Red Hat Enterprise Linux for SAP Solutions 8

配置 SAP HANA 扩展多目标系统复制以实现灾难恢复

Last Updated: 2024-02-02

Red Hat Enterprise Linux for SAP Solutions 8 配置 SAP HANA 扩展多目标系统复制以实现灾难恢复

法律通告

摘要

本指南概述了在扩展更新配置中配置 HANA 系统复制的过程，特别针对灾难恢复场景而量身定制。本指南解决了在多个站点间实施 HANA 系统复制，专注于涵盖三个或更多站点的环境。

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

第 2 章 参数 8

第 3 章 先决条件 9

第 4 章 安装 10
4.1. 使用故障转移测试检查 2 节点基本安装 10
4.2. 在第三个站点上安装 SAP HANA 10
4.3. 将 SAP HANA 系统复制设置为第三个站点 10

第 5 章 测试问题单 15
5.1. 准备测试 16
5.2. 监控环境 17
5.3. 测试 1：故障切换具有活跃第三个站点的主节点 22
5.4. 测试 2：故障切换具有被动第三个站点的主节点 26
5.5. 测试 3：将主站点故障切换到第三个站点 33
5.6. 测试 4：将主节点故障转移到第一个站点 42

第 6 章 有用的命令 55
6.1. SAP HANA 命令 55
6.2. PACEMAKER 命令 82
6.3. RHEL 和常规命令 100

第 7 章 参考 103
7.1. RED HAT 103
7.2. SAP 103

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）

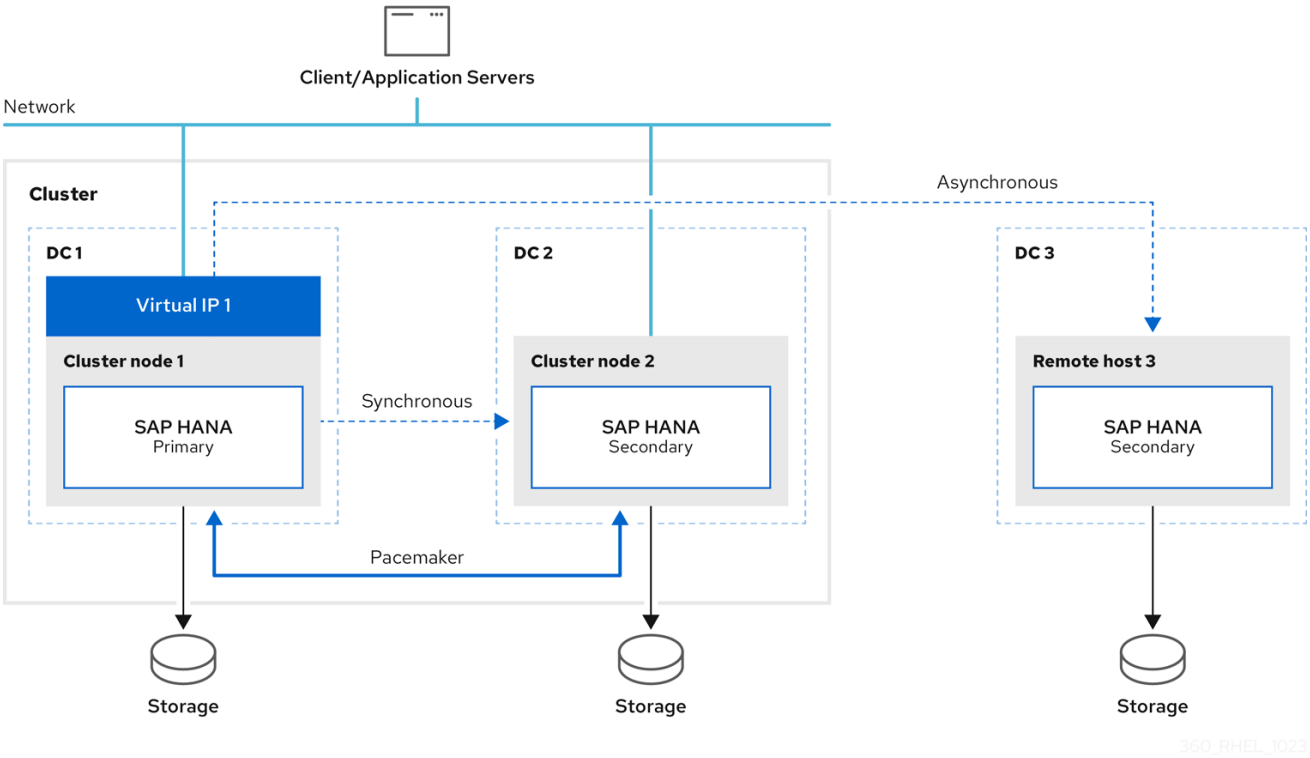
1. 确保您已登录到 JIRA 网站。
2. 通过单击此链接 来提供反馈。https://issues.redhat.com/secure/CreateIssueDetails!init.jspa?pid=12330720&issuetype=3&components=12387093&priority=10200&summary=Doc&description=775&assignee=rh-ee-pmohta
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您对改进的建议。包括到文档相关部分的链接。
5. 如果要通知将来的更新，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

第 1 章 概述

由于对可用性需求不断增长，数据的一个副本不够。
为确保业务连续性，可靠且高度可用的架构必须在多个复制站点之间复制数据。使用多目标系统复制时，主站点可将数据更改复制到多个次要站点。如需更多信息，请参阅 SAP HANA Multitarget System Replication。

本文档论述了如何在 2 节点集群中使用 SAP HANA Multitarget System Replication 配置额外的复制站点，如 使用 RHEL HA 附加组件自动化 SAP HANA Scale-Up 系统复制 中所述。

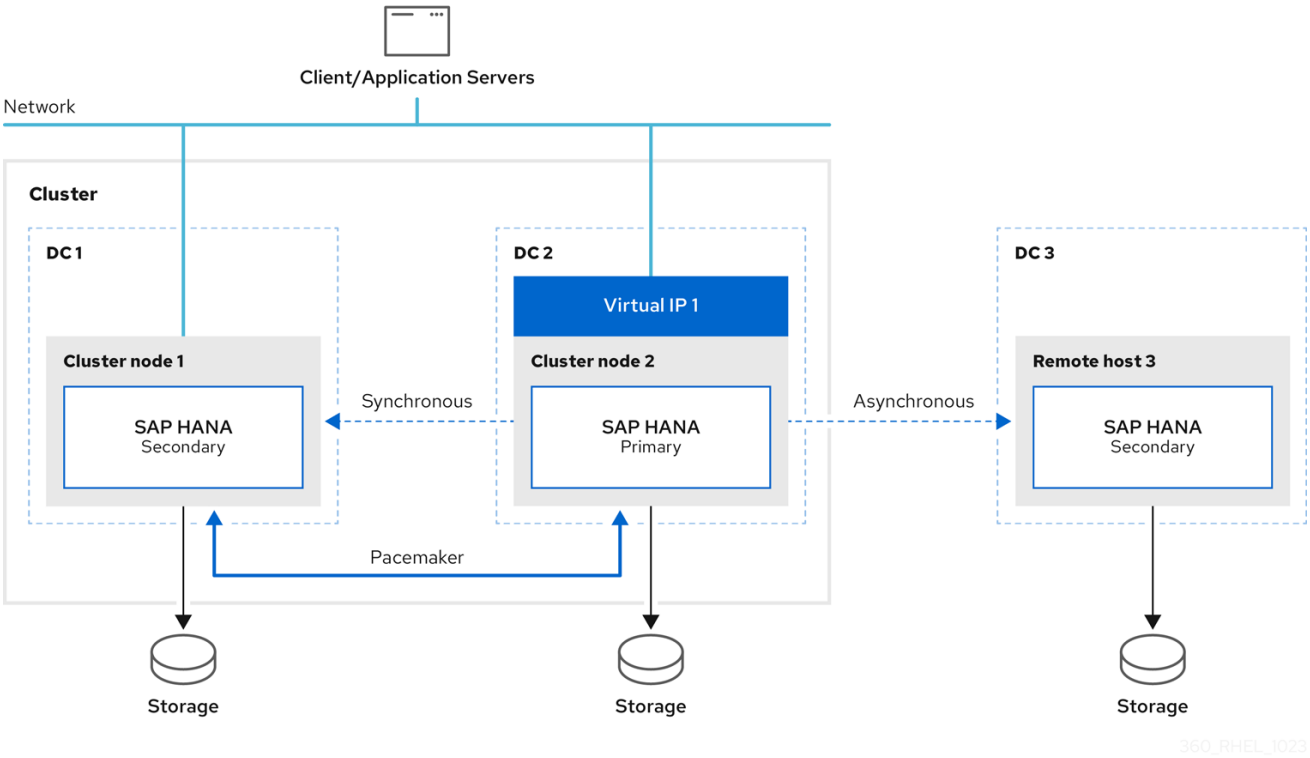
配置示例类似如下：



初始设置如下：

- 将主站点 1 (DC1)复制到次要站点 2 (DC2)
- 将主站点 1 (DC1)复制到次要站点 3 (DC3)

如果主站点失败，则主站点切换到次要站点 2 (DC2)，而以前的主站点 1 (DC1)将变为次要站点。

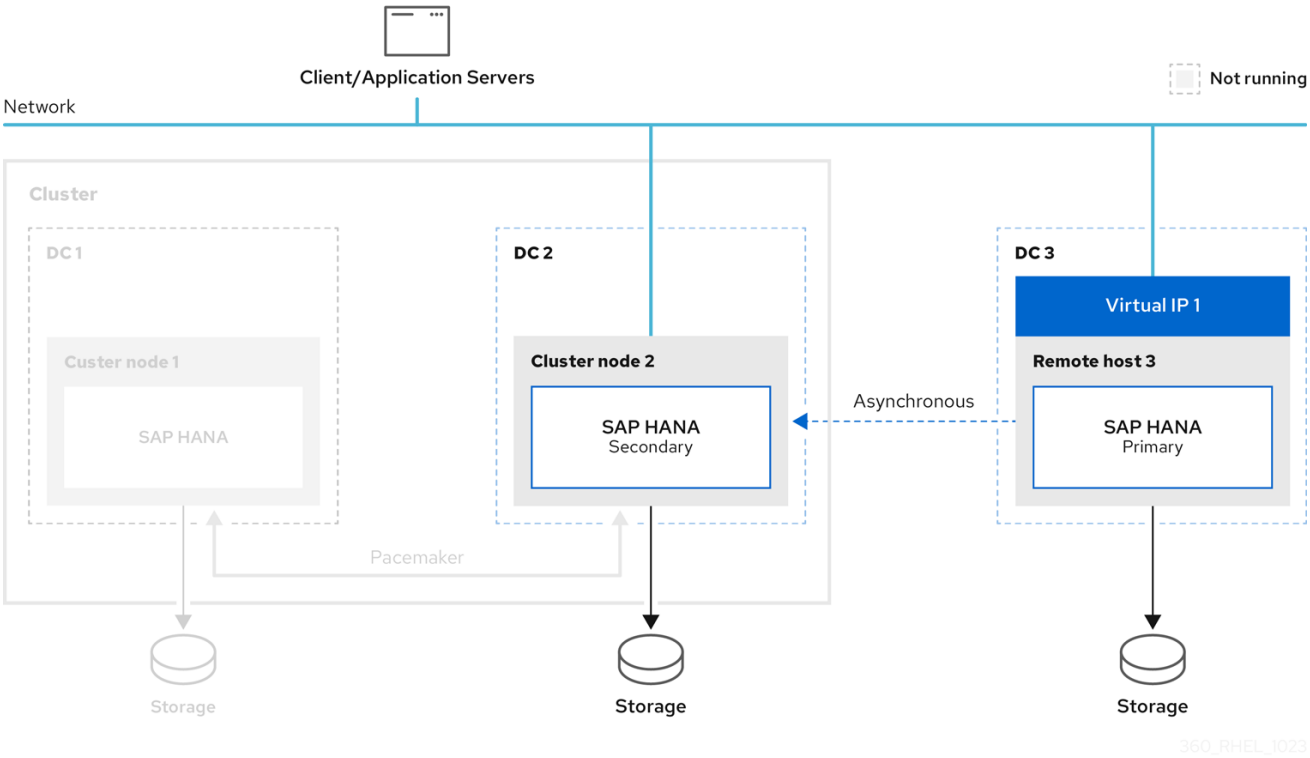


发生故障转移时，此解决方案可确保在第三个站点上切换配置的主站点。故障转移后的配置如下：

- 在 DC2 上运行的主
- 在 DC1 上运行辅助运行（从 DC2 同步）
- 在 DC3 上运行辅助运行（从 DC2 同步）

remotehost3 上的 SAP HANA 实例将自动重新注册到新主，只要此实例在故障转移期间启动并运行。

本文档还描述了将主数据库切换到第三个站点的示例。



请注意，将客户端连接到数据库需要进一步的网络配置。这不在本文档的讨论范围内。

如需更多信息，请检查以下内容：

- SAP HANA 平台的 SAP HANA 管理指南
- 如何设置 SAP HANA 多文件系统复制

第 2 章 参数

用于设置第三个站点，现有双节点集群的这些参数：

参数	示例	描述
SID	RH2	HANA 数据库的系统 ID
第一个 SITE	DC1	第一个数据中心 /site 的名称
第二 SITE	DC2	第二个数据中心/站点的名称
第三 SITE	DC3	第三个数据中心/站点的名称
InstanceNr	02	HANA 实例编号
<SID>adm uid	1000	SAP HANA admin 用户的用户 ID (rh2adm)
sapsys gid	980	sapsys 的组 ID

需要所有三个 HANA 实例都使用相同的值：

- SID
- InstanceNr
- <SID>adm uid
- sapsys gid

第 3 章 先决条件

要解决这一解决方案，必须满足以下要求：

所有节点必须具有相同的：

- CPU 和 RAM 数量
- 软件配置
- RHEL 发行版本（请注意，至少需要 RHEL 8.6）
- 防火墙设置
- SAP HANA 版本(SAP HANA 2.0 SPS04 或更高版本)

pacemaker 软件包仅安装在集群节点上，且必须使用相同的 resource-agents-sap-hana (0.162.1 或更高版本)。

要能够支持 SAP HANA Multitarget System Replication，请参阅 添加 SAP HANA Multitarget System Replication 支持。另外，设置以下内容：

- use **register_secondaries_on_takeover=true**
- use **log_mode=normal**

初始设置基于安装指南，使用 RHEL HA 附加组件自动化 SAP HANA Scale-Up 系统复制 。

所有 SAP HANA 实例的系统复制配置都基于 SAP 要求。如需更多信息，请参阅基于 SAP HANA 管理指南 的 SAP 指南。

第 4 章 安装

本章描述了额外 SAP HANA 实例的安装。

4.1. 使用故障转移测试检查 2 节点基本安装

使用 RHEL HA 附加组件，验证安装是否基于 Automating SAP HANA Scale-Up System Replication 来完成。

为了可以使用 SAP HANA Multitarget System Replication，resource-agents-sap-hana 的版本必须是 0.162.1 或更高版本。这可检查，如下所示：

```
# rpm -q resource-agents-sap-hana
```

您可以运行故障转移测试以确保环境正常工作。您可以移动 SAPHana 资源，它也在 使用 Move 故障切换 SAPHana 资源 中所述。

4.2. 在第三个站点上安装 SAP HANA

在第三个站点上，您还需要使用与在双节点 Pacemaker 集群上为 SAP HANA 实例相同的版本和参数安装 SAP HANA，如下所示：

参数	值
SID	RH2
InstanceNumber	02
<SID>adm 用户 ID	rh2adm 999
sapsys 组 ID	sapsys 999

SAP HANA 安装使用 **hdblcm** 完成。如需了解更多详细信息，请参阅使用 hdbclm 的 SAP HANA 安装。另外，也可使用 Ansible 进行安装。

在本章示例中，我们使用：

- site DC1 上的 hosts:clusternode1，clusternode2 on site DC2 和 remotehost3 on site DC3
- SID RH2
- adminuser rh2adm

4.3. 将 SAP HANA 系统复制设置为第三个站点

在现有安装中，已经在双节点集群中的主和次要 SAP HANA 实例之间配置了 SAP HANA 系统复制。在启动并运行主 SAP HANA 数据库实例上启用了 SAP HANA 系统复制。

本章论述了如何在站点 DC3 的节点 remotehost3 上将第三个 SAP HANA 实例注册为额外的辅助 HANA 系统复制站点。此步骤与在节点 clusternode2 上注册原始辅助 HANA 实例(DC2)类似。以下章节将更详细地阐述。如果需要更多信息，您还可以检查 配置 SAP HANA System Replication 的一般前提条件。

4.3.1. 检查主数据库

您必须检查其他数据库是否正在运行，系统复制是否正常工作。请参阅：

- 检查数据库
- 检查 SAP HANA System Replication 状态
- 发现主要和次要 SAP HANA 数据库

您可以使用以下方法发现主 HANA 实例：

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "primary masters|^mode"
mode: primary
```

4.3.2. 复制数据库密钥

在您能够注册新的 secondary HANA 实例之前，primary HANA 实例的数据库密钥需要复制到新的 additional HANA 复制站点。在我们的示例中，第三个站点的主机名是 remotehost3。

例如，在主节点 clusternode1 上运行：

```
clusternode1:rh2adm> scp -rp
/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/data/SSFS_${SAPSYSTEMNAME}.DAT
remotehost3:/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/data/SSFS_${SAPSYSTEMN
AME}.DAT
clusternode1:rh2adm> scp -rp
/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/key/SSFS_${SAPSYSTEMNAME}.KEY
remotehost3:/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/key/SSFS_${SAPSYSTEMN
AME}.KEY
```

4.3.3. 将额外的 HANA 实例注册为辅助 HANA 复制站点

您需要知道运行 主数据库的 节点的主机名。

要监控注册，您可以在主节点上的独立终端中运行以下命令：

```
clusternode1:rh2adm> watch python
/usr/sap/${SAPSYSTEMNAME}/HDB${TINSTANCE}/python_support/systemReplicationStatus.py
```

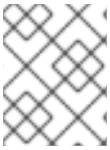
这将显示进度以及错误（如果它们发生）。

要将第 3 个站点(DC3)上注册为额外的次要 SAP HANA 实例，请在第三个站点主机 remotehost3 上运行以下命令：

```
remotehost3:rh2adm> hdbnsutil -sr_register --name=DC3 --remoteHost=clusternode1 --
remoteInstance=${TINSTANCE} --replicationMode=async --operationMode=logreplay --online
```

在本例中，DC3 是第三个站点的名称，clusternode1 是主节点的主机名。

如果数据库实例已在运行，则不必停止它，您可以使用 **--online** 选项，该选项将在实例上线时注册。然后，由 **hdbnsutil** 本身启动实例所需的重启（停止和启动）。



注意

选项 **--online** 在任何情况下都可以正常工作，当 HANA 实例上线并脱机时（此选项都可以使用 SAP HANA 2.0 SPS04 及更高版本）。

如果 HANA 实例离线，您必须在第三个节点注册后启动它。您可以在 SAP HANA System Replication 中找到更多信息。

4.3.4. 添加 SAP HANA Multitarget System Replication autoregister 支持

我们使用 SAP HANA System Replication 选项，称为 **register_secondaries_on_takeover = true**。这将自动重新使用新的主站点重新注册辅助 HANA 实例，以防在之前的主站点和其他次要站点之间进行故障转移。这个选项必须添加到所有潜在的主站点的 **global.ini** 文件中。

所有 HANA 实例都应在 **全局.ini** 中具有此条目：

```
[system_replication]
register_secondaries_on_takeover = true
```

以下两个章节详细描述了 **global.ini** 配置。

小心

尽管参数在启动故障转移时，如果第三个节点上的额外 second HANA 实例 ***停机***，则需要手动重新注册此 HANA 实例。

4.3.5. 在 pacemaker 节点上配置 global.ini

选项 **register_secondaries_on_takeover = true** 需要添加到由 pacemaker 集群管理的 SAP HANA 实例的 **global.ini** 中。请编辑对应节点上的 **global.ini** 文件，并且不要从另一个节点复制文件。

注意

只有在站点的 HANA 实例已停止处理时，才应编辑 **global.ini** 文件。

以 **rh2adm** 用户身份编辑 **global.ini** ：

```
clusternode1:rh2adm> vim /usr/sap/${SAPSYSTEMNAME}/SYS/global/hdb/custom/config/global.ini
```

Example:

```
# global.ini last modified 2023-07-14 16:31:14.120444 by hdbnsutil -sr_register --
remoteHost=remotehost3 --remoteInstance=02 --replicationMode=syncmem --
operationMode=logreplay --name=DC2
[multidb]
mode = multidb
database_isolation = low
singletenant = yes

[ha_dr_provider_SAPHanaSR]
provider = SAPHanaSR
path = /hana/shared/myHooks
execution_order = 1
```

```
[persistence]
basepath_datavolumes = /hana/data/RH2
basepath_logvolumes = /hana/log/RH2
log_mode = normal
enable_auto_log_backup = true

[system_replication]
register_secondaries_on_takeover = true
timetravel_logreplay_mode = auto
operation_mode = logreplay
mode = primary
actual_mode = syncmem
site_id = 1
site_name = DC2

[system_replication_site_masters]
2 = clusternode1:30201

[trace]
ha_dr_saphanasr = info
```

启动 SAP HANA 数据库实例后，此选项就会生效。

4.3.6. 在第三个站点上配置 global.ini

以 <**sid>adm** 用户身份编辑 **global.ini** ：

```
remotehost3:rh2adm> vim /usr/sap/${SAPSYSTEMNAME}/SYS/global/hdb/custom/config/global.ini
```

在 remotehost3 上，不使用 **ha_dr_provider_SAPHanaSR** 部分。

remotehost3 上的 **global.ini** 示例：

```
# global.ini last modified 2023-06-22 17:22:54.154508 by hdbnameserver
[multidb]
mode = multidb
database_isolation = low
singletenant = yes

[persistence]
basepath_datavolumes = /hana/data/RH2
basepath_logvolumes = /hana/log/RH2
log_mode = normal
enable_auto_log_backup = true

[system_replication]
operation_mode = logreplay
register_secondaries_on_takeover = true
reconnect_time_interval = 5
timetravel_logreplay_mode = auto
site_id = 3
mode = syncmem
actual_mode = syncmem
site_name = DC3
```

```
[system_replication_site_masters]
2 = clusternode1:30201
```

4.3.7. 验证安装

安装后，您必须检查所有 HANA 实例是否已启动并在运行，并且 HANA System Replication 在它们之间正常工作。最简单的方法是检查 **systemReplicationStatus**，如检查系统复制 状态 中详细介绍。

如需更多信息，请参阅检查数据库状态。

要使 HANA 系统复制正常工作，请确保"log_mode"参数设为"normal"。如需更多信息，请参阅 检查 SAP HANA 数据库的 log_mode。

要验证设置是否按预期工作，请运行 测试案例，如以下章节中所述。

第 5 章 测试问题单

安装完成后，建议运行一些基本测试来检查安装，并验证 SAP HANA Multitarget System Replication 如何工作，以及如何从故障中恢复。在开始生产之前，最好运行这些测试案例。如果可能，您还可以准备测试环境，以便在启动生产前验证更改。如果可能，您还可以准备测试环境，以便在生产中应用更改前检查更改。

所有情况都将描述：

- 测试的主题
- 测试先决条件
- 测试步骤
- 监控测试
- 启动测试
- 预期结果
- 返回初始状态的方法

要将以前的 primary HANA 复制站点自动注册为由集群管理的 HANA 实例的新次要 HANA 复制站点，您可以使用 SAPHana 资源中的选项 **AUTOMATED_REGISTER=true** 选项。如需了解更多详细信息，请参阅 AUTOMATED_REGISTER。

示例中使用的 HA 集群节点名称和 HANA 复制站点（在括号中）：

- clusternode1 (DC1)
- clusternode2 (DC2)
- remotehost3 (DC3)

以下参数用于配置 HANA 实例和集群：

- SID=RH2
- INSTANCENUMBER=02
- CLUSTERNAME=cluster1

您可以在测试环境中使用 clusternode1-2, remotehost3 作为别名。

测试更为详细，包括示例和其他前提条件检查。最后，有有关如何清理环境以进一步测试的示例。

在某些情况下，如果 clusternode1-2 和 remotehost3 之间的距离太长，您应该使用 -replcationMode=async 而不是 -replicationMode=syncmem。在选择正确的选项前，请询问您的 SAP HANA 管理员。

5.1. 准备测试

在运行测试前，整个环境需要处于正确的且健康的状态。

我们通过以下方式检查集群和数据库：

- pcs status --full
- python $DIR_EXECUTABLE/python_support/systemReplicationStatus.py
- df -h

pcs status --full 的示例可在 Check cluster status 中找到。如果 "Migration Summary" 中存在警告或之前失败，您应该在开始测试前清理集群。

```
[root@clusternode1]# pcs resource clear SAPHana_RH2_02-clone
```

集群清理 描述了进行它的一些更多方法。务必要启动集群及所有资源。

除了集群外，数据库也应该已启动并在同步。验证数据库的正确状态的最简单方法是检查系统复制状态。另请参阅 Replication Status。这应该会在主数据库中检查。

要发现主节点，您可以检查 发现主数据库 或使用：

```
[root@clusternode1]# pcs status | grep -E "Promoted|Master"
[root@clusternode1]# hdbnsutil -sr_stateConfiguration
```

运行以下命令，检查文件系统中是否有足够空间：

```
[root@clusternode1]# df -h
```

在继续操作前，请按照 系统检查 的指南进行操作。如果环境清理干净，就可以运行测试。

5.2. 监控环境

在本节中，我们专注于在测试期间监控环境。本节仅涵盖查看更改所需的 monitor。建议从专用终端运行这些监控器。为了能够在测试期间检测更改，建议在开始测试前启动监控。

在 Useful Commands 部分中，会显示更多示例。

5.2.1. 发现主节点

您需要发现主节点来监控故障转移，或运行某些命令，它们仅在主节点上执行时提供有关复制状态的信息。

要发现主节点，您可以以 < sid>adm 用户身份运行以下命令：

```
clusternode1:rh2adm> watch -n 5 'hdbnsutil -sr_stateConfiguration | egrep -e "primary masters|^mode"'
```

输出示例，当 clusternode2 是主数据库时：

```
mode: syncmem
primary masters: clusternode2
```

在运行主数据库的节点上的输出是：

```
mode: primary
```

5.2.2. 检查 Replication 状态

复制状态显示主数据库节点和次要数据库节点与复制的当前状态之间的关系。

要发现复制状态，您可以以 < sid>adm 用户身份运行：

```
clusternode1:rh2adm> hdbnsutil -sr_stateConfiguration
```

如果要永久监控系统复制状态的变化，请运行以下命令：

```
clusternode1:rh2adm> watch -n 5 'python
/usr/sap/${SAPSYSTEMNAME}/HDB${TINSTANCE}/exe/python_support/systemReplicationStat
us.py ; echo Status $?'
```

这个示例还决定了当前的返回代码。
只要返回代码（状态）为 15，复制状态就正常。其他返回代码包括：

- 10: NoHSR
- 11: error
- 12: unknown
- 13: 初始化
- 14: 同步
- 15: active

如果注册了新的次要设备，您可以在主节点上的单独窗口中运行它，您会看到复制的进度。如果要监控故障转移，您可以在旧主以及新的主数据库服务器上并行运行它。如需更多信息，请参阅 检查 SAP HANA 系统复制状态。

5.2.3. 检查 /var/log/messages 条目

Pacemaker 将大量信息写入 /var/log/messages 文件中。在故障转移过程中，大量信息会被写入这个消息文件中。要根据 SAP HANA 资源代理只遵循重要的消息，过滤 pacemaker SAP 资源的详细活动非常有用。足以检查单个集群节点上的消息文件。

例如，您可以使用这个别名：

```
[root@clusternode1]# tmsl='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register|WAITING4LPA|PROMOTED
|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED|LPT"'
```

在单独的窗口中运行 tmsl 来监控测试的进度。还要检查 monitor 故障切换和同步状态的示例。

5.2.4. 集群状态

有几种方法可以检查集群状态。

- 检查集群是否正在运行：
  - pcs cluster status
- 检查集群和所有资源：
  - pcs status
- 检查集群、所有资源和所有节点属性：
  - pcs status --full
- 仅检查资源：
  - pcs resource

pcs status --full 命令将为您提供所有必要的信息。要监控更改，您可以与 watch 一起运行这个命令：

```
[root@clusternode1]# watch pcs status --full
```

可以在 Check cluster status 中找到输出示例和更多选项。

5.2.5. 发现左侧

为确保您的环境准备好运行下一个测试，需要修复或删除之前测试中的左侧。

- stonith 用于隔离集群中的节点：
  - detect: [root@clusternode1]# pcs stonith history
  - Fix: [root@clusternode1]# pcs stonith cleanup
- 多个主数据库：
  - detect: clusternode1:rh2adm> hdbnsutil -sr_stateConfiguration | grep -i primary

    需要识别具有相同主的所有节点。
  - fix: 使用 --force_full_replica选项重新注册错误的主
- 移动导致的位置限制：
  - detect: [root@clusternode1]# pcs constraint location

    检查警告部分。
  - Fix: [root@clusternode1]# pcs resource clear <clone-resource-which was moved>
- 二级复制关系：
  - detect: on the primary database run clusternode1:rh2adm> python $DIR_EXECUTABLE/python_support/systemReplicationStatus.py

- 修复：取消注册并重新注册辅助数据库。

- 检查 siteReplicationMode （所有 SAP HANA 节点上的相同输出）
  - clusternode1:rh2adm> hdbnsutil -sr_state --sapcontrol=1 |grep site prerequisitesMode

- pcs 属性：
  - detect: [root@clusternode1]# pcs property config
  - Fix: [root@clusternode1]# pcs property set <key=value>

- 清除 maintenance-mode。
  - [root@clusternode1]# pcs property set maintenance-mode=false

- log_mode ：
  - detect: clusternode1:rh2adm> python systemReplicationStatus.py

    将响应 log_mode 通常需要的复制状态。可以检测到 log_mode，如 使用 hdbsql 检查 lnifile 内容 中所述。
  - fix ：将 log_mode 更改为 normal，然后重新启动主数据库。

- CIB 条目：
  - detect: 集群信息基础中的 SFAIL 条目。

请参阅 检查集群一致性，以查找和删除 CIB 条目。

- cleanup/clear:
  - detect: [root@clusternode1]# pcs status --full

    有时它会显示错误或警告。您可以清理/清理/清理资源，如果一切正常，则不会发生。在运行下一个测试前，您可以清理您的环境。

  - 修复示例：

    [root@clusternode1]# pcs resource clear <name-of-the-clone-resource>[root@clusternode1]# pcs resource cleanup <name-of-the-clone-resource>

如果要检查现有环境中是否有问题，这也很有用。

如需更多信息，请参阅 用法 命令。

5.3. 测试 1：故障切换具有活跃第三个站点的主节点

测试的主题	自动重新注册第三个站点。 清除后，将状态更改为 SOK。
测试先决条件	• DC1、DC2、DC3 上的 SAP HANA 正在运行。 • 集群已启动并运行，且没有错误或警告。
测试步骤	使用 **pcs [root@clusternode1]# resource move <sap-clone-resource> <target-node> 命令移动 SAPHana 资源**。
监控测试	在第三个站点中，作为 **rh2adm** 运行，命令在 table.conversion 的末尾提供 在辅助节点上运行：**[root@clusternode1]# watch pcs status --full**

启动测试	执行集群命令： **[root@clusternode1]# pcs move resource SAPHana_RH2_02-clone <target-node>** **[root@clusternode1]# pcs resource clear SAPHana_RH2_02-clone**
预期结果	在 site 3 上的 monitor 命令中，主主从 clusternode1 变为 clusternode2。 清除资源后，同步状态将从 **SFAIL** 更改为 **SOK**。
返回初始状态的方法	运行测试两次。

(*)

```
remotehost3:rh2adm>
watch hdbnsutil -sr_state
[root@clusternode1]# tail -1000f /var/log/messages |egrep -e 'SOK|SWAIT|SFAIL'
```

详细描述

- **以 root 用户身份在 clusternode1 或 clusternode2 上检查集群的初始状态。**

```
[root@clusternode1]# pcs status --full
Cluster name: cluster1
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode1 (1) (version 2.1.2-4.el8_6.6-ada5c3b36e2) - partition with
quorum
  * Last updated: Mon Sep  4 06:34:46 2023
  * Last change:  Mon Sep  4 06:33:04 2023 by root via crm_attribute on clusternode1
  * 2 nodes configured
  * 6 resource instances configured

Node List:
  * Online: [ clusternode1 (1) clusternode2 (2) ]

Full List of Resources:
  * auto_rhevm_fence1 (stonith:fence_rhevm):  Started clusternode1
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * SAPHanaTopology_RH2_02 (ocf::heartbeat:SAPHanaTopology):  Started
clusternode2
    * SAPHanaTopology_RH2_02 (ocf::heartbeat:SAPHanaTopology):  Started
clusternode1
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * SAPHana_RH2_02 (ocf::heartbeat:SAPHana):  Slave clusternode2
```

```
    * SAPHana_RH2_02 (ocf::heartbeat:SAPHana):  Master clusternode1
  * vip_RH2_02_MASTER (ocf::heartbeat:IPaddr2):  Started clusternode1

Node Attributes:
  * Node: clusternode1 (1):
    * hana_rh2_clone_state           : PROMOTED
    * hana_rh2_op_mode               : logreplay
    * hana_rh2_remoteHost            : clusternode2
    * hana_rh2_roles                 : 4:P:master1:master:worker:master
    * hana_rh2_site                  : DC1
    * hana_rh2_sra                   : -
    * hana_rh2_srah                  : -
    * hana_rh2_srmode                : syncmem
    * hana_rh2_sync_state            : PRIM
    * hana_rh2_version               : 2.00.062.00
    * hana_rh2_vhost                 : clusternode1
    * lpa_rh2_lpt                    : 1693809184
    * master-SAPHana_RH2_02          : 150
  * Node: clusternode2 (2):
    * hana_rh2_clone_state           : DEMOTED
    * hana_rh2_op_mode               : logreplay
    * hana_rh2_remoteHost            : clusternode1
    * hana_rh2_roles                 : 4:S:master1:master:worker:master
    * hana_rh2_site                  : DC2
    * hana_rh2_sra                   : -
    * hana_rh2_srah                  : -
    * hana_rh2_srmode                : syncmem
    * hana_rh2_sync_state            : SOK
    * hana_rh2_version               : 2.00.062.00
    * hana_rh2_vhost                 : clusternode2
    * lpa_rh2_lpt                    : 30
    * master-SAPHana_RH2_02          : 100

Migration Summary:

Tickets:

PCSD Status:
  clusternode1: Online
  clusternode2: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

此输出显示，HANA 在 clusternode1 上被提升，它是主 SAP HANA 服务器，克隆资源的名称为 SAPHana_RH2_02-clone 是可升级的。
您可以在测试期间在单独的窗口中运行它来查看更改：

```
[root@clusternode1]# watch pcs status --full
```

- 识别 SAP HANA 克隆资源的另一种方法是：

```
[root@clusternode2]# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * Started: [ clusternode1 clusternode2 ]
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * Promoted: [ clusternode2 ]
    * Unpromoted: [ clusternode1 ]
```

- 要在开始测试前，在 remotehost3 上查看主服务器在 remotehost3 上启动监控的更改。

```
remotehost3:rh2adm> watch 'hdbnsutil -sr_state | grep "primary masters"
```

输出类似如下：

```
Every 2.0s: hdbnsutil -sr_state | grep "primary masters"
remotehost3: Mon Sep  4 08:47:21 2023

primary masters: clusternode1
```

在测试过程中，预期的输出将更改为 clusternode2。

- 通过将上面发现的克隆资源移到 clusternode2 来启动测试：

```
[root@clusternode1]# pcs resource move SAPhana_RH2_02-clone clusternode2
```

- remotehost3 上的 monitor 的输出将更改为：

```
Every 2.0s: hdbnsutil -sr_state | grep "primary masters"
remotehost3: Mon Sep  4 08:50:31 2023

primary masters: clusternode2
```

Pacemaker 为移动克隆资源创建一个位置约束。这需要手动删除。您可以使用以下方法查看约束：

```
[root@clusternode1]# pcs constraint location
```

需要删除此约束。

- 清除克隆资源以删除位置约束：

```
[root@clusternode1]#  pcs resource clear SAPhana_RH2_02-clone
Removing constraint: cli-prefer-SAPHana_RH2_02-clone
```

- 清理资源：

```
[root@clusternode1]# pcs resource cleanup SAPHana_RH2_02-clone
Cleaned up SAPHana_RH2_02:0 on clusternode2
Cleaned up SAPHana_RH2_02:1 on clusternode1
Waiting for 1 reply from the controller
... got reply (done)
```

测试的结果

- remotehost3 上的"主 master"监控应该立即切换到新主节点。
- 如果您检查集群状态，则以前的次要会被提升，以前的主会被重新注册，并且 Clone_State 从 Promoted 变为 Undefined to WAITINGFORLPA to DEMOTED。
- 当 SAPHana monitor 在故障转移后第一次启动时，次要会将 sync_state 更改为 SFAIL。由于现有位置约束，资源需要被清除，并在次要的 sync_state 的短时间将再次更改为 SOK。
- 二级被提升。

要恢复初始状态，您只需运行下一个测试即可。完成测试后，运行 清理。

5.4. 测试 2：故障切换具有被动第三个站点的主节点

测试的主题	没有重新注册已停止的第三个站点。 即使第三个站点停机，故障转移也可以正常工作。

测试先决条件	• DC1 上的 SAP HANA，DC2 正在运行并在 DC3 上停止。 • 集群已启动并运行，且没有错误或警告。
测试步骤	使用 **pcs [root@clusternode1]# resource move** 命令移动 SAPHana 资源。
启动测试	执行集群命令 **：[root@clusterclusternode1]# pcs move resource SAPHana_RH2_02-clone clusterclusternode1**
预期结果	DC3 没有更改。SAP HANA 系统复制与旧关系保持同步。
返回初始状态的方法	在新主上重新注册 DC3，并启动 SAP HANA。

详细描述

- **以 root 用户身份在 clusternode1 或 clusternode2 上检查集群的初始状态：**

```
[root@clusternode1]# pcs status --full
Cluster name: cluster1
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode1 (1) (version 2.1.2-4.el8_6.6-ada5c3b36e2) - partition with
quorum
  * Last updated: Mon Sep  4 06:34:46 2023
  * Last change:  Mon Sep  4 06:33:04 2023 by root via crm_attribute on clusternode1
  * 2 nodes configured
  * 6 resource instances configured

Node List:
  * Online: [ clusternode1 (1) clusternode2 (2) ]

Full List of Resources:
  * auto_rhevm_fence1 (stonith:fence_rhevm):  Started clusternode1
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * SAPHanaTopology_RH2_02 (ocf::heartbeat:SAPHanaTopology):  Started
clusternode2
    * SAPHanaTopology_RH2_02 (ocf::heartbeat:SAPHanaTopology):  Started
clusternode1
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * SAPHana_RH2_02 (ocf::heartbeat:SAPHana):  Slave clusternode2
    * SAPHana_RH2_02 (ocf::heartbeat:SAPHana):  Master clusternode1
  * vip_RH2_02_MASTER (ocf::heartbeat:IPaddr2):  Started clusternode1

Node Attributes:
```

```
 * Node: clusternode1 (1):
   * hana_rh2_clone_state          : PROMOTED
   * hana_rh2_op_mode              : logreplay
   * hana_rh2_remoteHost           : clusternode2
   * hana_rh2_roles                : 4:P:master1:master:worker:master
   * hana_rh2_site                 : DC1
   * hana_rh2_sra                  : -
   * hana_rh2_srah                 : -
   * hana_rh2_srmode               : syncmem
   * hana_rh2_sync_state           : PRIM
   * hana_rh2_version              : 2.00.062.00
   * hana_rh2_vhost                : clusternode1
   * lpa_rh2_lpt                   : 1693809184
   * master-SAPHana_RH2_02         : 150
 * Node: clusternode2 (2):
   * hana_rh2_clone_state          : DEMOTED
   * hana_rh2_op_mode              : logreplay
   * hana_rh2_remoteHost           : clusternode1
   * hana_rh2_roles                : 4:S:master1:master:worker:master
    * hana_rh2_site                : DC2
   * hana_rh2_sra                  : -
   * hana_rh2_srah                 : -
   * hana_rh2_srmode               : syncmem
   * hana_rh2_sync_state           : SOK
   * hana_rh2_version              : 2.00.062.00
   * hana_rh2_vhost                : clusternode2
   * lpa_rh2_lpt                   : 30
   * master-SAPHana_RH2_02         : 100

Migration Summary:

Tickets:

PCSD Status:
  clusternode1: Online
  clusternode2: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

本例的输出显示，HANA 在 clusternode1 上被提升，它是主 SAP HANA 服务器，克隆资源的名称为 SAPHana_RH2_02-clone 是可升级的。如果您在 HANA 之前运行 test 3，则可能会在 clusternode2 上提升。

- 停止 remotehost3 上的数据库：

```
remotehost3:rh2adm> HDB stop
hdbdaemon will wait maximal 300 seconds for NewDB services finishing.
Stopping instance using: /usr/sap/RH2/SYS/exe/hdb/sapcontrol -prot NI_HTTP -nr 02 -
function Stop 400
```

```
12.07.2023 11:33:14
Stop
OK
Waiting for stopped instance using: /usr/sap/RH2/SYS/exe/hdb/sapcontrol -prot
NI_HTTP -nr 02 -function WaitforStopped 600 2


12.07.2023 11:33:30
WaitforStopped
OK
hdbdaemon is stopped.
```

- 检查 remotehost3 上的主数据库：

```
remotehost3:rh2adm> hdbnsutil -sr_stateConfiguration| grep -i "primary masters"

primary masters: clusterclusternode2
```

- 检查集群节点上集群中的当前主要信息：

```
[root@clusterclusternode1]# pcs resource | grep Masters
   * Masters: [ clusternode2 ]
```

- 检查 sr_state 以查看 SAP HANA 系统复制关系：

```
clusternode2:rh2adm> hdbnsutil -sr_state

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 2
site name: DC1

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false
is primary suspended: false

Host Mappings:
~~~~~~~~~~~~~~

clusternode1 -> [DC3] remotehost3
clusternode1 -> [DC1] clusternode1
```

```
clusternode1 -> [DC2] clusternode2


Site Mappings:
~~~~~~~~~~~~~~~
DC1 (primary/primary)
    |---DC3 (syncmem/logreplay)
    |---DC2 (syncmem/logreplay)

Tier of DC1: 1
Tier of DC3: 2
Tier of DC2: 2

Replication mode of DC1: primary
Replication mode of DC3: syncmem
Replication mode of DC2: syncmem

Operation mode of DC1: primary
Operation mode of DC3: logreplay
Operation mode of DC2: logreplay

Mapping: DC1 -> DC3
Mapping: DC1 -> DC2
done.
```

SAP HANA 系统复制关系仍然有一个主要(DC1)，它被复制到 DC2 和 DC3。remotehost3 （关闭）上的复制关系可以使用以下方法显示：

```
remothost3:rh2adm> hdbnsutil -sr_stateConfiguration

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

mode: syncmem
site id: 3
site name: DC3
active primary site: 1

primary masters: clusternode1
done.
```

remotehost3 上的数据库，其离线检查 global.ini 文件中的条目。

- 启动测试：在集群中启动故障转移，移动 SAPHana-clone-resource 示例：

```
[root@clusternode1]# pcs resource move SAPHana_RH2_02-clone clusternode2
```

注意

如果在 clusternode2 上提升 SAPHana，则必须将克隆资源移到 clusternode1。该示例要求 SAPHana 在 clusternode1 上提升。

没有输出。与之前的测试类似，会创建一个位置约束，该约束可以使用以下方法显示：

```
[root@clusternode1]# pcs constraint location
Location Constraints:
  Resource: SAPHana_RH2_02-clone
    Enabled on:
      Node: clusternode1 (score:INFINITY) (role:Started)
```

即使集群再次查找正常，此约束也会避免另一个故障转移，除非删除了约束。其中一种方法是清除资源。

- 清除资源：

```
[root@clusternode1]# pcs constraint location
Location Constraints:
  Resource: SAPHana_RH2_02-clone
    Enabled on:
      Node: clusternode1 (score:INFINITY) (role:Started)
[root@clusternode1]# pcs resource clear SAPHana_RH2_02-clone
Removing constraint: cli-prefer-SAPHana_RH2_02-clone
```

- 清理资源：

```
[root@clusternode1]# pcs resource cleanup SAPHana_RH2_02-clone
Cleaned up SAPHana_RH2_02:0 on clusternode2
Cleaned up SAPHana_RH2_02:1 on clusternode1
Waiting for 1 reply from the controller
... got reply (done)
```

- 检查当前状态。可以通过两种方式显示需要同步的复制状态。从 remotehost3 上的主设备开始：

```
remotehost3:rh2adm>  hdbnsutil -sr_stateConfiguration| grep -i primary
active primary site: 1
primary masters: clusternode1
```

输出显示站点 1 或 clusternode1，这是开始测试之前的主要内容，将主要移到 clusternode2。然后再检查新主上的系统复制状态。首先检测新主设备：

```
[root@clusternode1]# pcs resource | grep  Master
    * Masters: [ clusternode2 ]
```

在这里，我们有一个不一致的，需要重新注册 remotehost3。您可能认为，如果我们再次运行测试，我们可能会将主设备切回到原始 clusternode1。在这种情况下，我们有第三个方法来识别系统复制是否正常工作。在主节点上，我们的问题单 clusternode2 运行：

```
clusternode2:rh2adm> cdpy
clusternode2:rh2adm> python
$DIR_EXECUTABLE/python_support/systemReplicationStatus.py
|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary
|Secondary |Secondary |Secondary |Secondary     |Replication |Replication |Replication
|Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active
Status |Mode        |Status      |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |---
--------- |----------- |----------- |-------------- |------------ |
|SYSTEMDB |clusternode2 |30201 |nameserver   |       1 |      2 |DC2       |clusternode1    |
30201 |        1 |DC1       |YES           |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode2 |30207 |xsengine     |       2 |      2 |DC2       |clusternode1    |    30207 |
1 |DC1       |YES           |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode2 |30203 |indexserver  |       3 |      2 |DC2       |clusternode1    |    30203
|        1 |DC1       |YES           |SYNCMEM     |ACTIVE      |             |        True |

status system replication site "1": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 2
site name: DC2
```

如果在此输出中没有看到 remotehost3，则必须重新注册 remotehost3。在注册前，请在主节点上运行以下内容以观察注册的进度：

```
clusternode2:rh2adm> watch python
$DIR_EXECUTABLE/python_support/systemReplicationStatus.py
```

现在，您可以使用以下命令重新注册 remotehost3：

```
remotehost3:rh2adm> hdbnsutil -sr_register --remoteHost=clusternode2 --
remoteInstance=${TINSTANCE} --replicationMode=async --name=DC3 --
remoteName=DC2 --operation
```

```
Mode=logreplay --online
adding site ...
collecting information ...
updating local ini files ...
done.
```

即使 remotehost3 上的数据库尚未启动，您也无法在系统复制状态输出中看到第三个站点。通过在 remotehost3 上启动数据库，可以完成注册：

```
remotehost3:rh2adm> HDB start


StartService
Impromptu CCC initialization by 'rscpCInit'.
  See SAP note 1266393.
OK
OK
Starting instance using: /usr/sap/RH2/SYS/exe/hdb/sapcontrol -prot NI_HTTP -nr 02 -
function StartWait 2700 2


04.09.2023 11:36:47
Start
OK
```

上面启动的监控器将立即显示 remotehost3 的同步。

- 要切换回来，请再次运行测试。一个可选测试是将主要测试切换到节点，该节点在 remotehost3 上的 global.ini 上配置，然后启动数据库。数据库可能会启动，但永远不会显示在系统复制状态的输出中，除非它被重新注册。

如需更多信息，请参阅 检查 SAP HANA 系统复制状态。

5.5. 测试 3：将主站点故障切换到第三个站点

测试的主题	将主站点故障转移到第三个站点。 辅助将重新注册到第三个站点。

测试先决条件	• DC1、DC2、DC3 上的 SAP HANA 正在运行。 • 集群已启动并运行，且没有错误或警告。 • 系统复制已就位并处于同步状态（检查 **python systemReplicationStatus.py**）。
测试步骤	将集群设置为 **maintenance-mode** 以便能够恢复。 使用 **hdbnsutil -sr_takeover**组成第三个节点
启动测试	执行 SAP HANA 命令on remotehost3: **rh2adm>: hdbnsutil -sr_takeover**
监控测试	在第三个站点 run: **remotehost3:rh2adm> watch hdbnsutil -sr_state**
预期结果	• 第三个节点将成为主要节点。 • 辅助节点将主 master 更改为 remotehost3。以前的主节点需要重新注册到新主节点。
返回初始状态的方法	运行 Test 4：将主节点故障转移到第一个站点。

详细描述

- **检查数据库是否使用 Check 数据库 运行，并检查复制状态：**

```
clusternode2:rh2adm> hdbnsutil -sr_state | egrep -e "^mode:|primary masters"
```

例如，输出为：

```
mode: syncmem
primary masters: clusternode1
```

在这种情况下，主数据库是 clusternode1。如果在 clusternode1 上运行这个命令，您将获得：

```
mode: primary
```

在这个主节点上，您还可以显示系统复制状态。它应该类似如下：

```
clusternode1:rh2adm> cdpy
clusternode1:rh2adm> python systemReplicationStatus.py
|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replication    |Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |--------- |-------------
|----------- |----------- |-------------- |------------ |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |remotehost3    |    30201 |
3 |DC3       |YES          |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |remotehost3    |    30207 |       3
|DC3       |YES          |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |remotehost3    |    30203 |       3
|DC3       |YES          |SYNCMEM     |ACTIVE      |             |        True |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |clusternode2    |    30201 |
2 |DC2       |YES          |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |clusternode2    |    30207 |       2
|DC2       |YES          |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |clusternode2    |    30203 |       2
|DC2       |YES          |SYNCMEM     |ACTIVE      |             |        True |

status system replication site "3": ACTIVE
status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC1
```

- 现在，我们有一个适当的环境，我们可以开始监控所有 3 个节点上的系统复制状态。

在测试启动前，应启动 3 个监视器。执行测试时，输出将改变。因此，只要测试还没有完成，使它们保持运行。

在旧的主节点上，clusternode1 在测试过程中在一个单独的窗口中运行：

```
clusternode1:rh2adm> watch -n 5 'python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatus.py ; echo Status $?`
```

clusternode1 上的输出将是：

```
Every 5.0s: python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicati...
clusternode1: Tue XXX XX HH:MM:SS 2023

|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |
Replication |Replication |Replication    |Secondary    |
|        |  | |            |         |       |         |Host  |Port      |Site ID   |Site Name |Active Status |
Mode        |Status  |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |-------------
|
---------- |----------- |-------------- |------------ |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |  1 |DC1       |remotehost3    |    30201 |
3 |DC3   |YES   |
ASYNC       |ACTIVE  |             |       True |
|RH2   |clusternode1 |30207 |xsengine     |       2 |  1 |DC1       |remotehost3    |    30207 |       3 |DC3
|YES   |
ASYNC       |ACTIVE  |             |       True |
|RH2   |clusternode1 |30203 |indexserver  |       3 |  1 |DC1       |remotehost3    |    30203 |       3 |DC3
|YES   |
ASYNC       |ACTIVE  |             |       True |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |  1 |DC1       |clusternode2    |    30201 |
2 |DC2   |YES   |
SYNCMEM     |ACTIVE  |             |       True |
|RH2   |clusternode1 |30207 |xsengine     |       2 |  1 |DC1       |clusternode2    |    30207 |       2 |DC2
|YES   |
SYNCMEM     |ACTIVE  |             |       True |
|RH2   |clusternode1 |30203 |indexserver  |       3 |  1 |DC1       |clusternode2    |    30203 |       2 |DC2
|YES   |
SYNCMEM     |ACTIVE  |             |       True |

status system replication site "3": ACTIVE
status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC1
Status 15
```

在 remotehost3 上运行相同的命令：

```
remotehost3:rh2adm> watch -n 5 'python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py ; echo Status $?'
```

响应将是：

```
this system is either not running or not primary system replication site
```

测试启动故障转移后，输出将改变。在测试启动前，输出类似于主节点的示例。

在第二个节点上启动：

```
clusternode2:rh2adm> watch -n 10 'hdbnsutil -sr_state | grep masters'
```

这将显示当前 master clusternode1，并在启动故障转移后立即切换。

- 为确保一切配置正确，还要检查 global.ini。
- 在 DC1、DC2 和 DC3 上检查 global.ini。

在所有三个节点上，global.ini 应该包含：

```
[persistent]
log_mode=normal
[system_replication]
register_secondaries_on_takeover=true
```

您可以使用以下内容编辑 global.ini：

```
clusternode1:rh2adm>vim
/usr/sap/$SAPSYSTEMNAME/SYS/global/hdb/custom/config/global.ini
```

- [可选] 将集群置于 maintenance-mode ：

  ```
  [root@clusternode1]# pcs property set maintenance-mode=true
  ```

  在测试过程中，您会发现故障转移将使用 和，而不设置 maintenance-mode。因此，您可以在不使用它的情况下运行第一个测试。在恢复它时，我只想向您显示，而是使用。这是主要方面

的一个选项，不能访问。

- 开始测试：故障切换到 DC3。在 remotehost3 上，请运行：

```
remotehost3:rh2adm> hdbnsutil -sr_takeover
done.
```

测试已启动，现在请检查之前启动的 monitor 的输出。

在 clusternode1 上，系统复制状态将丢失与 remotehost3 和 clusternode2 (DC2)的关系：

```
Every 5.0s: python /usr/sap/RH2/HDB02/exe/python_support/systemReplicationStatus.py ; echo
Status $?                          clusternode1: Mon Sep  4 11:52:16 2023

|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replic
ation                  |Secondary    |
|        |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status
 Details               |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |--------- |-------------
|----------- |----------- |------
---------------------- |------------ |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |clusternode2    |    30201 |
2 |DC2       |YES           |SYNCMEM     |ERROR       |Commun
ication channel closed |       False |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |clusternode2    |    30207 |       2
|DC2       |YES           |SYNCMEM     |ERROR       |Commun
ication channel closed |       False |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |clusternode2    |    30203 |       2
|DC2       |YES           |SYNCMEM     |ERROR       |Commun
ication channel closed |       False |

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

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC1
Status 11
```

集群仍然不会注意到此行为。如果您检查系统复制状态的返回码，则返回码 11 表示错误，这告诉您出现问题。如果您有访问权限，现在最好进入 maintenance-mode。

remotehost3 变为新的主主，而 clusternode2 (DC2)会在 remotehost3 上自动注册为新主。

remotehost3 的系统复制状态的输出示例：

```
Every 5.0s: python /usr/sap/RH2/HDB02/exe/python_support/systemReplicationStatus.py ; echo
Status $?                          remotehost3: Mon Sep  4 13:55:29 2023

|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replic
ation    |Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status
 Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |--------- |-------------
|----------- |----------- |------
-------- |------------ |
|SYSTEMDB |remotehost3 |30201 |nameserver   |        1 |      3 |DC3       |clusternode2    |    30201 |
2 |DC2       |YES           |SYNCMEM     |ACTIVE      |
         |        True |
|RH2      |remotehost3 |30207 |xsengine     |        2 |      3 |DC3       |clusternode2    |    30207 |        2
|DC2       |YES           |SYNCMEM     |ACTIVE      |
         |        True |
|RH2      |remotehost3 |30203 |indexserver  |        3 |      3 |DC3       |clusternode2    |    30203 |        2
|DC2       |YES           |SYNCMEM     |ACTIVE      |
         |        True |

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

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 3
site name: DC3
Status 15
```

returncode 15 还表示一切都是 okay，但缺少 clusternode1。这必须手动重新注册。前一个主 clusternode1 没有被列出，因此复制关系会丢失。

- 设置 **maintenance-mode**。

如果还没有在集群的一个节点上设置 maintenance-mode 之前完成，使用以下命令：

```
[root@clusternode1]# pcs property  set maintenance-mode=true
```

您可以运行以下命令来检查 maintenance-mode 是否活跃：

```
[root@clusternode1]# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02] (unmanaged):
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode2 (unmanaged)
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode1 (unmanaged)
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable, unmanaged):
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Slave clusternode2 (unmanaged)
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Master clusternode1 (unmanaged)
  * vip_RH2_02_MASTER   (ocf::heartbeat:IPaddr2):       Started clusternode1 (unmanaged)
```

资源显示非受管状态，这表示集群处于 maintenance-mode=true。虚拟 IP 地址仍然在 clusternode1 上启动。如果要在另一个节点上使用此 IP，请在设置 maintanence-mode=true 前禁用 vip_RH2_02_MASTER。

```
[root@clusternode1]# pcs resource disable vip_RH2_02_MASTER
```

- 当我们检查 clusternode1 上的 sr_state 时，您将只看到一个与 DC2 的关系：

```
clusternode1:rh2adm> hdbnsutil -sr_state

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 1
site name: DC1

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false
is primary suspended: false

Host Mappings:
~~~~~~~~~~~~~~

clusternode1 -> [DC2] clusternode2
clusternode1 -> [DC1] clusternode1


Site Mappings:
~~~~~~~~~~~~~~
DC1 (primary/primary)
```

```
    |---DC2 (syncmem/logreplay)

Tier of DC1: 1
Tier of DC2: 2

Replication mode of DC1: primary
Replication mode of DC2: syncmem

Operation mode of DC1: primary
Operation mode of DC2: logreplay

Mapping: DC1 -> DC2
done.
```

但是，当检查 DC2 时，主数据库服务器为 DC3。因此，DC1 的信息不正确。

```
clusternode2:rh2adm> hdbnsutil -sr_state
```

如果我们在 DC1 上检查系统复制状态，则返回码为 12 （未知）。因此，需要重新注册 DC1。

您可以使用此命令将以前的主 clusternode1 注册为 remotehost3 的新辅助设备：

```
clusternode1:rh2adm> hdbnsutil -sr_register --remoteHost=remotehost3 --
remoteInstance=${TINSTANCE} --replicationMode=async --name=DC1 --remoteName=DC3 --
operationMode=logreplay --online
```

注册完成后，您将在 remotehost3 上看到复制三个站点，状态（重新代码）将变为 15。如果此操作失败，则必须手动删除 DC1 和 DC3 上的复制关系。请按照 Register Secondary 中描述的说明进行操作。例如，列出现有关系：

```
hdbnsutil -sr_state
```

删除您可以使用的现有关系示例：

```
clusternode1:rh2adm> hdbnsutil -sr_unregister --name=DC2
```

这通常不需要这样做。

我们假定测试 4 将在测试 3 后执行。因此，恢复步骤是运行测试 4。

5.6. 测试 4：将主节点故障转移到第一个站点

测试的主题	主切换回一个集群节点。 故障恢复并再次启用集群。 将第三个站点重新注册为次要站点。
测试先决条件	• SAP HANA 主节点在第三个站点上运行。 • 集群部分正在运行。 • 集群 **进入维护模式**。 • 前一个集群主要可以访问。
测试步骤	检查集群的预期主要信息。 从 DC3 节点故障转移到 DC1 节点。 检查前期次要是否已切换到新主设备。 重新注册 remotehost3 作为新的次要。 设置 cluster **maintenance-mode=false**，集群将继续工作。
监控测试	在新的主启动时： **remotehost3:rh2adm> watch python $DIR_EXECUTABLE/python_support/systemReplicationStatus.py [root@clusternode1]# watch pcs status --full On the secondary start: clusternode:rh2adm> watch hdbnsutil -sr_state**

启动测试	检查集群的预期主要内容：**[root@clusternode1]# pcs resource** VIP 和提升 SAP HANA 资源应该在同一节点上运行，而这是潜在的新主要资源。 在这个潜在的主要运行中： **clusternode1:rh2adm> hdbnsutil -sr_takeover** 重新将前一个主重新注册为新次要： **clusternode1:rh2adm> hdbnsutil -sr_register \ --remoteHost=clusternode1 \ --remoteInstance=${TINSTANCE} \ --replicationMode=syncmem \ --name=DC3 \ --remoteName=DC1 \ --operationMode=logreplay \ --force_full_replica \ --online** 在设置 **maintenance-mode=false** 后集群将继续工作。
预期结果	新主要是启动 SAP HANA。 复制状态将显示所有 3 个站点复制。 第二个集群站点会自动重新注册到新主站点。 灾难恢复(DR)站点成为数据库的额外副本。
返回初始状态的方法	运行测试 3。

详细描述

- **检查集群是否已设置为 maintenance-mode ：**

```
[root@clusternode1]# pcs property config maintenance-mode
Cluster Properties:
 maintenance-mode: true
```

如果 maintenance-mode 不是 true，您可以使用以下方法对其进行设置：

```
[root@clusternode1]# pcs property set  maintenance-mode=true
```

- **检查系统复制状态，并发现所有节点上的主数据库。首先使用以下命令发现主数据库：**

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "^mode:|primary masters"
```

其输出应如下所示：

在 clusternode1 上：

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "^mode:|primary masters"
mode: syncmem
primary masters: remotehost3
```

在 clusternode2 上：

```
clusternode2:rh2adm> hdbnsutil -sr_state | egrep -e "^mode:|primary masters"
mode: syncmem
primary masters: remotehost3
```

在 remotehost3:

```
remotehost3:rh2adm> hdbnsutil -sr_state | egrep -e "^mode:|primary masters"
mode: primary
```

在所有三个节点上，主数据库为 remotehost3。在这个主数据库中，您必须确保所有三个节点的系统复制状态处于活跃状态，返回码为 15：

```
remotehost3:rh2adm> python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatus.py
|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replication    |Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |---------- |
|----------- |----------- |-------------- |------------ |
|SYSTEMDB |remotehost3 |30201 |nameserver   |       1 |     3 |DC3       |clusternode2    |    30201 |
2 |DC2       |YES           |SYNCMEM     |ACTIVE      |            |        True |
|RH2      |remotehost3 |30207 |xsengine     |       2 |     3 |DC3       |clusternode2    |    30207 |       2
|DC2       |YES           |SYNCMEM     |ACTIVE      |            |        True |
|RH2      |remotehost3 |30203 |indexserver  |       3 |     3 |DC3       |clusternode2    |    30203 |       2
|DC2       |YES           |SYNCMEM     |ACTIVE      |            |        True |
|SYSTEMDB |remotehost3 |30201 |nameserver   |       1 |     3 |DC3       |clusternode1    |    30201 |
1 |DC1       |YES           |SYNCMEM     |ACTIVE      |            |        True |
|RH2      |remotehost3 |30207 |xsengine     |       2 |     3 |DC3       |clusternode1    |    30207 |       1
|DC1       |YES           |SYNCMEM     |ACTIVE      |            |        True |
|RH2      |remotehost3 |30203 |indexserver  |       3 |     3 |DC3       |clusternode1    |    30203 |       1
|DC1       |YES           |SYNCMEM     |ACTIVE      |            |        True |
```

```
status system replication site "2": ACTIVE
status system replication site "1": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 3
site name: DC3
[rh2adm@remotehost3: python_support]# echo $?
15
```

- 检查所有三个 **sr_states** 都一致。

请在所有三个节点上运行 **hdbnsutil -sr_state --sapcontrol=1 |grep site prerequisitesMode:**

```
clusternode1:rh2adm>hdbnsutil -sr_state --sapcontrol=1 |grep  site.*Mode
```

```
clusternode2:rh2adm>hsbnsutil -sr_state --sapcontrol=1 | grep site.*Mode
```

```
remotehost3:rh2adm>hsbnsutil -sr_state --sapcontrol=1 | grep site.*Mode
```

所有节点上的输出应该相同：

```
siteReplicationMode/DC1=primary
siteReplicationMode/DC3=async
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC3=logreplay
siteOperationMode/DC2=logreplay
```

- 在单独的窗口中启动监控。

在 **clusternode1** 启动时：

```
clusternode1:rh2adm> watch "python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py; echo \$?"
```

在 **remotehost3** 启动时：

```
remotehost3:rh2adm> watch "python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py; echo \$?"
```

在 clusternode2 启动时：

```
clusternode2:rh2adm> watch "hdbnsutil -sr_state --sapcontrol=1 |grep  siteReplicationMode"
```

- 启动测试

在 clusternode1 上切换到 clusternode1 启动：

```
clusternode1:rh2adm> hdbnsutil -sr_takeover
done.
```

- 检查 monitor 的输出。

clusternode1 上的监控器将更改为：

```
Every 2.0s: python systemReplicationStatus.py; echo $?
clusternode1: Mon Sep  4 23:34:30 2023

|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replication    |Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |--------- |-------------
|----------- |----------- |-------------- |------------ |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |clusternode2    |    30201 |
2 |DC2       |YES           |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |clusternode2    |    30207 |       2
|DC2       |YES           |SYNCMEM     |ACTIVE      |             |        True |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |clusternode2    |    30203 |       2
|DC2       |YES           |SYNCMEM     |ACTIVE      |             |        True |

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

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC1
15
```

-

重要信息也是返回代码 15。clusternode2 上的监控器将更改为：

```
Every 2.0s: hdbnsutil -sr_state --sapcontrol=1 |grep  site.*Mode
clusternode2: Mon Sep  4 23:35:18 2023

siteReplicationMode/DC1=primary
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC2=logreplay
```

DC3 已消失，需要重新注册。在 remotehost3 上，systemReplicationStatus 会报告一个错误，并将 returncode 变为 11。

- 检查集群节点是否已重新注册。

```
clusternode1:rh2adm>  hdbnsutil -sr_state

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: primary
operation mode: primary
site id: 1
site name: DC1

is source system: true
is secondary/consumer system: false
has secondaries/consumers attached: true
is a takeover active: false
is primary suspended: false

Host Mappings:
~~~~~~~~~~~~~~

clusternode1 -> [DC2] clusternode2
clusternode1 -> [DC1] clusternode1


Site Mappings:
~~~~~~~~~~~~~~
DC1 (primary/primary)
    |---DC2 (syncmem/logreplay)

Tier of DC1: 1
Tier of DC2: 2

Replication mode of DC1: primary
```

```
Replication mode of DC2: syncmem

Operation mode of DC1: primary
Operation mode of DC2: logreplay

Mapping: DC1 -> DC2
done.
```

站点映射显示，clusternode2 (DC2)被重新注册。

- 检查或启用 vip 资源：

```
[root@clusternode1]# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]
(unmanaged):
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode2 (unmanaged)
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode1
   (unmanaged)
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable, unmanaged):
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Master clusternode2
(unmanaged)
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Slave clusternode1
   (unmanaged)
  * vip_RH2_02_MASTER   (ocf::heartbeat:IPaddr2):       Stopped (disabled,
unmanaged)
```

vip 资源 vip_RH2_02_MASTER 已停止。要再次运行它：

```
[root@clusternode1]# pcs resource enable vip_RH2_02_MASTER
Warning: 'vip_RH2_02_MASTER' is unmanaged
```

警告正确，因为集群不会启动任何资源，除非 maintenance-mode=false。

- 停止集群 维护模式。

在我们停止 maintenance-mode 之前，我们应在单独的窗口中启动两个 monitor 以查看更改。在 clusternode2 上运行：

```
[root@clusternode2]# watch pcs status --full
```

在 clusternode1 上运行：

```
clusternode1:rh2adm> watch "python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py; echo $?"
```

现在，您可以运行以下命令，在 clusternode1 上取消设置 maintenance-mode ：

```
[root@clusternode1]# pcs property set maintenance-mode=false
```

clusternode2 上的 monitor 应该显示一切现在都如预期运行：

```
Every 2.0s: pcs status --full
clusternode1: Tue Sep  5 00:01:17 2023

Cluster name: cluster1
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode1
 (1) (version 2.1.2-4.el8_6.6-ada5c3b36e2) - partition with quorum
  * Last updated: Tue Sep  5 00:01:17 2023
  * Last change:  Tue Sep  5 00:00:30 2023 by root via crm_attribute on clusternode1

  * 2 nodes configured
  * 6 resource instances configured

Node List:
  * Online: [ clusternode1
 (1) clusternode2 (2) ]

Full List of Resources:
  * auto_rhevm_fence1   (stonith:fence_rhevm):   Started clusternode1

  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode2
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode1

  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Slave clusternode2
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Master clusternode1

  * vip_RH2_02_MASTER   (ocf::heartbeat:IPaddr2):       Started clusternode1


Node Attributes:
  * Node: clusternode1
 (1):
```

```
    * hana_rh2_clone_state      : PROMOTED
    * hana_rh2_op_mode          : logreplay
    * hana_rh2_remoteHost       : clusternode2
    * hana_rh2_roles            : 4:P:master1:master:worker:master
    * hana_rh2_site             : DC1
    * hana_rh2_sra              : -
    * hana_rh2_srah             : -
    * hana_rh2_srmode           : syncmem
    * hana_rh2_sync_state       : PRIM
    * hana_rh2_version          : 2.00.062.00
    * hana_rh2_vhost            : clusternode1

    * lpa_rh2_lpt               : 1693872030
    * master-SAPHana_RH2_02     : 150
  * Node: clusternode2 (2):
    * hana_rh2_clone_state      : DEMOTED
    * hana_rh2_op_mode          : logreplay
    * hana_rh2_remoteHost       : clusternode1

    * hana_rh2_roles            : 4:S:master1:master:worker:master
    * hana_rh2_site             : DC2
    * hana_rh2_sra              : -
    * hana_rh2_srah             : -
    * hana_rh2_srmode           : syncmem
    * hana_rh2_sync_state       : SOK
    * hana_rh2_version          : 2.00.062.00
    * hana_rh2_vhost            : clusternode2
    * lpa_rh2_lpt               : 30
    * master-SAPHana_RH2_02     : 100

Migration Summary:

Tickets:

PCSD Status:
  clusternode1
: Online
  clusternode2: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

手动交互后，最好清理集群，如 Cluster Cleanup 所述。

- 将 remotehost3 重新注册到 clusternode1 上的新主卷。

需要重新注册 Remotehost3。要监控进度，请在 clusternode1 上启动：

```
clusternode1:rh2adm> watch -n 5 'python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py ; echo Status $?'
```

在 remotehost3 上，请开始：

```
remotehost3:rh2adm> watch 'hdbnsutil -sr_state --sapcontrol=1 |grep  siteReplicationMode'
```

现在，您可以使用这个命令重新注册 remotehost3：

```
remotehost3:rh2adm> hdbnsutil -sr_register --remoteHost=clusternode1 --
remoteInstance=${TINSTANCE} --replicationMode=async --name=DC3 --remoteName=DC1 --
operationMode=logreplay --online
```

clusternode1 上的监控器将更改为：

```
Every 5.0s: python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatus.py ;
echo Status $?                                                    clusternode1: Tue Sep  5
00:14:40 2023

|Database |Host   |Port  |Service Name |Volume ID |Site ID |Site Name |Secondary |Secondary
|Secondary |Secondary |Secondary     |Replication |Replication |Replication    |Secondary    |
|         |       |      |             |          |        |          |Host      |Port      |Site ID   |Site Name |Active Status
|Mode        |Status      |Status Details |Fully Synced |
|-------- |------ |----- |------------ |--------- |------- |--------- |--------- |--------- |--------- |--------- |-------------
|----------- |----------- |-------------- |------------ |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |remotehost3    |    30201 |
3 |DC3       |YES          |ASYNC     |ACTIVE      |             |       True |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |remotehost3    |    30207 |       3
|DC3       |YES          |ASYNC     |ACTIVE      |             |       True |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |remotehost3    |    30203 |       3
|DC3       |YES          |ASYNC     |ACTIVE      |             |       True |
|SYSTEMDB |clusternode1 |30201 |nameserver   |       1 |      1 |DC1       |clusternode2    |    30201 |
2 |DC2       |YES          |SYNCMEM     |ACTIVE      |             |       True |
|RH2      |clusternode1 |30207 |xsengine     |       2 |      1 |DC1       |clusternode2    |    30207 |       2
|DC2       |YES          |SYNCMEM     |ACTIVE      |             |       True |
|RH2      |clusternode1 |30203 |indexserver  |       3 |      1 |DC1       |clusternode2    |    30203 |       2
|DC2       |YES          |SYNCMEM     |ACTIVE      |             |       True |

status system replication site "3": ACTIVE
status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
```

```
site id: 1
site name: DC1
Status 15
```

remotehost3 的监控器将更改为：

```
Every 2.0s: hdbnsutil -sr_state --sapcontrol=1 |grep  site.*Mode
remotehost3: Tue Sep  5 02:15:28 2023

siteReplicationMode/DC1=primary
siteReplicationMode/DC3=syncmem
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC3=logreplay
siteOperationMode/DC2=logreplay
```

现在，我们再次有 3 个条目，remotehost3 (DC3)再次是从 clusternode1 (DC1)复制的次要站点。

- **检查所有节点是否是 clusternode1 上的系统复制状态的一部分。**

请在所有三个节点上运行 hdbnsutil -sr_state --sapcontrol=1 |grep site prerequisitesMode:

```
clusternode1:rh2adm> hdbnsutil -sr_state --sapcontrol=1 |grep
site.*ModesiteReplicationMode
```

```
clusternode2:rh2adm> hsbnsutil -sr_state --sapcontrol=1 | grep site.*Mode
```

```
remotehost3:rh2adm> hsbnsutil -sr_state --sapcontrol=1 | grep site.*Mode
```

在所有节点上，我们应该获得相同的输出：

```
siteReplicationMode/DC1=primary
siteReplicationMode/DC3=syncmem
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC3=logreplay
siteOperationMode/DC2=logreplay
```

- **检查 pcs status --full 和 SOK。运行：**

```
[root@clusternode1]# pcs status --full| grep sync_state
```

输出应该是 PRIM 或 SOK ：

```
 * hana_rh2_sync_state            : PRIM
   * hana_rh2_sync_state            : SOK
```

最后，集群状态应如下所示，包括 sync_state PRIM 和 SOK ：

```
[root@clusternode1]# pcs status --full
Cluster name: cluster1
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode1
 (1) (version 2.1.2-4.el8_6.6-ada5c3b36e2) - partition with quorum
  * Last updated: Tue Sep  5 00:18:52 2023
  * Last change:  Tue Sep  5 00:16:54 2023 by root via crm_attribute on clusternode1

  * 2 nodes configured
  * 6 resource instances configured

Node List:
  * Online: [ clusternode1
 (1) clusternode2 (2) ]

Full List of Resources:
  * auto_rhevm_fence1   (stonith:fence_rhevm):   Started clusternode1

  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode2
    * SAPHanaTopology_RH2_02    (ocf::heartbeat:SAPHanaTopology):       Started
clusternode1

  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Slave clusternode2
    * SAPHana_RH2_02    (ocf::heartbeat:SAPHana):       Master clusternode1

  * vip_RH2_02_MASTER   (ocf::heartbeat:IPaddr2):       Started clusternode1


Node Attributes:
  * Node: clusternode1
 (1):
    * hana_rh2_clone_state            : PROMOTED
    * hana_rh2_op_mode                : logreplay
    * hana_rh2_remoteHost             : clusternode2
    * hana_rh2_roles                  : 4:P:master1:master:worker:master
    * hana_rh2_site                   : DC1
    * hana_rh2_sra                    : -
    * hana_rh2_srah                   : -
    * hana_rh2_srmode                 : syncmem
    * hana_rh2_sync_state             : PRIM
```

```
    * hana_rh2_version              : 2.00.062.00
    * hana_rh2_vhost                : clusternode1

    * lpa_rh2_lpt                   : 1693873014
    * master-SAPHana_RH2_02         : 150
  * Node: clusternode2 (2):
    * hana_rh2_clone_state          : DEMOTED
    * hana_rh2_op_mode              : logreplay
    * hana_rh2_remoteHost           : clusternode1

    * hana_rh2_roles                : 4:S:master1:master:worker:master
    * hana_rh2_site                 : DC2
    * hana_rh2_sra                  : -
    * hana_rh2_srah                 : -
    * hana_rh2_srmode               : syncmem
    * hana_rh2_sync_state           : SOK
    * hana_rh2_version              : 2.00.062.00
    * hana_rh2_vhost                : clusternode2
    * lpa_rh2_lpt                   : 30
    * master-SAPHana_RH2_02         : 100

Migration Summary:

Tickets:

PCSD Status:
  clusternode1
: Online
  clusternode2: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

请参阅 检查集群状态和 Check 数据库，以验证所有是否再次工作。

第 6 章 有用的命令

以下是有用的命令的 3 部分。在大多数情况下，它应该有助于验证操作或配置是否成功。示例与响应一起列出。在某些情况下，出于格式化原因，输出已被调整。

注意

- 当由 < sid>adm 用户以 > 开头时，此文档中列出的所有命令均以 > 开头。
- root 用户运行 的所有命令都以 # 开头。
- 要执行命令，请省略 prefix > 或 #。

6.1. SAP HANA 命令

SAP HANA 命令由 < sid>adm 用户执行。Example:

```
[root@clusternode1]# su - rh2adm
clusternode1:rh2adm> cdpy
clusternode1:rh2adm> pwd
/usr/sap/RH2/HDB02/exe/python_support
clusternode1:rh2adm> python systemReplicationStatus.py -h
systemReplicationStatus.py [-h|--help] [-a|--all] [-l|--localhost] [-m|--multiTaget] [-s|--site=<site
name>] [-t|--printLandscapeTree] [--omitSecondaryActiveStatus] [--sapcontrol=1]
clusternode1:rh2adm> python landscapeHostConfiguration.py -h
landscapeHostConfiguration.py [-h|--help] [--localhost] [--sapcontrol=1]
clusternode1:rh2adm> hdbnsutil # run hdbnsutil without parameters to get help
```

6.1.1. 使用 hdbclm进行 SAP HANA 安装

第三个站点的安装与第二个站点的安装类似。安装可以使用 hdblcm 作为用户 root 来完成。为确保之前未安装任何内容，请运行 hdbuninst 以检查 SAP HANA 是否尚未安装在此节点上。

HANA 卸载的输出示例：

```
[root@remotehost3]# cd /software/DATA_UNITS/HDB_SERVER_LINUX_X86_64
root@DC3/software/DATA_UNITS/HDB_SERVER_LINUX_X86_64# ./hdbuninst
Option 0 will remove an already existing HANA Installation
```

No SAP HANA Installation found is the expected answer

DC3 上 HANA 安装的输出示例：

```
----[root@remotehost3]# cd /software/DATA_UNITS/HDB_SERVER_LINUX_X86_64
# ./hdbuninst
Option 0 will remove an already existing HANA Installation
No SAP HANA Installation found is the expected answer
----
Example output of HANA installation:
[source,text]
----
[root@remotehost3]# ./hdblcm
1 install
2 server
/hana/shared is default directory
Enter Local Hostname [remotehost3]: use the default name
additional hosts only during Scale-Out Installation y default is n
ENTER SAP HANA System ID: RH2
Enter Instance Number [02]:
Enter Local Host Worker Group [default]:
Select System Usage / Enter Index [4]:
Choose encryption
Enter Location of Data Volumes [/hana/data/RH2]:
Enter Location of Log Volumes [/hana/log/RH2]:
Restrict maximum memory allocation? [n]:
Enter Certificate Host Name
Enter System Administrator (rh2adm) Password: <Y0urPasswd>
Confirm System Administrator (rh2adm) Password: <Y0urPasswd>
Enter System Administrator Home Directory [/usr/sap/RH2/home]:
Enter System Administrator Login Shell [/bin/sh]:
Enter System Administrator User ID [1000]:
Enter System Database User (SYSTEM) Password: <Y0urPasswd>
Confirm System Database User (SYSTEM) Password: <Y0urPasswd>
Restart system after machine reboot? [n]:
----
```

在安装开始前，会列出概述：

```
SAP HANA Database System Installation
  Installation Parameters
    Remote Execution: ssh
    Database Isolation: low
    Install Execution Mode: standard
    Installation Path: /hana/shared
    Local Host Name: dc3host
    SAP HANA System ID: RH2
    Instance Number: 02
    Local Host Worker Group: default
    System Usage: custom
    Location of Data Volumes: /hana/data/RH2
```

```
    Location of Log Volumes: /hana/log/RH2
    SAP HANA Database secure store: ssfs
    Certificate Host Names: remotehost3 -> remotehost3      System Administrator Home
Directory: /usr/sap/RH2/home
    System Administrator Login Shell: /bin/sh
    System Administrator User ID: 1000
    ID of User Group (sapsys): 1010
  Software Components
    SAP HANA Database
      Install version 2.00.052.00.1599235305
      Location: /software/DATA_UNITS/HDB_SERVER_LINUX_X86_64/server
    SAP HANA Local Secure Store
      Do not install
    SAP HANA AFL (incl.PAL,BFL,OFL)
      Do not install
    SAP HANA EML AFL
      Do not install
    SAP HANA EPM-MDS
      Do not install
    SAP HANA Database Client
      Do not install
    SAP HANA Studio
      Do not install
    SAP HANA Smart Data Access
      Do not install
    SAP HANA XS Advanced Runtime
      Do not install
  Log File Locations
    Log directory: /var/tmp/hdb_RH2_hdblcm_install_2021-06-09_18.48.13
    Trace location: /var/tmp/hdblcm_2021-06-09_18.48.13_31307.trc

Do you want to continue? (y/n):
```

输入 y 开始安装。

6.1.2. 使用 hdbsql 检查 Inifile 内容

```
clusternode1:rh2adm> hdbsql -i ${TINSTANCE} -u system -p Y0urP8ssw0rd

Welcome to the SAP HANA Database interactive terminal.

Type:  \h for help with commands
       \q to quit

hdbsql RH2=> select * from M_INIFILE_CONTENTS where section='system_replication'
FILE_NAME,LAYER_NAME,TENANT_NAME,HOST,SECTION,KEY,VALUE
"global.ini","DEFAULT","","","system_replication","actual_mode","primary"
"global.ini","DEFAULT","","","system_replication","mode","primary"
"global.ini","DEFAULT","","","system_replication","operation_mode","logreplay"
"global.ini","DEFAULT","","","system_replication","register_secondaries_on_takeover
","true"
"global.ini","DEFAULT","","","system_replication","site_id","1"
"global.ini","DEFAULT","","","system_replication","site_name","DC2"
```

```
"global.ini","DEFAULT","","","system_replication","timetravel_logreplay_mode","auto
"
"global.ini","DEFAULT","","","system_replication","alternative_sources",""
"global.ini","DEFAULT","","","system_replication","datashipping_logsize_threshold",
"5368709120"
"global.ini","DEFAULT","","","system_replication","datashipping_min_time_interval",
"600"
"global.ini","DEFAULT","","","system_replication","datashipping_parallel_channels",
"4"
"global.ini","DEFAULT","","","system_replication","datashipping_parallel_processing
","true"
"global.ini","DEFAULT","","","system_replication","datashipping_snapshot_max_retent
ion_time","300"
"global.ini","DEFAULT","","","system_replication","enable_data_compression","false"
"global.ini","DEFAULT","","","system_replication","enable_full_sync","false"
"global.ini","DEFAULT","","","system_replication","enable_log_compression","false"
"global.ini","DEFAULT","","","system_replication","enable_log_retention","auto"
"global.ini","DEFAULT","","","system_replication","full_replica_on_failed_delta_syn
c_check","false"
"global.ini","DEFAULT","","","system_replication","hint_based_routing_site_name",""
"global.ini","DEFAULT","","","system_replication","keep_old_style_alert","false"
"global.ini","DEFAULT","","","system_replication","logshipping_async_buffer_size","
67108864"
"global.ini","DEFAULT","","","system_replication","logshipping_async_wait_on_buffer
_full","true"
"global.ini","DEFAULT","","","system_replication","logshipping_max_retention_size",
"1048576"
"global.ini","DEFAULT","","","system_replication","logshipping_replay_logbuffer_cac
he_size","1073741824"
"global.ini","DEFAULT","","","system_replication","logshipping_replay_push_persiste
nt_segment_count","5"
"global.ini","DEFAULT","","","system_replication","logshipping_snapshot_logsize_thr
eshold","3221225472"
"global.ini","DEFAULT","","","system_replication","logshipping_snapshot_min_time_in
terval","900"
"global.ini","DEFAULT","","","system_replication","logshipping_timeout","30"
"global.ini","DEFAULT","","","system_replication","preload_column_tables","true"
"global.ini","DEFAULT","","","system_replication","propagate_log_retention","off"
"global.ini","DEFAULT","","","system_replication","reconnect_time_interval","30"
"global.ini","DEFAULT","","","system_replication","retries_before_register_to_alter
native_source","20"
"global.ini","DEFAULT","","","system_replication","takeover_esserver_without_log_ba
ckup","false"
"global.ini","DEFAULT","","","system_replication","takeover_wait_until_esserver_res
tart","true"
"global.ini","DEFAULT","","","system_replication","timetravel_call_takeover_hooks",
"off"
"global.ini","DEFAULT","","","system_replication","timetravel_log_retention_policy"
,"none"
"global.ini","DEFAULT","","","system_replication","timetravel_max_retention_time","
0"
"global.ini","DEFAULT","","","system_replication","timetravel_snapshot_creation_int
erval","1440"
"indexserver.ini","DEFAULT","","","system_replication","logshipping_async_buffer_si
ze","268435456"
"indexserver.ini","DEFAULT","","","system_replication","logshipping_replay_logbuffe
```

```
r_cache_size","4294967296"
"indexserver.ini","DEFAULT","","","system_replication","logshipping_replay_push_per
sistent_segment_count","20"
41 rows selected (overall time 1971.958 msec; server time 31.359 msec)
```

6.1.3. 检查数据库

检查数据库是否正在运行，并发现当前的主节点。

列出数据库实例

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function
GetSystemInstanceList

23.06.2023 12:08:17
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
node1, 2, 50213, 50214, 0.3, HDB|HDB_WORKER, GREEN
```

如果输出为绿色，则实例正在运行。

列出数据库进程

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function GetProcessList
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2023 09 04 14:34:01, 18:41:33, 3788067
hdbcompileserver, HDB Compileserver, GREEN, Running, 2023 09 04 22:35:40, 10:39:54, 445299
hdbindexserver, HDB Indexserver-RH2, GREEN, Running, 2023 09 04 22:35:40, 10:39:54, 445391
hdbnameserver, HDB Nameserver, GREEN, Running, 2023 09 04 22:35:34, 10:40:00, 445178
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2023 09 04 22:35:40, 10:39:54, 445306
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2023 09 04 22:35:53, 10:39:41, 445955
hdbxsengine, HDB XSEngine-RH2, GREEN, Running, 2023 09 04 22:35:40, 10:39:54, 445394
```

通常，所有数据库进程都处于 **GREEN** 状态。

列出 **SAP HANA** 进程

```
clusternode1:rh2adm> HDB info
USER          PID     PPID  %CPU        VSZ        RSS COMMAND
```

```
rh2adm      1560     1559   0.0      6420      3136 watch -n 5 sapcontrol -nr 02 -functi
rh2adm      1316     1315   0.0      8884      5676 -sh
rh2adm      2549     1316   0.0      7516      4072  \_ /bin/sh /usr/sap/RH2/HDB02/HDB i
rh2adm      2579     2549   0.0     10144      3576      \_ ps fx -U rh2adm -o user:8,pi
rh2adm      2388        1   0.0    679536     55520 hdbrsutil  --start --port 30203 --vo
rh2adm      1921        1   0.0    679196     55312 hdbrsutil  --start --port 30201 --vo
rh2adm      1469        1   0.0      8852      3260 sapstart pf=/usr/sap/RH2/SYS/profile
rh2adm      1476     1469   0.7    438316     86288  \_ /usr/sap/RH2/HDB02/remotehost3/trace/
rh2adm      1501     1476  11.7   9690172   1574796      \_ hdbnameserver
rh2adm      1845     1476   0.8    410696    122988      \_ hdbcompileserver
rh2adm      1848     1476   1.0    659464    154072      \_ hdbpreprocessor
rh2adm      1899     1476  14.7   9848276   1765208      \_ hdbindexserver -port 30203
rh2adm      1902     1476   8.4   5023288   1052768      \_ hdbxsengine -port 30207
rh2adm      2265     1476   5.2   2340284    405016      \_ hdbwebdispatcher
rh2adm      1117        1   1.1    543532     30676 /usr/sap/RH2/HDB02/exe/sapstartsrv p
rh2adm      1029        1   0.0     20324     11572 /usr/lib/systemd/systemd --user
rh2adm      1030     1029   0.0     23256      3536  \_ (sd-pam)
```

显示 SAP HANA landscape 配置

```
clusternode1:rh2adm>
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/landscapeHostConfiguration.py
;echo $?
| Host   | Host   | Host   | Failover | Remove | Storage   | Storage   | Failover | Failover | NameServer |
NameServer | IndexServer | IndexServer | Host   | Host   | Worker  | Worker  |
|        | Active | Status | Status   | Status | Config    | Actual    | Config   | Actual   | Config     | Actual     |
Config       | Actual       | Config | Actual | Config  | Actual  |
|        |        |        |          |        | Partition | Partition | Group    | Group    | Role       | Role       | Role        |
Role         | Roles  | Roles  | Groups  | Groups  |
| ------ | ------ | ------ | -------- | ------ | --------- | --------- | -------- | -------- | ---------- | ---------- | -----------
| ----------- | ------ | ------ | ------- | ------- |
| clusternode1 | yes    | ok     |          |        |         1 |         1 | default  | default  | master 1   | master     |
worker       | master       | worker | worker | default | default |

overall host status: ok
4
```

返回码：

- **0: fatal**
- **1: error**
- **2: warning**

- 3: info
- 4: OK

发现主数据库

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "primary masters|^mode"
```

辅助检查示例：

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "primary masters|^mode"
mode: syncmem
primary masters: clusternode1
```

在当前主上检查示例：

```
clusternode1:rh2adm> hdbnsutil -sr_state | egrep -e "primary masters|^mode"
mode: primary

clusternode1:rh2adm>hdbnsutil -sr_state --sapcontrol=1 |grep  site.*Mode
siteReplicationMode/DC1=primary
siteReplicationMode/DC3=async
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC3=logreplay
siteOperationMode/DC2=logreplay
```

显示数据库版本

使用 SQL 查询的示例：

```
hdbsql RH2=> select * from m_database
SYSTEM_ID,DATABASE_NAME,HOST,START_TIME,VERSION,USAGE
"RH2","RH2","node1","2023-06-22 15:33:05.235000000","2.00.059.02.1647435895","CUSTOM"
1 row selected (overall time 29.107 msec; server time 927 usec)
```

使用 systemOverview.py 的示例：

```
clusternode1:rh2adm> python ./systemOverview.py
| Section    | Name           | Status  | Value                                                |
| ---------- | -------------- | ------- | ---------------------------------------------------- |
| System     | Instance ID    |         | RH2                                                  |
| System     | Instance Number |        | 02                                                   |
| System     | Distributed    |         | No                                                   |
| System     | Version        |         | 2.00.059.02.1647435895 (fa/hana2sp05)                |
| System     | Platform       |         | Red Hat Enterprise Linux 9.2 Beta (Plow) 9.2 (Plow) |
| Services   | All Started    | OK      | Yes                                                  |
| Services   | Min Start Time |         | 2023-07-14 16:31:19.000                              |
| Services   | Max Start Time |         | 2023-07-26 11:23:17.324                              |
| Memory     | Memory         | OK      | Physical 31.09 GB, Swap 10.00 GB, Used 26.38         |
| CPU        | CPU            | OK      | Available 4, Used 1.04                               |
| Disk       | Data           | OK      | Size 89.0 GB, Used 59.3 GB, Free 33 %                |
| Disk       | Log            | OK      | Size 89.0 GB, Used 59.3 GB, Free 33 %                |
| Disk       | Trace          | OK      | Size 89.0 GB, Used 59.3 GB, Free 33 %                |
| Statistics | Alerts         | WARNING | cannot check statistics w/o SQL connection           |
```

6.1.4. 启动和停止 SAP HANA

选项 **1**：**HDB** 命令

```
clusternode1:rh2adm> HDB help
Usage: /usr/sap/RH2/HDB02/HDB { start|stop|reconf|restart|version|info|proc|admin|kill|kill-<sig>|term }
  kill or kill-9 should never be used in a productive environment!
```

- 启动数据库

  ```
  clusternode1:rh2adm> HDB start
  ```

- 停止数据库

  ```
  clusternode1:rh2adm> HDB stop
  ```

选项 **2**（推荐）：使用 **sapcontrol**

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function StartSystem HDB

03.07.2023 14:08:30
StartSystem
OK
```

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function StopSystem HDB
```

```
03.07.2023 14:09:33
StopSystem
OK
```

使用 GetProcessList 来监控 HANA 服务的启动和停止：

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function GetProcessList
```

6.1.5. 检查 SAP HANA System Replication 状态

有多种方法可以检查 SAP HANA 系统复制状态：

- 'clusternode1:rh2adm> python systemReplicationStatus.py ' on the primary node
- clusternode1:rh2adm> echo $? # (Return code of systemReplicationStatus)
- clusternode1:rh2adm> hdbnsutil -sr_state
- clusternode1:rh2adm> hdbnsutil -sr_stateConfiguration

作为监控器运行的 systemReplicationStatus.py 输出示例：

```
clusternode1:rh2adm> watch -n 5 "python
/usr/sap/${SAPSYSTEMNAME}/HDB{TINSTACE}/exe/python_support/systemReplicationStatus.py;ech
o \$?"
 concurrent-fencing: true
Every 5.0s: python systemReplicationStatus.py;echo $?
hana08: Fri Jul 28 17:01:05 2023

|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 |hana08 |30201 |nameserver   |       1 |  1 |DC2       |hana09    |    30201 |       3 |DC3
|YES          |SYNCMEM     |ACTIVE    |             |
|RH2   |hana08 |30207 |xsengine     |       2 |  1 |DC2       |hana09    |    30207 |       3 |DC3   |YES
|SYNCMEM     |ACTIVE    |             |
|RH2   |hana08 |30203 |indexserver  |       3 |  1 |DC2       |hana09    |    30203 |       3 |DC3   |YES
|SYNCMEM     |ACTIVE    |             |
```

```
|SYSTEMDB |hana08 |30201 |nameserver   |       1 |  1 |DC2       |remotehost3    |    30201 |       2
|DC1   |YES          |SYNCMEM     |ACTIVE    |             |
|RH2   |hana08 |30207 |xsengine     |       2 |  1 |DC2       |remotehost3    |    30207 |       2 |DC1
|YES          |SYNCMEM     |ACTIVE    |             |
|RH2   |hana08 |30203 |indexserver  |       3 |  1 |DC2       |remotehost3    |    30203 |       2 |DC1
|YES          |SYNCMEM     |ACTIVE    |             |

status system replication site "3": ACTIVE
status system replication site "2": ACTIVE
overall system replication status: ACTIVE

Local System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mode: PRIMARY
site id: 1
site name: DC2
15
```

返回代码的预期结果为：

- **10: NoHSR**
- **11: error**
- **12: unknown**
- **13: 初始化**
- **14: 同步**
- **15: active**

在大多数情况下，系统复制检查将返回返回码 15。另一个显示选项是使用 -t (printLandscapeTree)。

当前主机上的输出示例：

```
clusternode1:rh2adm> python systemReplicationStatus.py -t
```

```
HANA System Replication landscape:
  DC1  ( primary )
    | --- DC3  ( syncmem )
    | --- DC2  ( syncmem )
```

hdbnsutil -sr_state 示例：

```
[root@clusternode1]# su - rh2adm
clusternode1:rh2adm> watch -n 10 hdbnsutil -sr_state
Every 10.0s: hdbnsutil -sr_state                                clusternode1: Thu
Jun 22 08:42:00 2023


System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

online: true

mode: syncmem
operation mode: logreplay
site id: 2
site name: DC1

is source system: false
is secondary/consumer system: true
has secondaries/consumers attached: false
is a takeover active: false
is primary suspended: false
is timetravel enabled: false
replay mode: auto
active primary site: 1

primary masters: clusternode2

Host Mappings:
~~~~~~~~~~~~~~

clusternode1 -> [DC3] remotehost3
clusternode1 -> [DC1] clusternode1
clusternode1 -> [DC2] clusternode2


Site Mappings:
~~~~~~~~~~~~~~
DC2 (primary/primary)
    |---DC3 (syncmem/logreplay)
    |---DC1 (syncmem/logreplay)

Tier of DC2: 1
Tier of DC3: 2
Tier of DC1: 2

Replication mode of DC2: primary
[2] 0:ssh*
```

-

主上的 sr_stateConfiguation 示例：

```
clusternode1:rh2adm> hdbnsutil -sr_stateConfiguration

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

mode: primary
site id: 2
site name: DC1
done.
```

二级上的 sr_stateConfiguration 示例：

```
clusternode1:rh2adm> hdbnsutil -sr_stateConfiguration

System Replication State
~~~~~~~~~~~~~~~~~~~~~~~~

mode: syncmem
site id: 1
site name: DC2
active primary site: 2

primary masters: clusternode1
done.
```

您还可以在节点是当前主的辅助数据库中检查。在故障转移过程中，会出现两个主数据库，并且需要这些信息来确定潜在的主数据库错误，需要重新注册为次要数据库。

如需更多信息，请参阅示例 ：检查 Primary 和 Secondary Systems 上的状态。

6.1.6. 注册辅助节点

为 SAP HANA 系统复制环境注册辅助数据库的先决条件：

- 创建 SAP HANA 备份
- 在主节点上启用 SAP HANA 系统复制

- 复制数据库密钥
- 注册二级节点

注册示例：

```
clusternode1:rh2adm> hdbnsutil -sr_register --remoteHost=clusternode2 --
remoteInstance=${TINSTANCE} --replicationMode=syncmem --name=DC1 --online
--operationMode not set; using default from global.ini/[system_replication]/operation_mode:
logreplay
adding site ...
collecting information ...
updating local ini files ...
done.
```

在注册了 global.ini 文件后，将自动更新

…来自：

```
# global.ini last modified 2023-06-15 09:55:05.665341 by /usr/sap/RH2/HDB02/exe/hdbnsutil -
initTopology --workergroup=default --set_user_system_pw
[multidb]
mode = multidb
database_isolation = low
singletenant = yes

[persistence]
basepath_datavolumes = /hana/data/RH2
basepath_logvolumes = /hana/log/RH2
```

…至：

```
# global.ini last modified 2023-06-15 11:25:44.516946 by hdbnsutil -sr_register --
remoteHost=node2 --remoteInstance=02 --replicationMode=syncmem --name=DC1 --online
[multidb]
mode = multidb
database_isolation = low
singletenant = yes

[persistence]
basepath_datavolumes = /hana/data/RH2
basepath_logvolumes = /hana/log/RH2
```

```
[system_replication]
timetravel_logreplay_mode = auto
site_id = 3
mode = syncmem
actual_mode = syncmem
site_name = DC1
operation_mode = logreplay

[system_replication_site_masters]
1 = clusternode2:30201
```

6.1.7. sapcontrol GetProcessList

检查活跃的 SAP HANA 数据库的进程

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function GetProcessList
clusternode1: Wed Jun  7 08:23:03 2023

07.06.2023 08:23:03
GetProcessList
OK
name, description, dispstatus, textstatus, starttime, elapsedtime, pid
hdbdaemon, HDB Daemon, GREEN, Running, 2023 06 02 16:59:42, 111:23:21, 4245
hdbcompileserver, HDB Compileserver, GREEN, Running, 2023 06 02 17:01:35, 111:21:28, 7888
hdbindexserver, HDB Indexserver-RH2, GREEN, Running, 2023 06 02 17:01:36, 111:21:27, 7941
hdbnameserver, HDB Nameserver, GREEN, Running, 2023 06 02 17:01:29, 111:21:34, 7594
hdbpreprocessor, HDB Preprocessor, GREEN, Running, 2023 06 02 17:01:35, 111:21:28, 7891
hdbwebdispatcher, HDB Web Dispatcher, GREEN, Running, 2023 06 02 17:01:42, 111:21:21,
8339
hdbxsengine, HDB XSEngine-RH2, GREEN, Running, 2023 06 02 17:01:36, 111:21:27, 7944
```

6.1.8. sapcontrol GetInstanceList

这将列出 SAP HANA 数据库的实例状态。它还将显示端口。有三个不同的状态名称：

- GREEN （运行）
- GRAY （停止）
- YELLOW （当前正在更改）

活跃实例示例：

-

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function GetSystemInstanceList
clusternode1: Wed Jun  7 08:24:13 2023


07.06.2023 08:24:13
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
remotehost3, 2, 50213, 50214, 0.3, HDB|HDB_WORKER, GREEN
```

停止的实例示例：

```
clusternode1:rh2adm> sapcontrol -nr ${TINSTANCE} -function GetSystemInstanceList

22.06.2023 09:14:55
GetSystemInstanceList
OK
hostname, instanceNr, httpPort, httpsPort, startPriority, features, dispstatus
remotehost3, 2, 50213, 50214, 0.3, HDB|HDB_WORKER, GRAY
```

6.1.9. hdbcons 示例

您还可以使用 HDB 控制台显示数据库的信息：

- hdbcons -e hdbindexserver 'replication info'
- hdbcons -e hdbindexserver 帮助 更多选项

'replication info' 示例：

```
clusternode1:rh2adm> hdbcons -e hdbindexserver 'replication info'
hdbcons -p `pgrep hdbindex` 'replication info'
SAP HANA DB Management Client Console (type '\?' to get help for client commands)
Try to open connection to server process with PID 451925
SAP HANA DB Management Server Console (type 'help' to get help for server commands)
Executable: hdbindexserver (PID: 451925)
[OK]
--
## Start command at: 2023-06-22 09:05:25.211
listing default statistics for volume 3
System Replication Primary Information
=====================================
System Replication Primary Configuration
 [system_replication] logshipping_timeout                = 30
 [system_replication] enable_full_sync                   = false
```

```
 [system_replication] preload_column_tables                = true
 [system_replication] ensure_backup_history                = true
 [system_replication_communication] enable_ssl             = off
 [system_replication] keep_old_style_alert                 = false
 [system_replication] enable_log_retention                 = auto
 [system_replication] logshipping_max_retention_size       = 1048576
 [system_replication] logshipping_async_buffer_size        = 268435456
 - lastLogPos              : 0x4ab2700
 - lastLogPosTimestamp     : 22.06.2023-07.05.25 (1687417525193952)
 - lastConfirmedLogPos     : 0x4ab2700
 - lastConfirmedLogPosTimestamp: 22.06.2023-07.05.25 (1687417525193952)
 - lastSavepointVersion    : 1286
 - lastSavepointLogPos     : 0x4ab0602
 - lastSavepointTimestamp  : 22.06.2023-07.02.42 (1687417362853007)
 2 session registered.
Session index 0
 - SiteID          : 3
 - RemoteHost      : 192.168.5.137
Log Connection
 - ptr             : 0x00007ff04c0a1000
 - channel         : {<NetworkChannelSSLFilter>={<NetworkChannelBase>=
{this=140671686293528, fd=70, refCnt=2, idx=5, local=192.168.5.134/40203_tcp,
remote=192.168.5.137/40406_tcp, state=Connected, pending=[r---]}}}
 - SSLActive       : false
 - mode            : syncmem
Data Connection
 - ptr             : 0x00007ff08b730000
 - channel         : {<NetworkChannelSSLFilter>={<NetworkChannelBase>=
{this=140671436247064, fd=68, refCnt=2, idx=6, local=192.168.5.134/40203_tcp,
remote=192.168.5.137/40408_tcp, state=Connected, pending=[r---]}}}
 - SSLActive       : false
Primary Statistics
 - Creation Timestamp          : 20.06.2023-13.55.07 (1687269307772532)
 - Last Reset Timestamp        : 20.06.2023-13.55.07 (1687269307772532)
 - Statistic Reset Count       : 0
 - ReplicationMode             : syncmem
 - OperationMode               : logreplay
 - ReplicationStatus           : ReplicationStatus_Active
 - ReplicationStatusDetails    :
 - ReplicationFullSync         : DISABLED
 - shippedLogPos               : 0x4ab2700
 - shippedLogPosTimestamp      : 22.06.2023-07.05.25 (1687417525193952)
 - sentLogPos                  : 0x4ab2700
 - sentLogPosTimestamp         : 22.06.2023-07.05.25 (1687417525193952)
 - sentMaxLogWriteEndPosition  : 0x4ab2700
 - sentMaxLogWriteEndPositionReqCnt: 0x1f6b8
 - shippedLogBuffersCount      : 142439
 - shippedLogBuffersSize       : 805855232 bytes
 - shippedLogBuffersSizeUsed   : 449305792 bytes (55.76clusternode1:rh2adm>)
 - shippedLogBuffersSizeNet    : 449013696 bytes (55.72clusternode1:rh2adm>)
 - shippedLogBufferDuration    : 83898615 microseconds
 - shippedLogBufferDurationMin : 152 microseconds
 - shippedLogBufferDurationMax : 18879 microseconds
 - shippedLogBufferDurationSend: 7301067 microseconds
 - shippedLogBufferDurationComp: 0 microseconds
 - shippedLogBufferThroughput  : 9709099.18 bytes/s
```

```
- shippedLogBufferPendingDuration : 80583785 microseconds
- shippedLogBufferRealThrougput   : 10073190.40 bytes/s
- replayLogPos                    : 0x4ab2700
- replayLogPosTimestamp           : 22.06.2023-07.05.25 (1687417525193952)
- replayBacklog                   : 0 microseconds
- replayBacklogSize               : 0 bytes
- replayBacklogMax                : 822130896 microseconds
- replayBacklogSizeMax            : 49455104 bytes
- shippedSavepointVersion         : 0
- shippedSavepointLogPos          : 0x0
- shippedSavepointTimestamp       : not set
- shippedFullBackupCount          : 0
- shippedFullBackupSize           : 0 bytes
- shippedFullBackupSizeNet        : 0 bytes (-nanclusternode1:rh2adm>)
- shippedFullBackupDuration       : 0 microseconds
- shippedFullBackupDurationComp   : 0 microseconds
- shippedFullBackupThroughput     : 0.00 bytes/s
- shippedFullBackupStreamCount    : 0
- shippedFullBackupResumeCount    : 0
- shippedLastFullBackupSize       : 0 bytes
- shippedLastFullBackupSizeNet    : 0 bytes (-nanclusternode1:rh2adm>)
- shippedLastFullBackupStart      : not set
- shippedLastFullBackupEnd        : not set
- shippedLastFullBackupDuration   : 0 microseconds
- shippedLastFullBackupStreamCount : 0
- shippedLastFullBackupResumeCount : 0
- shippedDeltaBackupCount         : 0
- shippedDeltaBackupSize          : 0 bytes
- shippedDeltaBackupSizeNet       : 0 bytes (-nanclusternode1:rh2adm>)
- shippedDeltaBackupDuration      : 0 microseconds
- shippedDeltaBackupDurationComp  : 0 microseconds
- shippedDeltaBackupThroughput    : 0.00 bytes/s
- shippedDeltaBackupStreamCount   : 0
- shippedDeltaBackupResumeCount   : 0
- shippedLastDeltaBackupSize      : 0 bytes
- shippedLastDeltaBackupSizeNet   : 0 bytes (-nanclusternode1:rh2adm>)
- shippedLastDeltaBackupStart     : not set
- shippedLastDeltaBackupEnd       : not set
- shippedLastDeltaBackupDuration  : 0 microseconds
- shippedLastDeltaBackupStreamCount : 0
- shippedLastDeltaBackupResumeCount : 0
- currentTransferType             : None
- currentTransferSize             : 0 bytes
- currentTransferPosition         : 0 bytes (0clusternode1:rh2adm>)
- currentTransferStartTime        : not set
- currentTransferThroughput       : 0.00 MB/s
- currentTransferStreamCount      : 0
- currentTransferResumeCount      : 0
- currentTransferResumeStartTime  : not set
- Secondary sync'ed via Log Count : 1
- syncLogCount                    : 3
- syncLogSize                     : 62840832 bytes
- backupHistoryComplete           : 1
- backupLogPosition               : 0x4a99980
- backupLogPositionUpdTimestamp   : 22.06.2023-06.56.27 (0x5feb26227e7af)
- shippedMissingLogCount          : 0
```

```
 - shippedMissingLogSize          : 0 bytes
 - backlogSize                    : 0 bytes
 - backlogTime                    : 0 microseconds
 - backlogSizeMax                 : 0 bytes
 - backlogTimeMax                 : 0 microseconds
 - Secondary Log Connect time     : 20.06.2023-13.55.31 (1687269331361049)
 - Secondary Data Connect time    : 20.06.2023-13.55.33 (1687269333768341)
 - Secondary Log Close time       : not set
 - Secondary Data Close time      : 20.06.2023-13.55.31 (1687269331290050)
 - Secondary Log Reconnect Count  : 0
 - Secondary Log Failover Count   : 0
 - Secondary Data Reconnect Count : 1
 - Secondary Data Failover Count  : 0
----------------------------------------------------------------
Session index 1
 - SiteID            : 2
 - RemoteHost        : 192.168.5.133
Log Connection
 - ptr               : 0x00007ff0963e4000
 - channel           : {<NetworkChannelSSLFilter>={<NetworkChannelBase>=
{this=140671506282520, fd=74, refCnt=2, idx=0, local=192.168.5.134/40203_tcp,
remote=192.168.5.133/40404_tcp, state=Connected, pending=[r---]}}}
 - SSLActive         : false
 - mode              : syncmem
Data Connection
 - ptr               : 0x00007ff072c04000
 - channel           : {<NetworkChannelSSLFilter>={<NetworkChannelBase>=
{this=140671463146520, fd=75, refCnt=2, idx=1, local=192.168.5.134/40203_tcp,
remote=192.168.5.133/40406_tcp, state=Connected, pending=[r---]}}}
 - SSLActive         : false
Primary Statistics
 - Creation Timestamp             : 20.06.2023-13.55.49 (1687269349892111)
 - Last Reset Timestamp           : 20.06.2023-13.55.49 (1687269349892111)
 - Statistic Reset Count          : 0
 - ReplicationMode                : syncmem
 - OperationMode                  : logreplay
 - ReplicationStatus              : ReplicationStatus_Active
 - ReplicationStatusDetails       :
 - ReplicationFullSync            : DISABLED
 - shippedLogPos                  : 0x4ab2700
 - shippedLogPosTimestamp         : 22.06.2023-07.05.25 (1687417525193952)
 - sentLogPos                     : 0x4ab2700
 - sentLogPosTimestamp            : 22.06.2023-07.05.25 (1687417525193952)
 - sentMaxLogWriteEndPosition     : 0x4ab2700
 - sentMaxLogWriteEndPositionReqCnt: 0x1f377
 - shippedLogBuffersCount         : 142326
 - shippedLogBuffersSize          : 793939968 bytes
 - shippedLogBuffersSizeUsed      : 437675200 bytes (55.13clusternode1:rh2adm>)
 - shippedLogBuffersSizeNet       : 437565760 bytes (55.11clusternode1:rh2adm>)
 - shippedLogBufferDuration       : 76954026 microseconds
 - shippedLogBufferDurationMin    : 115 microseconds
 - shippedLogBufferDurationMax    : 19285 microseconds
 - shippedLogBufferDurationSend   : 2951495 microseconds
 - shippedLogBufferDurationComp   : 0 microseconds
 - shippedLogBufferThroughput     : 10446578.53 bytes/s
 - shippedLogBufferPendingDuration : 73848247 microseconds
```

```
- shippedLogBufferRealThrougput   : 10875889.97 bytes/s
- replayLogPos                    : 0x4ab2700
- replayLogPosTimestamp           : 22.06.2023-07.05.25 (1687417525193952)
- replayBacklog                   : 0 microseconds
- replayBacklogSize               : 0 bytes
- replayBacklogMax                : 113119944 microseconds
- replayBacklogSizeMax            : 30171136 bytes
- shippedSavepointVersion         : 0
- shippedSavepointLogPos          : 0x0
- shippedSavepointTimestamp       : not set
- shippedFullBackupCount          : 0
- shippedFullBackupSize           : 0 bytes
- shippedFullBackupSizeNet        : 0 bytes (-nanclusternode1:rh2adm>)
- shippedFullBackupDuration       : 0 microseconds
- shippedFullBackupDurationComp   : 0 microseconds
- shippedFullBackupThroughput     : 0.00 bytes/s
- shippedFullBackupStreamCount    : 0
- shippedFullBackupResumeCount    : 0
- shippedLastFullBackupSize       : 0 bytes
- shippedLastFullBackupSizeNet    : 0 bytes (-nanclusternode1:rh2adm>)
- shippedLastFullBackupStart      : not set
- shippedLastFullBackupEnd        : not set
- shippedLastFullBackupDuration   : 0 microseconds
- shippedLastFullBackupStreamCount : 0
- shippedLastFullBackupResumeCount : 0
- shippedDeltaBackupCount         : 0
- shippedDeltaBackupSize          : 0 bytes
- shippedDeltaBackupSizeNet       : 0 bytes (-nanclusternode1:rh2adm>)
- shippedDeltaBackupDuration      : 0 microseconds
- shippedDeltaBackupDurationComp  : 0 microseconds
- shippedDeltaBackupThroughput    : 0.00 bytes/s
- shippedDeltaBackupStreamCount   : 0
- shippedDeltaBackupResumeCount   : 0
- shippedLastDeltaBackupSize      : 0 bytes
- shippedLastDeltaBackupSizeNet   : 0 bytes (-nanclusternode1:rh2adm>)
- shippedLastDeltaBackupStart     : not set
- shippedLastDeltaBackupEnd       : not set
- shippedLastDeltaBackupDuration  : 0 microseconds
- shippedLastDeltaBackupStreamCount : 0
- shippedLastDeltaBackupResumeCount : 0
- currentTransferType             : None
- currentTransferSize             : 0 bytes
- currentTransferPosition         : 0 bytes (0clusternode1:rh2adm>)
- currentTransferStartTime        : not set
- currentTransferThroughput       : 0.00 MB/s
- currentTransferStreamCount      : 0
- currentTransferResumeCount      : 0
- currentTransferResumeStartTime  : not set
- Secondary sync'ed via Log Count : 1
- syncLogCount                    : 3
- syncLogSize                     : 61341696 bytes
- backupHistoryComplete           : 1
- backupLogPosition               : 0x4a99980
- backupLogPositionUpdTimestamp   : 22.06.2023-06.56.27 (0x5feb26227e670)
- shippedMissingLogCount          : 0
- shippedMissingLogSize           : 0 bytes
```

```
 - backlogSize                  : 0 bytes
 - backlogTime                   : 0 microseconds
 - backlogSizeMax                : 0 bytes
 - backlogTimeMax                : 0 microseconds
 - Secondary Log Connect time     : 20.06.2023-13.56.21 (1687269381053599)
 - Secondary Data Connect time    : 20.06.2023-13.56.27 (1687269387399610)
 - Secondary Log Close time       : not set
 - Secondary Data Close time      : 20.06.2023-13.56.21 (1687269381017244)
 - Secondary Log Reconnect Count  : 0
 - Secondary Log Failover Count   : 0
 - Secondary Data Reconnect Count : 1
 - Secondary Data Failover Count  : 0
----------------------------------------------------------------
[OK]
## Finish command at: 2023-06-22 09:05:25.212 command took: 572.000 usec
--
[EXIT]
--
[BYE]
```

帮助示例：

```
clusternode1:rh2adm> hdbcons  -e hdbindexserver help
SAP HANA DB Management Client Console (type '\?' to get help for client commands)
Try to open connection to server process with PID 451925
SAP HANA DB Management Server Console (type 'help' to get help for server commands)
Executable: hdbindexserver (PID: 451925)
[OK]
--
## Start command at: 2023-06-22 09:07:16.784
Synopsis:
help [<command name>]: Print command help
  - <command name> - Command name for which to display help
Available commands:
ae_tableload - Handle loading of column store tables and columns
all - Print help and other info for all hdbcons commands
authentication - Authentication management.
binarysemaphore - BinarySemaphore management
bye - Exit console client
cd - ContainerDirectory management
cfgreg - Basis Configurator
checktopic - CheckTopic management
cnd - ContainerNameDirectory management
conditionalvariable - ConditionalVariable management
connection - Connection management
context - Execution context management (i.e., threads)
converter - Converter management
cpuresctrl - Manage cpu resources such as last-level cache allocation
crash - Crash management
crypto - Cryptography management (SSL/SAML/X509/Encryption).
csaccessor - Display diagnostics related to the CSAccessor library
ddlcontextstore - Get DdlContextStore information
deadlockdetector - Deadlock detector.
debug - Debug management
```

```
distribute - Handling distributed systems
dvol - DataVolume management
ELF - ELF symbol resolution management
encryption - Persistence encryption management
eslog - Manipulate logger on extended storage
event - Event management
exit - Exit console client
flightrecorder - Flight Recorder
hananet - HANA-Net command interface
help - Display help for a command or command list
hkt - HANA Kernal Tracer (HKT) management
indexmanager - Get IndexManager information, especially for IndexHandles
itab - Internaltable diagnostics
jexec - Information and actions for Job Executor/Scheduler
licensing - Licensing management.
log - Show information about logger and manipulate logger
machine - Information about the machine topology
mm - Memory management
monitor - Monitor view command
mproxy - Malloc proxy management
msl - Mid size LOB management
mutex - Mutex management
numa - Provides NUMA statistics for all columns of a given table, broken down by column
constituents like dictionary, data vector and index.
nvmprovider - NVM Provider
output - Command for managing output from the hdbcons
page - Page management
pageaccess - PageAccess management
profiler - Profiler
quit - Exit console client
readwritelock - ReadWriteLock management
replication - Monitor data and log replication
resman - ResourceManager management
rowstore - Row Store
runtimedump - Generate a runtime dump.
savepoint - Savepoint management
semaphore - Semaphore management
servicethreads - Thread information M_SERVICE_THREADS
snapshot - Snapshot management
stat - Statistics management
statisticsservercontroller - StatisticsServer internals
statreg - Statistics registry command
syncprimi - Syncprimitive management (Mutex, CondVariable, Semaphore, BinarySemaphore,
ReadWriteLock)
table - Table Management
tablepreload - Manage and monitor table preload
trace - Trace management
tracetopic - TraceTopic management
transaction - Transaction management
ut - UnifiedTable Management
version - Version management
vf - VirtualFile management
x2 - get X2 info
[OK]
## Finish command at: 2023-06-22 09:07:16.785 command took: 209.000 usec
--
```

```
[EXIT]
--
[BYE]
```

6.1.10. 创建 SAP HANA 备份

如果要使用 SAP HANA 系统复制，必须首先在主系统上创建一个备份。

如何执行此操作的示例为用户 < sid>adm:

```
clusternode1:rh2adm> hdbsql -i ${TINSTANCE} -u system -d SYSTEMDB "BACKUP DATA
USING FILE ('/hana/backup/')"
clusternode1:rh2adm> hdbsql -i ${TINSTANCE} -u system -d ${SAPSYSTEMNAME} "BACKUP
DATA USING FILE ('/hana/backup/')"
```

6.1.11. 在主数据库上启用 SAP HANA 系统复制

必须在主节点上启用 SAP HANA 系统复制。这要求首先进行备份。

```
clusternode1:rh2adm> hdbnsutil -sr_enable --name=DC1
nameserver is active, proceeding ...
successfully enabled system as system replication source site
done.
```

6.1.12. 将数据库密钥复制到辅助节点

数据库密钥需要从主数据库复制到次要数据库，然后才能将其注册为次要数据库。

例如：

```
clusternode1:rh2adm> scp -rp
/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/data/SSFS_${SAPSYSTEMNAME}.
DAT
remotehost3:/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/data/SSFS_${SAPSY
STEMNAME}.DAT
clusternode1:rh2adm> scp -rp
/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/key/SSFS_${SAPSYSTEMNAME}.K
EY
remotehost3:/usr/sap/${SAPSYSTEMNAME}/SYS/global/security/rsecssfs/key/SSFS_${SAPSYS
TEMNAME}.KEY
```

6.1.13. 为 SAP HANA 系统复制注册辅助节点

请确定数据库密钥已首先复制到次要节点。然后运行注册命令：

```
clusternode1:rh2adm> hdbnsutil -sr_register --remoteHost=remotehost3 --
remoteInstance=${TINSTANCE} --replicationMode=syncmem --name=DC1 --remoteName=DC3
--operationMode=logreplay --online
```

参数描述：

- remotehost: 运行源（主）数据库的活动节点的主机名
- remoteInstance ：数据库的实例数
- replicationMode ：以下选项之一
  - 同步 ：硬盘同步
  - async: 异步复制
  - syncmem: 内存同步
- 名称 ：这是此复制站点的别名
- remoteName ：源数据库的别名名称
- operationMode ：以下选项之一
  - delta_datashipping ：定期传输数据。接管需要更长的时间。
  - logreplay: 在远程站点上立即为 redone 日志。接管速度更快。

- logreplay_readaccess ：可能对第二个站点进行额外的 logreplay 只读访问。

6.1.14. 检查 SAP HANA 数据库的 log_mode

设置 log_mode 有两个选项：

- log_mode=overwrite
- log_mode=normal ：这是默认值，在数据库实例作为主要实例运行时也是必需的。使用 SAP HANA Multitarget System Replication，您必须使用 log_mode=normal。检查 log_mode 的最佳方法是使用 hdbsql ：

包括错误的 覆盖 条目示例：

```
clusternode1:rh2adm> hdbsql -i ${TINSTANCE} -d ${SAPSYSTEMNAME} -u system
Password:

Welcome to the SAP HANA Database interactive terminal.

Type:  \h for help with commands
       \q to quit

hdbsql RH2=> select * from m_inifile_contents where key='log_mode'
FILE_NAME,LAYER_NAME,TENANT_NAME,HOST,SECTION,KEY,VALUE
"global.ini","DEFAULT","","","persistence","log_mode","normal"
"global.ini","HOST","","node2","persistence","log_mode","overwrite"
2 rows selected (overall time 46.931 msec; server time 30.845 msec)

hdbsql RH2=>exit
```

在这种情况下，我们有两个 global.ini 文件：

- DEFAULT
  - /usr/sap/${SAPSYSTEMNAME}/SYS/global/hdb/custom/config/global.ini
- HOST

- /HANA/shared/${SAPSYSTEMNAME}/HDB${TINSTANCE}/${HOSTNAME}/global.ini The HOST 值覆盖 DEFAULT 值。您还可以在数据库启动前检查这两个文件，然后再次使用 hdbsql 验证正确的设置。您可以通过编辑 global.ini 文件来更改 log_mode。

Example:

```
clusternode1:rh2adm> vim
/hana/shared/${SAPSYSTEMNAME}/HDB${TINSTANCE}/${HOSTNAME}/global.ini
# global.ini last modified 2023-04-06 16:15:03.521715 by hdbnameserver
[persistence]
log_mode = overwrite
```

```
# global.ini last modified 2023-04-06 16:15:03.521715 by hdbnameserver
[persistence]
log_mode = normal
```

检查或更新 global.ini 文件后，验证 log_mode 值：

```
clusternode1:rh2adm> hdbsql -d ${SAPSYSTEMNAME} -i ${TINSTANCE} -u SYSTEM;
hdbsql RH2=> select * from m_inifile_contents where section='persistence' and
key='log_mode'
FILE_NAME,LAYER_NAME,TENANT_NAME,HOST,SECTION,KEY,VALUE
"global.ini","DEFAULT","","","persistence","log_mode","normal"
"global.ini","HOST","","node2","persistence","log_mode","normal"
2 rows selected (overall time 60.982 msec; server time 20.420 msec)
```

部分还显示在 [persistence] 部分中需要设置此参数。当您将日志模式从 覆盖 改为 normal 时，建议您创建一个完整的数据备份，以确保数据库可以被恢复。

6.1.15. 发现主数据库

例如，有几种方法可以识别主节点：

- pcs status | grep Promoted
- hdbnsutil -sr_stateConfiguration
- systemReplicationStatus.py

选项 1 - 以下 systemReplicationStatus.py 脚本示例，过滤器将返回所有节点上的主数据库位置：

```
clusternode1:rh2adm>
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicationStatu
s.py --sapcontrol=1 | egrep -e
"3${TINSTANCE}01/HOST|PRIMARY_MASTERS"| head -1 | awk -F"=" '{ print $2 }'
```

输出：

```
clusternode2
```

选项 2 - 以下示例以类似方法为所有节点显示 systemReplicationStatus ：

```
rh2adm>hdbnsutil -sr_state --sapcontrol=1 | grep site.*Mode
```

输出：

```
siteReplicationMode/DC1=primary
siteReplicationMode/DC3=async
siteReplicationMode/DC2=syncmem
siteOperationMode/DC1=primary
siteOperationMode/DC3=logreplay
siteOperationMode/DC2=logreplay
```

6.1.16. 接管主

请参阅 检查复制状态 部分，以检查主节点和次要节点。另外：

- 将集群设置为 maintenance-mode
- 在辅助节点上启动接管

为集群启用 maintenance-mode 的示例：

```
[root@clusternode1]# pcs property set maintenance-mode=true
```

在成为新主的二级中，以 < sidadm> 用户身份运行：

```
clusternode1:rh2adm> hdbnsutil -sr_takeover
```

这个二级成为主要的，其他活跃二级数据库会重新注册到新主，需要手动重新注册为次要主。

6.1.17. 重新注册以前的主主作为辅助

请确定集群停止或置于 maintenance-mode 中。Example:

```
clusternode2:rh2adm> hdbnsutil -sr_register --remoteHost=remotehost3 --
remoteInstance=${TINSTANCE} --replicationMode=syncmem --name=DC2 --online --
remoteName=DC3 --operationMode=logreplay --force_full_replica --online
```

在我们的示例中，我们使用完整复制。需要完整复制时，您的 SAP HANA 系统管理员应知道。

6.1.18. 从故障切换中恢复

请参阅 检查 SAP HANA 系统复制状态 并发现 主要节点。信息一致非常重要。如果节点不是 systemReplicationStatus.py 输出的一部分，且具有不同的系统复制状态，如果需要重新注册此节点，请检查您的数据库管理员。

解决这种情况的一种方法是 重新注册 此站点作为新次要站点。

有时，辅助实例仍会没有启动。然后，在重新注册前取消注册此站点。取消注册二级 DC1 的示例：

```
clusternode1:rh2adm> hdbnsutil -sr_unregister --name=DC1
```

重新注册 DC1 的示例：

```
clusternode1:rh2adm> hdbnsutil -sr_register --name=DC1 --remoteHost=node2 --
remoteInstance=02 --replicationMode=sync --operationMode=logreplay --online
```

需要启动数据库 并检查 它 是否在运行。最后， 检查复制状态。

6.2. PACEMAKER 命令

6.2.1. 启动和停止集群

要启动所有节点上的集群，请执行以下命令：

```
# pcs cluster start -all
```

重启后，只有在启用该服务时，集群才会自动启动。命令有助于了解集群是否已启动，以及是否启用守护进程是否自动启动。

```
# pcs cluster status
```

集群自动启动可以通过以下方式启用：

```
# pcs cluster enable --all
```

其他选项有：

- 停止集群。
- 将节点设置为待机。
- 将集群设置为 maintenance-mode。

如需了解更多详细信息，请检查 pcs 集群 帮助：

```
# pcs cluster stop --all
# pcs cluster help
```

6.2.2. 将集群设置为 maintenance-mode

如果要进行更改，并且希望避免 pacemaker 集群的干扰，您可以通过将其置于 maintenance-mode 来"忽略"集群：

```
# pcs property set maintenance-mode=true
```

验证 maintenance-mode 的一种简单方法是检查资源是否是非受管状态：

```
# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02] (unmanaged):
    * SAPHanaTopology_RH2_02    (ocf:heartbeat:SAPHanaTopology):        Started
clusternode1 (unmanaged)
    * SAPHanaTopology_RH2_02    (ocf:heartbeat:SAPHanaTopology):        Started
clusternode2 (unmanaged)
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable, unmanaged):
    * SAPHana_RH2_02    (ocf:heartbeat:SAPHana):        Unpromoted clusternode1
(unmanaged)
    * SAPHana_RH2_02    (ocf:heartbeat:SAPHana):        Promoted clusternode2 (unmanaged)
  * vip_RH2_02_MASTER   (ocf:heartbeat:IPaddr2):        Started clusternode2 (unmanaged)
```

刷新集群资源以便在集群处于 maintenance-mode 时检测资源状态，且不更新资源状态更改：

```
# pcs resource refresh
```

这将指示任何内容是否尚未正确，并会在不使用 maintenance-mode 时立即导致补救操作。

运行以下命令来删除 maintenance-mode ：

```
# pcs property set maintenance-mode=false
```

现在，集群将继续正常工作。如果配置了错误，它将现在做出反应。

6.2.3. 检查集群状态

以下是检查集群状态的几种方法：

- 检查集群是否正在运行：

  ```
  # pcs cluster status
  ```

- 检查集群和所有资源：

  -

```
# pcs status
```

- 检查集群、所有资源和所有节点属性：

```
# pcs status --full
```

- 仅检查资源：

```
# pcs resource status --full
```

- 检查 Stonith 历史记录：

```
# pcs stonith history
```

- 检查位置限制：

```
# pcs constraint location
```

注意

必须配置和测试隔离。要获取尽可能自动化的解决方案，集群必须持续激活，然后让集群在重启后自动启动。在生产环境中，禁用重启允许手动干预，例如崩溃后实例的人工干预。还要检查守护进程状态。

Example:

```
# pcs status --full
Cluster name: cluster1
Status of pacemakerd: 'Pacemaker is running' (last updated 2023-06-22 17:56:01 +02:00)
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode2 (2) (version 2.1.5-7.el9-a3f44794f94) - partition with quorum
  * Last updated: Thu Jun 22 17:56:01 2023
  * Last change:  Thu Jun 22 17:53:34 2023 by root via crm_attribute on clusternode1
  * 2 nodes configured
  * 6 resource instances configured
Node List:
  * Node clusternode1 (1): online, feature set 3.16.2
  * Node clusternode2 (2): online, feature set 3.16.2
Full List of Resources:
  * h7fence (stonith:fence_rhevm):  Started clusternode2
```

```
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode1
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode2
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Promoted clusternode1
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Unpromoted clusternode2
  * vip_RH2_02_MASTER (ocf:heartbeat:IPaddr2):  Started clusternode1
Node Attributes:
  * Node: clusternode1 (1):
    * hana_rh2_clone_state            : PROMOTED
    * hana_rh2_op_mode                : logreplay
    * hana_rh2_remoteHost             : clusternode2
    * hana_rh2_roles                  : 4:P:master1:master:worker:master
    * hana_rh2_site                   : DC1
    * hana_rh2_sra                    : -
    * hana_rh2_srah                   : -
    * hana_rh2_srmode                 : syncmem
    * hana_rh2_sync_state             : PRIM
    * hana_rh2_version                : 2.00.059.02
    * hana_rh2_vhost                  : clusternode1
    * lpa_rh2_lpt                     : 1687449214
    * master-SAPHana_RH2_02           : 150
  * Node: clusternode2 (2):
    * hana_rh2_clone_state            : DEMOTED
    * hana_rh2_op_mode                : logreplay
    * hana_rh2_remoteHost             : clusternode1
    * hana_rh2_roles                  : 4:S:master1:master:worker:master
    * hana_rh2_site                   : DC2
    * hana_rh2_sra                    : -
    * hana_rh2_srah                   : -
    * hana_rh2_srmode                 : syncmem
    * hana_rh2_sync_state             : SOK
    * hana_rh2_version                : 2.00.059.02
    * hana_rh2_vhost                  : clusternode2
    * lpa_rh2_lpt                     : 30
    * master-SAPHana_RH2_02           : 100
Migration Summary:
Tickets:
PCSD Status:
  clusternode1: Online
  clusternode2: Online
Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

6.2.4. 检查资源状态

使用 pcs resource 检查所有资源的状态。这会输出列表和资源的当前状态。

Example:

```
# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * Started: [ clusternode1 clusternode2 ]
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * Promoted: [ clusternode1 ]
    * Unpromoted: [ clusternode2 ]
  * vip_RH2_02_MASTER (ocf:heartbeat:IPaddr2):  Started clusternode1
```

6.2.5. 检查资源配置

以下显示当前资源配置：

```
# pcs resource config
Resource: vip_RH2_02_MASTER (class=ocf provider=heartbeat type=IPaddr2)
  Attributes: vip_RH2_02_MASTER-instance_attributes
    ip=192.168.5.136
  Operations:
    monitor: vip_RH2_02_MASTER-monitor-interval-10s
      interval=10s
      timeout=20s
    start: vip_RH2_02_MASTER-start-interval-0s
      interval=0s
      timeout=20s
    stop: vip_RH2_02_MASTER-stop-interval-0s
      interval=0s
      timeout=20s
Clone: SAPHanaTopology_RH2_02-clone
  Meta Attributes: SAPHanaTopology_RH2_02-clone-meta_attributes
    clone-max=2
    clone-node-max=1
    interleave=true
  Resource: SAPHanaTopology_RH2_02 (class=ocf provider=heartbeat
type=SAPHanaTopology)
    Attributes: SAPHanaTopology_RH2_02-instance_attributes
      InstanceNumber=02
      SID=RH2
    Operations:
      methods: SAPHanaTopology_RH2_02-methods-interval-0s
        interval=0s
        timeout=5
      monitor: SAPHanaTopology_RH2_02-monitor-interval-10
        interval=10
        timeout=600
      reload: SAPHanaTopology_RH2_02-reload-interval-0s
        interval=0s
        timeout=5
      start: SAPHanaTopology_RH2_02-start-interval-0s
        interval=0s
        timeout=600
      stop: SAPHanaTopology_RH2_02-stop-interval-0s
        interval=0s
        timeout=600
Clone: SAPHana_RH2_02-clone
```

```
 Meta Attributes: SAPHana_RH2_02-clone-meta_attributes
  clone-max=2
  clone-node-max=1
  interleave=true
  notify=true
  promotable=true
 Resource: SAPHana_RH2_02 (class=ocf provider=heartbeat type=SAPHana)
  Attributes: SAPHana_RH2_02-instance_attributes
   AUTOMATED_REGISTER=true
   DUPLICATE_PRIMARY_TIMEOUT=300
   HANA_CALL_TIMEOUT=10
   InstanceNumber=02
   PREFER_SITE_TAKEOVER=true
   SID=RH2
  Operations:
   demote: SAPHana_RH2_02-demote-interval-0s
    interval=0s
    timeout=3600
   methods: SAPHana_RH2_02-methods-interval-0s
    interval=0s
    timeout=5
   monitor: SAPHana_RH2_02-monitor-interval-251
    interval=251
    timeout=700
    role=Unpromoted
   monitor: SAPHana_RH2_02-monitor-interval-249
    interval=249
    timeout=700
    role=Promoted
   promote: SAPHana_RH2_02-promote-interval-0s
    interval=0s
    timeout=3600
   reload: SAPHana_RH2_02-reload-interval-0s
    interval=0s
    timeout=5
   start: SAPHana_RH2_02-start-interval-0s
    interval=0s
    timeout=3200
   stop: SAPHana_RH2_02-stop-interval-0s
    interval=0s
    timeout=3100
```

这将列出用于配置安装和配置的资源代理的所有参数。

6.2.6. SAPHana 资源选项 AUTOMATED_REGISTER=true

如果在 SAPHana 资源中使用这个选项，则 pacemaker 将自动重新注册二级数据库。

建议您在第一次测试中使用这个选项。使用 AUTOMATED_REGISTER=false 时，管理员需要手动重新注册次要节点。

6.2.7. 资源处理

管理资源有几个选项。如需更多信息，请查看帮助信息：

```
# pcs resource help
```

列出使用的资源代理：

```
# pcs resource config | grep "type=" | awk -F"type=" '{ print $2 }' | sed -e "s/)//g"
```

输出示例：

```
IPaddr2
SAPHanaTopology
SAPHana
```

显示特定的资源代理描述和配置参数：

```
# pcs resource describe <resource agent>
```

示例（不带输出）：

```
# pcs resource describe IPaddr2
```

资源代理 IPaddr2 示例（带有输出）：

```
Assumed agent name 'ocf:heartbeat:IPaddr2' (deduced from 'IPaddr2')
ocf:heartbeat:IPaddr2 - Manages virtual IPv4 and IPv6 addresses (Linux specific version)

This Linux-specific resource manages IP alias IP addresses. It can add an IP alias, or remove
one. In
addition, it can implement Cluster Alias IP functionality if invoked as a clone resource.  If used
as a
clone, "shared address with a trivial, stateless (autonomous) load-balancing/mutual exclusion
on
ingress" mode gets applied (as opposed to "assume resource uniqueness" mode otherwise).
For that, Linux
firewall (kernel and userspace) is assumed, and since recent distributions are ambivalent in
plain
"iptables" command to particular back-end resolution, "iptables-legacy" (when present) gets
prioritized
so as to avoid incompatibilities (note that respective ipt_CLUSTERIP firewall extension in use
```

```
here is,
at the same time, marked deprecated, yet said "legacy" layer can make it workable, literally, to this
day) with "netfilter" one (as in "iptables-nft"). In that case, you should explicitly set clone-node-max
>= 2, and/or clone-max < number of nodes. In case of node failure, clone instances need to be re-
allocated on surviving nodes. This would not be possible if there is already an instance on those nodes,
and clone-node-max=1 (which is the default).  When the specified IP address gets assigned to a
respective interface, the resource agent sends unsolicited ARP (Address Resolution Protocol, IPv4) or NA
(Neighbor Advertisement, IPv6) packets to inform neighboring machines about the change. This
functionality is controlled for both IPv4 and IPv6 by shared 'arp_*' parameters.

Resource options:
  ip (required) (unique): The IPv4 (dotted quad notation) or IPv6 address (colon hexadecimal notation)
      example IPv4 "192.168.1.1". example IPv6 "2001:db8:DC28:0:0:FC57:D4C8:1FFF".
  nic: The base network interface on which the IP address will be brought online.  If left empty, the
      script will try and determine this from the routing table.  Do NOT specify an alias interface in
      the form eth0:1 or anything here; rather, specify the base interface only. If you want a label,
      see the iflabel parameter.  Prerequisite:  There must be at least one static IP address, which is
      not managed by the cluster, assigned to the network interface. If you can not assign any static IP
      address on the interface, modify this kernel parameter:  sysctl -w
      net.ipv4.conf.all.promote_secondaries=1 # (or per device)
  cidr_netmask: The netmask for the interface in CIDR format (e.g., 24 and not 255.255.255.0) If
      unspecified, the script will also try to determine this from the routing table.
  broadcast: Broadcast address associated with the IP. It is possible to use the special symbols '+' and
      '-' instead of the broadcast address. In this case, the broadcast address is derived by
      setting/resetting the host bits of the interface prefix.
  iflabel: You can specify an additional label for your IP address here. This label is appended to your
      interface name.  The kernel allows alphanumeric labels up to a maximum length of 15 characters
      including the interface name and colon (e.g. eth0:foobar1234)  A label can be specified in nic
      parameter but it is deprecated. If a label is specified in nic name, this parameter has no effect.
  lvs_support: Enable support for LVS Direct Routing configurations. In case a IP address is stopped,
      only move it to the loopback device to allow the local node to continue to service requests, but
      no longer advertise it on the network.  Notes for IPv6: It is not necessary to enable this option
      on IPv6. Instead, enable 'lvs_ipv6_addrlabel' option for LVS-DR usage on IPv6.
  lvs_ipv6_addrlabel: Enable adding IPv6 address label so IPv6 traffic originating from the
```

```
address's
     interface does not use this address as the source. This is necessary for LVS-DR health
checks to
     realservers to work. Without it, the most recently added IPv6 address (probably the
address added
     by IPaddr2) will be used as the source address for IPv6 traffic from that interface and
since that
     address exists on loopback on the realservers, the realserver response to
pings/connections will
     never leave its loopback. See RFC3484 for the detail of the source address selection.  See
also
     'lvs_ipv6_addrlabel_value' parameter.
  lvs_ipv6_addrlabel_value: Specify IPv6 address label value used when 'lvs_ipv6_addrlabel' is
enabled.
     The value should be an unused label in the policy table which is shown by 'ip addrlabel
list'
     command. You would rarely need to change this parameter.
  mac: Set the interface MAC address explicitly. Currently only used in case of the Cluster IP
Alias.
     Leave empty to chose automatically.
  clusterip_hash: Specify the hashing algorithm used for the Cluster IP functionality.
  unique_clone_address: If true, add the clone ID to the supplied value of IP to create a unique
address
     to manage
  arp_interval: Specify the interval between unsolicited ARP (IPv4) or NA (IPv6) packets in
     milliseconds.  This parameter is deprecated and used for the backward compatibility only.
It is
     effective only for the send_arp binary which is built with libnet, and send_ua for IPv6. It
has no
     effect for other arp_sender.
  arp_count: Number of unsolicited ARP (IPv4) or NA (IPv6) packets to send at resource
initialization.
  arp_count_refresh: For IPv4, number of unsolicited ARP packets to send during resource
monitoring.
     Doing so helps mitigate issues of stuck ARP caches resulting from split-brain situations.
  arp_bg: Whether or not to send the ARP (IPv4) or NA (IPv6) packets in the background. The
default is
     true for IPv4 and false for IPv6.
  arp_sender: For IPv4, the program to send ARP packets with on start. Available options are:
-
     send_arp: default  - ipoibarping: default for infiniband interfaces if ipoibarping is available
-
     iputils_arping: use arping in iputils package  - libnet_arping: use another variant of arping
     based on libnet
  send_arp_opts: For IPv4, extra options to pass to the arp_sender program. Available options
are vary
     depending on which arp_sender is used.  A typical use case is specifying '-A' for
iputils_arping
     to use ARP REPLY instead of ARP REQUEST as Gratuitous ARPs.
  flush_routes: Flush the routing table on stop. This is for applications which use the cluster
IP
     address and which run on the same physical host that the IP address lives on. The Linux
kernel may
     force that application to take a shortcut to the local loopback interface, instead of the
     interface the address is really bound to. Under those circumstances, an application may,
somewhat
```

```
    unexpectedly, continue to use connections for some time even after the IP address is
deconfigured.
    Set this parameter in order to immediately disable said shortcut when the IP address goes
away.
  run_arping: For IPv4, whether or not to run arping for collision detection check.
  nodad: For IPv6, do not perform Duplicate Address Detection when adding the address.
  noprefixroute: Use noprefixroute flag (see 'man ip-address').
  preferred_lft: For IPv6, set the preferred lifetime of the IP address. This can be used to
ensure that
    the created IP address will not be used as a source address for routing. Expects a value as
    specified in section 5.5.4 of RFC 4862.
  network_namespace: Specifies the network namespace to operate within. The namespace
must already
    exist, and the interface to be used must be within the namespace.

Default operations:
  start:
    interval=0s
    timeout=20s
  stop:
    interval=0s
    timeout=20s
  monitor:
    interval=10s
    timeout=20s
```

如果集群停止，则所有资源也会停止；如果集群 处于维护模式，则所有资源将保持运行，但不会受监控或管理。

6.2.8. 集群属性处理，用于 maintenance-mode

列出所有定义的属性：

```
[root@clusternode1] pcs property
Cluster Properties:
 cluster-infrastructure: corosync
 cluster-name: cluster1
 concurrent-fencing: true
 dc-version: 2.1.5-7.el9-a3f44794f94
 hana_rh2_site_srHook_DC1: PRIM
 hana_rh2_site_srHook_DC2: SFAIL
 have-watchdog: false
 last-lrm-refresh: 1688548036
 maintenance-mode: true
 priority-fencing-delay: 10s
 stonith-enabled: true
 stonith-timeout: 900
```

要重新配置数据库，必须指示集群忽略任何更改，直到配置完成为止。您可以使用以下方法将集群 置于维护模式 ：

```
# pcs property set maintenance-mode=true
```

检查 **maintenance-mode** ：

```
# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02] (unmanaged):
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode1
(unmanaged)
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode2
(unmanaged)
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable, unmanaged):
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Promoted clusternode1 (unmanaged)
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Unpromoted clusternode2 (unmanaged)
  * vip_RH2_02_MASTER (ocf:heartbeat:IPaddr2):  Started clusternode1 (unmanaged)
```

验证所有资源都是 **"unmanaged"** ：

```
[root@clusternode1]# pcs status
Cluster name: cluster1
Status of pacemakerd: 'Pacemaker is running' (last updated 2023-06-27 16:02:15 +02:00)
Cluster Summary:
  * Stack: corosync
  * Current DC: clusternode2 (version 2.1.5-7.el9-a3f44794f94) - partition with quorum
  * Last updated: Tue Jun 27 16:02:16 2023
  * Last change:  Tue Jun 27 16:02:14 2023 by root via cibadmin on clusternode1
  * 2 nodes configured
  * 6 resource instances configured

              *** Resource management is DISABLED ***
  The cluster will not attempt to start, stop or recover services

Node List:
  * Online: [ clusternode1 clusternode2 ]

Full List of Resources:
  * h7fence (stonith:fence_rhevm):  Started clusternode2 (unmanaged)
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02] (unmanaged):
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode1
(unmanaged)
    * SAPHanaTopology_RH2_02 (ocf:heartbeat:SAPHanaTopology):  Started clusternode2
(unmanaged)
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable, unmanaged):
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Promoted clusternode1 (unmanaged)
    * SAPHana_RH2_02 (ocf:heartbeat:SAPHana):  Unpromoted clusternode2 (unmanaged)
  * vip_RH2_02_MASTER (ocf:heartbeat:IPaddr2):  Started clusternode1 (unmanaged)

Daemon Status:
```

```
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

如果您取消设置 maintenance-mode，则资源将切回到受管：

```
# pcs property set maintenance-mode=false
```

6.2.9. 使用 Move 故障转移 SAPHana 资源

有关如何故障转移 SAP HANA 数据库的简单示例是使用 pcs resource move 命令。您需要使用克隆资源名称并移动资源，如下所示：

```
# pcs resource move <SAPHana-clone-resource>
```

在本例中，克隆资源是 SAPHana_RH2_02-clone ：

```
[root@clusternode1]# pcs resource
  * Clone Set: SAPHanaTopology_RH2_02-clone [SAPHanaTopology_RH2_02]:
    * Started: [ clusternode1 clusternode2 ]
  * Clone Set: SAPHana_RH2_02-clone [SAPHana_RH2_02] (promotable):
    * Promoted: [ clusternode1 ]
    * Unpromoted: [ clusternode2 ]
  * vip_RH2_02_MASTER (ocf:heartbeat:IPaddr2):  Started clusternode1
```

移动资源：

```
# pcs resource move SAPHana_RH2_02-clone
Location constraint to move resource 'SAPHana_RH2_02-clone' has been created
Waiting for the cluster to apply configuration changes...
Location constraint created to move resource 'SAPHana_RH2_02-clone' has been removed
Waiting for the cluster to apply configuration changes...
resource 'SAPHana_RH2_02-clone' is promoted on node 'clusternode2'; unpromoted on node
'clusternode1'
```

检查是否有剩余的限制：

```
# pcs constraint location
```

您可以通过清除资源来删除在故障切换过程中创建的位置限制。Example:

```
[root@clusternode1]# pcs resource clear SAPHana_RH2_02-clone
```

检查 "Migration Summary" 中是否存在剩余的警告或条目：

```
# pcs status --full
```

检查 stonith 历史记录：

```
# pcs stonith history
```

如果需要，清除 stonith 历史记录：

```
# pcs stonith history cleanup
```

如果您使用早于 2.1.5 的 pacemaker 版本，请参阅 运行 pcs resource move 时是否存在管理限制的方法？ 并检查剩余的限制。

6.2.10. 监控故障切换和同步状态

所有 pacemaker 活动都记录在集群节点上的 /var/log/messages 文件中。由于还有许多其他消息，有时很难阅读与 SAP 资源代理相关的消息。您可以配置命令别名，仅过滤与 SAP 资源代理相关的消息。

别名 tmsl 示例：

```
# alias tmsl='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register|WAITING4LPA|PROMOTED
|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED|LPT"'
```

tsml 的输出示例：

```
[root@clusternode1]# tmsl
Jun 22 13:59:54 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC: secondary
with sync status SOK ==> possible takeover node
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
```

```
hana_rh2_sync_state=SOK
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC:
saphana_monitor_secondary: scoring_crm_master(4:S:master1:master:worker:master,SOK)
Jun 22 13:59:55 clusternode1 SAPHana(SAPHana_RH2_02)[907482]: INFO: DEC:
scoring_crm_master: sync(SOK) is matching syncPattern (SOK)
Jun 22 14:04:06 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:04:06 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:04:06 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC: secondary
with sync status SOK ==> possible takeover node
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
saphana_monitor_secondary: scoring_crm_master(4:S:master1:master:worker:master,SOK)
Jun 22 14:04:09 clusternode1 SAPHana(SAPHana_RH2_02)[914625]: INFO: DEC:
scoring_crm_master: sync(SOK) is matching syncPattern (SOK)
Jun 22 14:08:21 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:08:21 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:08:21 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:08:23 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:08:23 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:08:23 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC: secondary
with sync status SOK ==> possible takeover node
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC: Finally
```

```
get_SRHOOK()=SOK
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
saphana_monitor_secondary: scoring_crm_master(4:S:master1:master:worker:master,SOK)
Jun 22 14:08:24 clusternode1 SAPHana(SAPHana_RH2_02)[922136]: INFO: DEC:
scoring_crm_master: sync(SOK) is matching syncPattern (SOK)
Jun 22 14:12:35 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:12:35 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:12:36 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:12:38 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:12:38 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:12:38 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:12:38 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC: secondary
with sync status SOK ==> possible takeover node
Jun 22 14:12:39 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:12:39 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:12:39 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:12:39 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
saphana_monitor_secondary: scoring_crm_master(4:S:master1:master:worker:master,SOK)
Jun 22 14:12:39 clusternode1 SAPHana(SAPHana_RH2_02)[929408]: INFO: DEC:
scoring_crm_master: sync(SOK) is matching syncPattern (SOK)
Jun 22 14:14:01 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_clone_state[clusternode2]: PROMOTED -> DEMOTED
Jun 22 14:14:02 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_clone_state[clusternode2]: DEMOTED -> UNDEFINED
Jun 22 14:14:19 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_clone_state[clusternode1]: DEMOTED -> PROMOTED
Jun 22 14:14:21 clusternode1 SAPHana(SAPHana_RH2_02)[932762]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:14:21 clusternode1 SAPHana(SAPHana_RH2_02)[932762]: INFO: DEC:
hana_rh2_site_srHook_DC1 is empty or SWAIT. Take polling attribute:
hana_rh2_sync_state=SOK
Jun 22 14:14:21 clusternode1 SAPHana(SAPHana_RH2_02)[932762]: INFO: DEC: Finally
get_SRHOOK()=SOK
Jun 22 14:15:14 clusternode1 SAPHana(SAPHana_RH2_02)[932762]: INFO: DEC:
hana_rh2_site_srHook_DC1=SWAIT
Jun 22 14:15:22 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_sync_state[clusternode1]: SOK -> PRIM
Jun 22 14:15:23 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_sync_state[clusternode2]: PRIM -> SOK
Jun 22 14:15:23 clusternode1 SAPHana(SAPHana_RH2_02)[934810]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:15:25 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_clone_state[clusternode2]: UNDEFINED -> DEMOTED
Jun 22 14:15:32 clusternode1 pacemaker-attrd[10150]: notice: Setting
```

```
hana_rh2_sync_state[clusternode2]: SOK -> SFAIL
Jun 22 14:19:36 clusternode1 pacemaker-attrd[10150]: notice: Setting
hana_rh2_sync_state[clusternode2]: SFAIL -> SOK
Jun 22 14:19:36 clusternode1 SAPHana(SAPHana_RH2_02)[942693]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:23:49 clusternode1 SAPHana(SAPHana_RH2_02)[950623]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:28:02 clusternode1 SAPHana(SAPHana_RH2_02)[958633]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:32:15 clusternode1 SAPHana(SAPHana_RH2_02)[966683]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:36:27 clusternode1 SAPHana(SAPHana_RH2_02)[974736]: INFO: ACT site=DC1,
setting SOK for secondary (1)
Jun 22 14:40:40 clusternode1 SAPHana(SAPHana_RH2_02)[982934]: INFO: ACT site=DC1,
setting SOK for secondary (1)
```

通过过滤器，可以更轻松地了解正在发生哪些状态更改。如果缺少详细信息，您可以打开整个消息文件来读取所有信息。

故障转移后，您可以清除资源。另请检查是否有剩余位置限制。

6.2.11. 检查集群一致性

在安装过程中，资源有时会在配置最终完成前启动。这可能导致 Cluster Information Base (CIB)中的条目，这可能会导致行为不正确。这可以轻松检查，并在配置完成后手动更正。

如果您启动 SAPHana 资源，则会重新创建缺少的条目。pcs 命令无法解决错误的条目，需要手动删除。

检查 CIB 条目：

```
# cibadmin --query
```

DC3 和 SFAIL 是不应存在于 Cluster Information Base、当群集成员为 DC1 和 DC2 以及节点之间的同步状态报告为 SOK 的条目。

检查对应条目的示例：

```
# cibadmin --query |grep '"DC3"'
# cibadmin --query |grep '"SFAIL"'
```

该命令可以作为 root 用户在集群中的任何节点上执行。通常，命令的输出为空。如果配置中仍然出现错误，输出可能会类似如下：

```
      <nvpair id="SAPHanaSR-hana_rh1_glob_sec" name="hana_rh1_glob_sec"
value="DC3"/>
```

使用以下命令可以删除这些条目：

```
# cibadmin --delete --xml-text '<...>'
```

要删除上例中的条目，您必须输入以下内容：请注意，输出包含双引号，因此文本必须嵌入到单引号中：

```
# cibadmin --delete --xml-text '       <nvpair id="SAPHanaSR-hana_rh1_glob_sec"
name="hana_rh1_glob_sec" value="DC3"/>'
```

验证没有删除的 CIB 条目。返回的输出应为空。

```
# cibadmin --query |grep 'DC3"'
```

6.2.12. 集群清理

在故障转移测试过程中，可能留下在限制后，其他仍然保留在以前的测试中。在启动下一个测试前，需要从它们清除集群。

检查失败事件的集群状态：

```
# pcs status --full
```

如果您在 "Migration Summary" 中看到集群警告或条目，您应该清除并清理资源：

```
# pcs resource clear SAPHana_RH2_02-clone
# pcs resource cleanup SAPHana_RH2_02-clone
```

输出：

```
Cleaned up SAPHana_RH2_02:0 on clusternode1
Cleaned up SAPHana_RH2_02:1 on clusternode2
```

检查是否有不需要的位置限制，例如来自以前的故障切换：

```
# pcs constraint location
```

更详细地检查现有限制：

```
# pcs constraint --full
```

资源移动后位置约束示例：

```
    Node: hana08 (score:-INFINITY) (role:Started) (id:cli-ban-SAPHana_RH2_02-clone-on-
hana08)
```

清除此位置约束：

```
# pcs resource clear SAPHana_RH2_02-clone
```

验证约束是否已从约束列表中显示。如果保留，则使用其约束 id 显式删除它：

```
# pcs constraint delete cli-ban-SAPHana_RH2_02-clone-on-hana08
```

如果您使用隔离运行多个测试，您可能也清除 stonith 历史记录：

```
# pcs stonith history cleanup
```

所有 pcs 命令都是以 root 用户身份执行的。另外，请检查 发现左侧。

6.2.13. 其他集群命令

各种集群命令示例

```
# pcs status --full
# crm_mon -1Arf # Provides an overview
```

```
# pcs resource # Lists all resources and shows if they are running
# pcs constraint --full # Lists all constraint ids which should be removed
# pcs cluster start --all # This will start the cluster on all nodes
# pcs cluster stop --all # This will stop the cluster on all nodes
# pcs node attribute # Lists node attributes
```

6.3. RHEL 和常规命令

6.3.1. 发现当前状态

您必须按照以下步骤了解环境的当前状态。请参阅监控 环境。另外，我们建议您进行以下操作：

- 检查 /var/log/messages，使用 Aliases 来监控 日志以方便日志检查。
- 有时，集群必须从以前的活动中清理才能继续正常工作。发现 左侧并在需要时清除它们。

6.3.2. yum info

```
# yum info resource-agents-sap-hana
Last metadata expiration check: 2:47:28 ago on Tue 06 Jun 2023 03:13:57 AM CEST.
Installed Packages
Name         : resource-agents-sap-hana
Epoch        : 1
Version      : 0.162.1
Release      : 2.el9_2
Architecture : noarch
Size         : 174 k
Source       : resource-agents-sap-hana-0.162.1-2.el9_2.src.rpm
Repository   : @System
Summary      : SAP HANA cluster resource agents
URL          : https://github.com/SUSE/SAPHanaSR
License      : GPLv2+
Description  : The SAP HANA resource agents interface with Pacemaker to allow
             : SAP instances to be managed in a cluster environment.
```

6.3.3. RPM 显示版本

```
# rpm -q resource-agents-sap-hana
resource-agents-sap-hana-0.162.1-2.el9_2.noarch
```

6.3.4. 监控的别名

您可以将其添加到 shell 配置集中。在示例中，根别名依赖于 < sid>adm 别名，因此还必须定义它。

- root （在 ~/.bashrc中添加）：

```
# export ListInstances=$(/usr/sap/hostctrl/exe/saphostctrl -function ListInstances|
head -1 )
export sid=$(echo "$ListInstances" |cut -d " " -f 5| tr [A-Z] [a-z])
export SID=$(echo $sid | tr [a-z] [A-Z])
export Instance=$(echo "$ListInstances" |cut -d " " -f 7 )
alias crmm='watch -n 1 crm_mon -1Arf'
alias crmv='watch -n 1 /usr/local/bin/crmmv'
alias cglo='su - ${sid}adm -c cglo'
alias cdh='cd /usr/lib/ocf/resource.d/heartbeat'
alias gtr='su - ${sid}adm  -c gtr'
alias hdb='su - ${sid}adm  -c hdb'
alias hdbi='su - ${sid}adm  -c hdbi'
alias hgrep='history | grep $1'
alias hri='su - ${sid}adm  -c hri'
alias hris='su - ${sid}adm  -c hris'
alias killnode="echo 'b' > /proc/sysrq-trigger"
alias lhc='su - ${sid}adm  -c lhc'
alias pit='ssh pitunnel'
alias python='/usr/sap/${SID}/HDB${Instance}/exe/Python/bin/python'
alias srstate='su - ${sid}adm  -c srstate'
alias shr='watch -n 5 "SAPHanaSR-monitor --sid=${SID}"'
alias sgsi='su - ${sid}adm  -c sgsi'
alias srm='su - ${sid}adm  -c srm'
alias srs='su - ${sid}adm  -c srs'
alias sapstart='su - ${sid}adm  -c sapstart'
alias sapstop='su - ${sid}adm  -c sapstop'
alias tma='tmux attach -t `tmux ls | grep -v atta| head -1 |cut -d " " -f 1`'
alias tm='tail -100f /var/log/messages |grep -v systemd'
alias tms='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_${SID}_HDB${Instance}|sr_register|WAITING4
LPA|EXCLUDE as possible takeover
node|SAPHanaSR|failed|${HOSTNAME}|PROMOTED|DEMOTED|UNDEFINED|master_w
alk|SWAIT|WaitforStop
ped|FAILED"'
alias tmss='tail -1000f /var/log/messages | grep -v systemd| egrep -s "secondary with
sync status|Setting master-rsc_SAPHa
na_${SID}_HDB${Instance}|sr_register|WAITING4LPA|EXCLUDE as possible takeover
node|SAPHanaSR|failed|${HOSTNAME}|PROMOTED|DE
MOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED"'
alias tmm='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_${SID}_HDB${Instance}|sr_register|WAITING4
LPA|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED|
LPT|SOK|SFAIL|SAPHanaSR-mon"| grep -v systemd'
alias tmsl='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_${SID}_HDB${Instance}|sr_register|WAITING
4LPA|PROMOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILE
D|LPT|SOK|SFAIL|SAPHanaSR-mon"'
alias vih='vim /usr/lib/ocf/resource.d/heartbeat/SAPHanaStart'
alias vglo='su - ${sid}adm  -c vglo'
```

-

- **<SID>adm （添加到 ~/.customer.sh）：**

```
alias tm='tail -100f /var/log/messages |grep -v systemd'
alias tms='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register|WAITING4LPA|EXC
LUDE as possible takeover
node|SAPHanaSR|failed|${HOSTNAME}|PROMOTED|DEMOTED|UNDEFINED|master_w
alk|SWAIT|WaitforStopped|FAILED"'
alias tmsl='tail -1000f /var/log/messages | egrep -s "Setting master-
rsc_SAPHana_$SAPSYSTEMNAME_HDB${TINSTANCE}|sr_register|WAITING4LPA|PRO
MOTED|DEMOTED|UNDEFINED|master_walk|SWAIT|WaitforStopped|FAILED|LPT"'
alias sapstart='sapcontrol -nr ${TINSTANCE} -function StartSystem HDB;hdbi'
alias sapstop='sapcontrol -nr ${TINSTANCE} -function StopSystem HDB;hdbi'
alias sgsi='watch sapcontrol -nr ${TINSTANCE} -function GetSystemInstanceList'
alias spl='watch sapcontrol -nr ${TINSTANCE} -function GetProcessList'
alias splh='watch "sapcontrol -nr ${TINSTANCE} -function GetProcessList| grep
hdbdaemon"'
alias srm='watch "hdbnsutil -sr_state --sapcontrol=1 |grep site.*Mode"'
alias srs="watch -n 5 'python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/systemReplicati
onStatus.py ; echo Status \$?'"
alias srstate='watch -n 10 hdbnsutil -sr_state'
alias hdb='watch -n 5 "sapcontrol -nr ${TINSTANCE} -function GetProcessList| egrep -
s hdbdaemon\|hdbnameserver\|hdbindexserver "'
alias hdbi='watch -n 5 "sapcontrol -nr ${TINSTANCE} -function GetProcessList| egrep -
s hdbdaemon\|hdbnameserver\|hdbindexserver;sapcontrol -nr ${TINSTANCE} -
function GetSystemInstanceList "'
alias hgrep='history | grep $1'
alias vglo="vim /usr/sap/$SAPSYSTEMNAME/SYS/global/hdb/custom/config/global.ini"
alias vgloh="vim
/hana/shared/${SAPSYSTEMNAME}/HDB${TINSTANCE}/${HOSTNAME}/global.ini"
alias hri='hdbcons -e hdbindexserver "replication info"'
alias hris='hdbcons -e hdbindexserver "replication info" | egrep -e
"SiteID|ReplicationStatus_"'
alias gtr='watch -n 10
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/getTakeoverRec
ommendation.py --sapcontrol=1'
alias lhc='/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/Python/bin/python
/usr/sap/$SAPSYSTEMNAME/HDB${TINSTANCE}/exe/python_support/landscapeHostC
onfiguration.py;echo $?'
```

第 7 章 参考

7.1. RED HAT

- RHEL 高可用性集群的支持策略 - 在集群中管理 SAP HANA
- 使用 RHEL HA 附加组件自动化 SAP HANA 扩展系统复制
- 因为失败而移动资源
- 运行 pcs resource move 时是否有管理约束的方法？

7.2. SAP

- SAP HANA 平台的 SAP HANA 管理指南
- 多目标系统复制的灾难恢复场景
- SAP HANA 系统复制配置参数
- 示例：检查 Primary 和 Secondary 系统上的状态
- 配置 SAP HANA 系统复制的一般先决条件
- 更改日志模式
- 缺少日志，将之前的主站点重新注册为新的二级站点失败
- 检查 Status with landscapeHostConfiguration.py

- 如何设置 SAP HANA 多文件系统复制
- SAP HANA Multitarget System Replication