SFAIL = /usr/sbin/crm_attribute -n hana_rh1_site_srHook_DC1 -v SFAIL -t
crm_config -s SAPHanaSR
Cmdnd_Alias DC2_SOK = /usr/sbin/crm_attribute -n hana_rh1_site_srHook_DC2 -v SOK -t
crm_config -s SAPHanaSR
Cmdnd_Alias DC2_SFAIL = /usr/sbin/crm_attribute -n hana_rh1_site_srHook_DC2 -v SFAIL -t
crm_config -s SAPHanaSR
rh1adm ALL=(ALL) NOPASSWD: DC1_SOK, DC1_SFAIL, DC2_SOK, DC2_SFAIL
Defaults!DC1_SOK, DC1_SFAIL, DC2_SOK, DC2_SFAIL !requiretty
```

将 `rh1` 替换为您的 SAP HANA 安装的小写 SID，并将 `DC1` 和 `DC2` 替换为您的 SAP HANA 站点名称。

有关为什么需要 `Defaults` 设置的更多信息，请参阅 [srHook 属性在管理 SAP HANA 系统复制的 Pacemaker 集群中设置为 SFAIL](#)，即使复制处于健康状态。

4. 在两个 HA 集群节点上手动启动 SAP HANA 实例，而无需启动 HA 集群：

```
[rh1adm]$ HDB start
```

- 验证 hook 脚本是否按预期工作。执行一些操作来触发 hook，如停止 SAP HANA 实例。然后，使用如下方法检查 hook 是否记录任何内容：

```
[rh1adm]$ cdtrace
[rh1adm]$ awk '/ha_dr_SAPHanaSR.*crm_attribute/ { printf "%s %s %s %s\n", $2, $3, $5, $16 }'
nameserver_*
2018-05-04 12:34:04.476445 ha_dr_SAPHanaSR SFAIL
2018-05-04 12:53:06.316973 ha_dr_SAPHanaSR SOK
[rh1adm]# grep ha_dr_*
```



注意

有关如何验证 SAP HANA hook 是否正常工作的更多信息，请参阅 SAP 文档：[安装和配置 HA/DR 提供程序脚本](#)。

- 验证 hook 的功能后，可以再次启动 HA 集群。

```
[root]# pcs cluster start --all
```

3.3. 配置常规 HA 集群属性

为了避免对资源的不必要的故障切换，必须设置 **resource-stickiness** 和 **migration-threshold** 参数的以下默认值（这只需要在一个节点上完成）：

```
[root]# pcs resource defaults update resource-stickiness=1000
[root]# pcs resource defaults update migration-threshold=5000
```



注意

从 RHEL 9 开始，上述命令已弃用。请改用以下命令。

```
[root]# pcs resource defaults update resource-stickiness=1000
[root]# pcs resource defaults update migration-threshold=5000
```

resource-stickiness=1000 将鼓励资源保持运行，而 **migration-threshold=5000** 将导致资源仅在 5000 失败后移至新节点。**5000** 通常足以防止资源被预先切换到另一个节点。这也可确保资源故障切换时间保持在可控制的限制中。

3.4. 创建克隆的 SAPHANATOPOLOGY 资源

SAPHanaTopology 资源代理收集每个节点上 SAP HANA 系统复制的状态和配置的信息。此外，它启动并监控本地 **SAP HostAgent**，这是启动、停止和监控 SAP HANA 实例所需要的。

SAPHanaTopology 资源代理具有以下属性：

属性名称	必需?	默认值	描述
SID	是	null	SAP HANA 安装的 SAP System Identifier (SID) (所有节点必须相同)。示例：RH1
InstanceNumber	是	null	SAP HANA 安装的实例号 (所有节点必须相同)。示例：02

以下是创建 **SAPHanaTopology** 克隆资源的示例命令。

```
[root]# pcs resource create SAPHanaTopology_RH1_02 SAPHanaTopology SID=RH1
InstanceNumber=02 \
op start timeout=600 \
op stop timeout=300 \
op monitor interval=10 timeout=600 \
clone clone-max=2 clone-node-max=1 interleave=true
```

生成的资源应类似如下：

```
[root]# pcs resource config SAPHanaTopology_RH1_02-clone
Clone: SAPHanaTopology_RH1_02-clone
Meta Attrs: clone-max=2 clone-node-max=1 interleave=true Resource: SAPHanaTopology_RH1_02
(class=ocf provider=heartbeat type=SAPHanaTopology)
Attributes: SID=RH1 InstanceNumber=02
Operations: start interval=0s timeout=600 (SAPHanaTopology_RH1_02-start-interval-0s)
stop interval=0s timeout=300 (SAPHanaTopology_RH1_02-stop-interval-0s)
monitor interval=10 timeout=600 (SAPHanaTopology_RH1_02-monitor-interval-10s)
```



注意

为资源操作显示的超时只是示例，可能需要根据实际的 SAP HANA 设置进行调整（例如，大型 SAP HANA 数据库可能需要更长的时间才能启动，因此可能需要增加启动超时）。

启动资源后，您将看到以节点属性的形式存储的信息，这些属性可通过 **pcs status --full** 命令查看。以下是仅在启动 **SAPHanaTopology** 时显示哪些属性的示例。

```
[root]# pcs status --full
...
Node Attributes:
* Node node1:
+ hana_rh1_remoteHost      : node2
+ hana_rh1_roles           : 1:P:master1::worker:
+ hana_rh1_site            : DC1
+ hana_rh1_srmode          : syncmem
+ hana_rh1_vhost           : node1
* Node node2:
+ hana_rh1_remoteHost      : node1
```

```

+ hana_rh1_roles           : 1:S:master1::worker:
+ hana_rh1_site           : DC2
+ hana_rh1_srmode         : syncmem
+ hana_rh1_vhost          : node2
...

```

3.5. 创建 PROMOTABLE SAPHANA 资源

SAPHana 资源代理管理 SAP HANA 实例，它们是 SAP HANA Scale-Up System Replication 的一部分，同时监控 SAP HANA 系统复制的状态。如果 SAP HANA 主复制实例失败，**SAPHana** 资源代理可以根据资源代理参数的设置方式触发 SAP HANA 系统复制。

SAPHana 资源代理具有以下属性：

属性名称	必需？	默认值	描述
SID	是	null	SAP HANA 安装的 SAP System Identifier (SID) (所有节点必须相同)。示例：RH1
InstanceNumber	是	null	SAP HANA 安装的实例号 (所有节点必须相同)。示例：02
PREFER_SITE_TAKEOVER	否	null	资源代理是否希望切换到二级实例，而不是在本地重启主实例？true: 优先使用二级站点；false: do prefer restart local; never: 不发生其他节点。
AUTOMATED_REGISTRAR	否	false	如果发生接管事件，如果以前的主实例应注册为 secondary？"false": no, 则需要手动干预；"true": yes, 则前者主要由资源代理注册为次要实例。

属性名称	必需?	默认值	描述
DUPLICATE_PRIMARY_TIMEOUT	否	7200	如果在集群响应之前出现双主要情况，则需要两个主时间戳之间的时间区别。如果时间差小于时间差距，集群在"WAITING"状态中包含一个或两个实例。这是为了让管理员有机会响应故障转移。如果以前的主崩溃的完整节点会崩溃，则前者主节点将在经过时间差后注册。如果"仅" SAP HANA 实例崩溃，则前者主要将立即注册。在此注册新主后，所有数据都会被系统复制覆盖。

PREFER_SITE_TAKEOVER、**AUTOMATED_REGISTER** 和 **DUPLICATE_PRIMARY_TIMEOUT** 参数必须根据 HA 集群管理的 SAP HANA 系统复制的要求来设置。

通常，**PREFER_SITE_TAKEOVER** 应设为 true，以允许 HA 集群在检测到主 SAP HANA 实例失败时触发接管，因为新 SAP HANA 主实例通常要花费的时间比原始 SAP HANA 主实例完全激活，而将原始 SAP HANA 主实例重新重新加载到内存中。

为了能够在 HA 集群触发后验证新主 SAP HANA 实例上的所有数据是否正确，**AUTOMATED_REGISTER** 应设置为 false。这将让操作员能够切回到旧的主 SAP HANA 实例，以防意外发生；或者，如果接管正确，旧的主 SAP HANA 实例可以注册为新的次要 SAP HANA 实例，从而再次获得 SAP HANA 系统复制工作。

如果 **AUTOMATED_REGISTER** 设为 true，则在发生 HA 集群后，旧的主 SAP HANA 实例将自动注册为新的二级 SAP HANA 实例。这将提高 SAP HANA 系统复制设置的可用性，并防止 SAP HANA 系统复制环境中的"最终"情况。但是，它可能会增加数据表/数据破坏的风险，因为如果 HA 集群触发了接管，即使次要 SAP HANA 实例上的数据没有完全同步，那么旧的主 SAP HANA 实例的自动化注册，因为新的次要 SAP HANA 实例将导致此实例上的所有数据被删除，因此，在发生接管之前没有同步的任何数据。

可以按照以下示例创建用于管理 SAP HANA 实例和 SAP HANA 系统复制的可升级 **SAPHana** 集群资源，如下例所示：

```
[root]# pcs resource create SAPHana_RH1_02 SAPHana SID=RH1 InstanceNumber=02 \
PREFER_SITE_TAKEOVER=true DUPLICATE_PRIMARY_TIMEOUT=7200
AUTOMATED_REGISTER=true \
op start timeout=3600 \
op stop timeout=3600 \
op monitor interval=61 role="Slave" timeout=700 \
op monitor interval=59 role="Master" timeout=700 \
op promote timeout=3600 \
op demote timeout=3600 \
promotable notify=true clone-max=2 clone-node-max=1 interleave=true
```

生成的 HA 集群资源应类似如下：

```
[root]# pcs resource config SAPHana_RH1_02-clone
Clone: SAPHana_RH1_02-clone
Meta Attrs: clone-max=2 clone-node-max=1 interleave=true notify=true promotable=true
Resource: SAPHana_RH1_02 (class=ocf provider=heartbeat type=SAPHana)
Attributes: AUTOMATED_REGISTER=true DUPLICATE_PRIMARY_TIMEOUT=180
InstanceNumber=02 PREFER_SITE_TAKEOVER=true SID=RH1
Operations: methods interval=0s timeout=5 (SAPHana_RH1_02-methods-interval-0s)
            monitor interval=61 role=Slave timeout=700 (SAPHana_RH1_02-monitor-interval-61)
            monitor interval=59 role=Master timeout=700 (SAPHana_RH1_02-monitor-interval-59)
            promote interval=0s timeout=3600 (SAPHana_RH1_02-promote-interval-0s)
            demote interval=0s timeout=3600 (SAPHana_RH1_02-demote-interval-0s)
            start interval=0s timeout=3600 (SAPHana_RH1_02-start-interval-0s)
            stop interval=0s timeout=3600 (SAPHana_RH1_02-stop-interval-0s)
```



注意

资源操作的超时时间只是示例，可能需要根据实际的 SAP HANA 设置进行调整（例如，大型 SAP HANA 数据库可能需要更长的时间才能启动，因此可能需要增加启动超时）。

启动资源并且 HA 集群执行第一个 monitor 操作后，它会添加额外的节点属性来描述节点上 SAP HANA 数据库的当前状态，如下所示：

```
[root]# pcs status --full
...
Node Attributes:
* Node node1:
  + hana_rh1_clone_state      : PROMOTED
  + hana_rh1_op_mode         : delta_datashipping
  + hana_rh1_remoteHost     : node2
  + hana_rh1_roles           : 4:P:master1:master:worker:master
  + hana_rh1_site            : DC1
  + hana_rh1_sync_state      : PRIM
  + hana_rh1_srmode          : syncmem
  + hana_rh1_version         : 2.00.064.00.1660047502
  + hana_rh1_vhost           : node1
  + lpa_rh1_lpt              : 1495204085
  + master-SAPHana_RH1_02    : 150
* Node node2:
  + hana_r12_clone_state     : DEMOTED
  + hana_rh1_op_mode         : delta_datashipping
  + hana_rh1_remoteHost     : node1
  + hana_rh1_roles           : 4:S:master1:master:worker:master
  + hana_rh1_site            : DC2
  + hana_rh1_srmode          : syncmem
  + hana_rh1_sync_state      : SOK
  + hana_rh1_version         : 2.00.064.00.1660047502
  + hana_rh1_vhost           : node2
  + lpa_rh1_lpt              : 30
  + master-SAPHana_RH1_02    : -INFINITY
...
```

3.6. 创建虚拟 IP 地址资源

为了让客户端能够独立于当前运行的 HA 集群节点访问主 SAP HANA 实例，则需要一个虚拟 IP 地址，HA 集群将在运行主 SAP HANA 实例的节点上启用。

要允许 HA 集群管理 VIP，请创建 IP 为 **192.168.0.15** 的 **IPAddr2** 资源。

```
[root]# pcs resource create vip_RH1_02 IPAddr2 ip="192.168.0.15"
```

请使用适当的资源代理来管理虚拟 IP 地址，具体取决于运行 HA 集群的平台。

生成的 HA 集群资源应如下所示：

```
[root]# pcs resource show vip_RH1_02
Resource: vip_RH1_02 (class=ocf provider=heartbeat type=IPAddr2)
Attributes: ip=192.168.0.15
Operations: start interval=0s timeout=20s (vip_RH1_02-start-interval-0s)
            stop interval=0s timeout=20s (vip_RH1_02-stop-interval-0s)
            monitor interval=10s timeout=20s (vip_RH1_02-monitor-interval-10s)
```

3.7. 创建限制

为实现正确的操作，我们需要确保在启动 **SAPHana** 资源之前启动 **SAPHanaTopology** 资源，并且虚拟 IP 地址也存在于运行主 SAP HANA 实例的节点上。

要达到此目的，需要以下限制：

3.7.1. 约束 - 在 SAPHana 前启动 SAPHanaTopology

以下示例将创建约束，该约束强制使用 **这些资源的开始** 顺序。这里有两个值得提及的事情：

- **symmetrical=false** 属性定义我们只关注资源启动，不需要以相反的顺序停止。
- 两个资源(**SAPHana** 和 **SAPHanaTopology**)都有属性 **interleave=true**，它允许在节点上并行启动这些资源。这允许，尽管设置顺序限制，但我们不会等待所有节点启动 **SAPHanaTopology**，但我们可以立即在 **SAPHana** Topology 运行后在任何节点上启动 **SAPHana** 资源。

创建约束的命令：

```
[root]# pcs constraint order SAPHanaTopology_RH1_02-clone then SAPHana_RH1_02-clone
symmetrical=false
```

生成的约束应类似以下示例：

```
[root]# pcs constraint
...
Ordering Constraints:
  start SAPHanaTopology_RH1_02-clone then start SAPHana_RH1_02-clone (kind:Mandatory) (non-
symmetrical)
...
```

3.7.2. 约束 - 将 IPAddr2 资源与 SAPHana 资源的主并置

以下是一个示例命令，它会将 **IPAddr2** 资源与 **SAPHana** 资源并置，该资源被提升为 Master。

```
[root]# pcs constraint colocation add vip_RH1_02 with master SAPHana_RH1_02-clone 2000
```

请注意，约束使用 score 为 2000，而不是默认的 INFINITY。这允许 **IPaddr2** 资源在 **SAPHana** 资源中没有 Master 时保持活动状态，因此仍然可以使用 SAP 管理控制台(MMC)或 SAP Landscape Management (LaMa)等工具来查询 SAP 实例的状态信息。

生成的约束应类似如下：

```
[root]# pcs constraint
...
Colocation Constraints:
vip_RH1_02 with SAPHana_RH1_02-clone (score:2000) (rsc-role:Started) (with-rsc-role:Master)
...
```

3.8. 为 ACTIVE/ACTIVE (READ-ENABLED) SAP HANA 系统复制设置添加辅助虚拟 IP 地址（可选）

从 SAP HANA 2.0 SPS1 开始，SAP HANA 支持 SAP HANA 系统复制的 **Active/Active (启用Read)** 设置，其中 SAP HANA 系统复制设置的辅助实例可用于只读访问。

为了支持此类设置，需要第二个虚拟 IP 地址，以便客户端能够访问二级 SAP HANA 实例。为确保在发生接管后仍然可以访问次要复制站点，HA 集群需要使用可升级 SAPHana 资源的从设备移动虚拟 IP 地址。

要在 SAP HANA 中启用 **Active/Active (Read Enabled)** 模式，必须在注册二级 SAP HANA 实例时将 **operationMode** 设置为 **logreplay_readaccess**。

3.8.1. 创建用于管理二级虚拟 IP 地址的资源

```
[root]# pcs resource create vip2_RH1_02 IPaddr2 ip="192.168.1.11"
```

请使用适当的资源代理来管理虚拟 IP 地址，具体取决于运行 HA 集群的平台。

3.8.2. 创建位置限制

这是为了确保将辅助虚拟 IP 地址放在正确的 HA 集群节点上。

```
[root]# pcs constraint location vip2_RH1_02 rule score=INFINITY hana_rh1_sync_state eq SOK and
hana_rh1_roles eq 4:S:master1:master:worker:master
[root]# pcs constraint location vip2_RH1_02 rule score=2000 hana_rh1_sync_state eq PRIM and
hana_rh1_roles eq 4:P:master1:master:worker:master
```

这些位置限制可确保第二个虚拟 IP 资源具有以下行为：

- 如果主 SAP HANA 实例和次要 SAP HANA 实例都已启动和运行，并且 SAP HANA System Replication 处于同步状态，第二个虚拟 IP 将在运行次要 SAP HANA 实例的 HA 集群节点上处于活动状态。
- 如果次要 SAP HANA 实例没有运行，或者 SAP HANA 系统复制未同步，第二个虚拟 IP 将在运行主 SAP HANA 实例的 HA 集群节点上处于活动状态。当次要 SAP HANA 实例正在运行并且 SAP HANA 系统复制再次同步时，第二个虚拟 IP 将回到运行次要 SAP HANA 实例的 HA 集群节点。

- 如果主 SAP HANA 实例没有运行，并且 HA 集群触发 SAP HANA 接管，第二个虚拟 IP 将继续在同一节点上运行，直到其他节点上的 SAP HANA 实例注册为新的次要且 SAP HANA 系统复制再次同步。

这可最大化第二个虚拟 IP 资源将分配给运行健康 SAP HANA 实例的节点的时间。

3.9. 为 HDBINDEXSERVER 进程失败操作启用 SAP HANA

SRSERVICESTATECHANGED () HOOK (可选)

当 HANA 检测到 indexserver 进程时，它将通过停止并通过内置在 SAP HANA 中的内置功能来自动重新启动它。

然而，在某些情况下，服务可能需要很长时间才能“停止”阶段。在此期间，System Replication 可能会不同步，而 HANA 仍继续工作并接受新的连接。最后，该服务完成 stop-and-restart 进程并恢复。

ChkSrv.py hook 脚本无法等待这个长时间运行的重启（这可能会造成数据一致性的风险），否则 **ChkSrv.py** hook 脚本可以响应该情况并停止 HANA 实例，以便更快地恢复。在启用了自动故障切换的设置中，如果辅助节点处于健康状态，实例会停止会导致启动。否则，恢复将在本地继续，但强制的实例重启会加快它的速度。

在 **global.ini** 配置文件中配置时，SAP HANA 会为实例中的任何事件调用 **ChkSrv.py** hook 脚本。该脚本根据它应用到事件详情的过滤器的结果处理事件并执行操作。这样，它可以区分在从与实例关闭过程中停止相同的进程后被 HANA 停止并重新启动的 HANA 索引 进程。

以下是可以采取的不同可能的操作：

- ignore：此操作只是将解析的事件和决策信息写入专用日志文件，这可用于验证 hook 脚本的作用。
- stop：此操作通过 **sapcontrol** 命令为实例执行安全 **StopSystem**。
- kill：此操作使用默认的信号 **9** 执行 **HDB kill-<signal>** 命令，可以配置它。

请注意，stop 和 kill 操作都会导致已停止的 HANA 实例，但最后的终止速度会更快。

此时，集群会注意到 HANA 资源失败，并以其配置方式对其做出反应；通常是重新启动实例，如果启用，它也负责接管。

3.9.1. 验证 resource-agents-sap-hana 软件包的版本

请验证是否安装了 RHEL 9 版本启用 **srServiceStateChanged ()** hook 的 **resource-agents-sap-hana** 软件包的正确版本，因为 Pacemaker 集群在主 HANA 实例的 **hdbindexserver** 进程挂起/crashes 时不会触发 HANA 系统复制。

3.9.2. 在所有 SAP HANA 实例上激活 srServiceStateChanged () hook



注意

需要为所有 HA 集群节点上的每个 SAP HANA 实例执行激活 **srServiceStateChanged ()** hook 的步骤。

1. 更新每个节点上的 SAP HANA **global.ini** 文件，以启用 SAP HANA 实例（例如，在 **/hana/shared/RH1/global/hdb/custom/config/global.ini** 文件中）使用 hook 脚本：

```
[ha_dr_provider_chksrv]
provider = ChkSrv
path = /usr/share/SAPHanaSR/srHook
execution_order = 2
action_on_lost = stop

[trace]
ha_dr_saphanasr = info
ha_dr_chksrv = info
```

设置可选参数，如下所示：

- **action_on_lost**（默认为 ignore）
- **stop_timeout**（默认为 20）
- **kill_signal**（默认为 9）

以下是 **action_on_lost** 可用选项的说明：

- **忽略**：这启用了该功能，但只启用日志事件。这可用于监控配置环境中的 hook 活动。
- **stop**：这会执行安全 **sapcontrol -nr <nr> -function StopSystem**。
- **kill**：这会对最快的停止执行 HDB **kill-<signal>**。
- 请注意，**stop_timeout** 被添加到 stop 和 kill 操作的命令执行中，而 **kill_signal** 作为 **HDB kill-<signal>** 命令的一部分被使用。

2. 通过重新载入 **HA/DR** 提供程序，在 HANA 运行时激活新的 hook：

```
[rh1adm]$ hdbnsutil -reloadHADRProviders
```

3. 通过检查新的 trace 文件来验证 hook 初始化：

```
[rh1adm]$ cdtrace
[rh1adm]$ cat nameserver_chksrv.trc
```

第 4 章 测试设置

在将 HA 集群设置放入生产之前，需要对其进行全面测试，以验证一切按预期工作，同时允许操作员获得 HA 集群在某些情况下的行为，以及如何在出现问题时将设置重新置于健康状态。

至少应执行以下测试：

- 通过 HA 集群命令手动移动主 SAP HANA 实例。
预期的结果：在 SAP HANA 端触发接管，提升次要 SAP HANA 实例，成为新的主要 SAP HANA 实例。根据 **SAPHana** 资源的 **AUTOMATED_REGISTER** 参数的配置，HA 集群将自动注册以前的主实例，作为新的次要实例，或者操作员将决定之前的主实例应该发生的情况
- 崩溃 HA 集群节点，其中主 SAP HANA 实例正在运行。
预期的结果：应隔离 HA 集群节点，并在 SAP HANA 端触发接管，从而提升二级 SAP HANA 实例，成为新的主要 SAP HANA 实例。根据 **SAPHana** 资源的 **AUTOMATED_REGISTER** 参数的配置，HA 集群将自动注册前主实例作为新次要实例，或者操作员将决定之前的主实例应该发生什么。
- 在 HA 集群之外手动停止主 SAP HANA 实例。
预期的结果：在 SAP HANA 端触发接管，提升次要 SAP HANA 实例，成为新的主要 SAP HANA 实例。根据 **SAPHana** 资源的 **AUTOMATED_REGISTER** 参数的配置，HA 集群将自动注册前主实例作为新次要实例，或者操作员将决定之前的主实例应该发生什么。
- 崩溃运行辅助 SAP HANA 实例的节点。
预期的结果：应隔离 HA 群集节点，并且当 HA 群集节点恢复恢复时，应启动次要 SAP HANA 实例，并且 SAP HANA 系统复制应恢复。
- 在 HA 集群外手动停止二级 SAP HANA 实例
预期的结果：HA 集群应重启辅助 SAP HANA 实例
- 禁用 SAP HANA 系统复制使用的网络连接
预期的结果：HA 集群应该检测到发生 SAP HANA 系统复制失败，但应该保持 SAP HANA 实例在两个节点上运行

第 5 章 维护步骤

5.1. 更新操作系统和 HA 集群组件

如需更多信息，请参阅[将软件更新应用到 RHEL 高可用性或弹性存储集群的建议实践](#)。

5.2. 更新 SAP HANA 实例

如果 SAP HANA System Replication 设置是使用本文档中描述的 HA 集群配置进行管理的，除了更新前后更新 SAP HANA 实例的实际流程外，还需要一些额外的步骤。执行以下步骤：

1. 将 SAPHana 资源置于非受管模式。

```
[root]# pcs resource unmanage SAPHana_RH1_02-clone
```

2. 使用 SAP 提供的流程更新 SAP HANA 实例。
3. 当 SAP HANA 实例更新完成后，并且验证了 SAP HANA System Replication 是否再次工作，需要刷新 SAPHana 资源的状态，以确保集群了解 SAP HANA 系统复制设置的当前状态。

```
[root]# pcs resource refresh SAPHana_RH1_02-clone
```

4. 当 HA 集群正确获取 SAP HANA 系统复制设置的当前状态时，请将 SAPHana 资源重新设置为受管模式，以便 HA 集群能够再次对 SAP HANA 系统复制设置中的任何问题做出反应。

```
[root]# pcs resource manage SAPHana_RH1_02-clone
```

5.3. 手动将 SAPHANA 资源移动到另一节点(SAP HANA 系统复制由 HA 集群接管)

可以通过移动可升级的克隆资源来触发 SAP HANA 系统复制的手动接管：

```
[root]# pcs resource move SAPHana_RH1_02-clone <other-node>
```

注意

完成接管后，以前的 SAP HANA 主实例发生的情况，并且约束已被删除取决于 SAPHana 资源的 **AUTOMATED_REGISTER** 参数的设置：

- 如果 **Automated_REGISTER=true**，则以前的 SAP HANA 主实例将注册为新的次要和 SAP HANA 系统复制将再次激活。
- 如果 **AUTOMATED_REGISTER=false**，则由操作员决定在接管后将之前的 SAP HANA 主实例发生的情况。

第 6 章 参考

6.1. RED HAT

- [为 SAP HANA2 安装配置 RHEL 9](#)
- [在 RHEL 9 中配置和管理高可用性集群](#)
- [RHEL 高可用性集群的支持政策](#)
- [RHEL 高可用性集群的支持政策 - fence/STONITH](#)
- [RHEL 高可用性集群的支持策略 - 在集群中管理 SAP HANA](#)
- [Red Hat HA Solutions for SAP HANA, S/4HANA and NetWeaver based SAP Applications](#)

6.2. SAP

- [SAP HANA 服务器安装和更新指南](#)
- [SAP HANA 系统复制](#)
- [实施 HA/DR 提供程序](#)
- [SAP Note 2057595 - FAQ: SAP HANA 高可用性](#)
- [SAP Note 2063657 - SAP HANA System Replication over Decision Guideline](#)
- [SAP Note 3007062 - FAQ: SAP HANA 和第三方集群解决方案](#)

6.3. 其他

- [为 SAP HANA 使用 Pacemaker 集群准备 - 第 1 部分：基础](#)
- [为 SAP HANA 使用 Pacemaker 集群准备 - 第 2 部分：两个节点失败](#)