



# Red Hat Enterprise Linux for SAP Solutions 9

RHEL HA Add-On を使用した SAP HANA スケーラブルアップシステムレプリケーションの自動化



# Red Hat Enterprise Linux for SAP Solutions 9 RHEL HA Add-On を使用した SAP HANA スケールアップシステムレプリケーションの自動化

---

## 法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## 概要

このガイドでは、サポート対象の RHEL リリース上の Pacemaker クラスターのスケールアップで自動化された HANA システムレプリケーションを設定する方法について説明します。

## 目次

多様性を受け入れるオープンソースの強化 .....	3
RED HAT ドキュメントへのフィードバック (英語のみ) .....	4
<b>第1章 概要</b> .....	<b>5</b>
1.1. サポートポリシー .....	6
1.2. 必要なサブスクリプションとリポジトリ .....	6
<b>第2章 SAP HANA システムレプリケーションの設定</b> .....	<b>7</b>
2.1. 前提条件 .....	7
2.2. SAP HANA データベースの初期バックアップの実行 .....	8
2.3. SAP HANA プライマリーレプリケーションインスタンスの設定 .....	8
2.4. SAP HANA セカンダリーレプリケーションインスタンスの設定 .....	9
2.5. SAP HANA システムレプリケーションの状態の確認 .....	10
2.6. SAP HANA システムレプリケーションのテスト .....	11
<b>第3章 SAP HANA スケールアップシステムレプリケーションセットアップを管理するための HA クラスターの設定</b> .....	<b>12</b>
3.1. RHEL HA アドオンを使用した SAP HANA スケールアップシステムレプリケーションの管理に必要なリソース エージェントとその他のコンポーネントのインストール .....	12
3.2. SAP HANA SRCONNECTIONCHANGED() フックの有効化 .....	12
3.3. 一般的な HA クラスタープロパティの設定 .....	14
3.4. クローンされた SAPHANATOPOLOGY リソースの作成 .....	15
3.5. プロモーション可能な SAPHANA リソースの作成 .....	16
3.6. 仮想 IP アドレスリソースの作成 .....	20
3.7. 制約の作成 .....	20
3.8. ACTIVE/ACTIVE (READ-ENABLED) SAP HANA システムレプリケーションセットアップ用のセカンダリー仮 想 IP アドレスの追加 (オプション) .....	21
3.9. HDBINDEXSERVER プロセス失敗アクション (オプション) に対する SAP HANA SRSERVICESTATECHANGED() フックの有効化 .....	22
<b>第4章 セットアップのテスト</b> .....	<b>25</b>
<b>第5章 メンテナンス手順</b> .....	<b>26</b>
5.1. OS と HA クラスターコンポーネントの更新 .....	26
5.2. SAP HANA インスタンスの更新 .....	26
5.3. SAPHANA リソースの別のノードへの手動移動 (HA クラスターによる SAP HANA システムレプリケーション のテイクオーバー) .....	26
<b>第6章 参考資料</b> .....	<b>28</b>
6.1. RED HAT .....	28
6.2. SAP .....	28
6.3. その他 .....	28



## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメントにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。多様性を受け入れる用語に変更する取り組みの詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。

## RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

### Jira からのフィードバック送信 (アカウントが必要)

1. [Jira](#) の Web サイトにログインしていることを確認してください。
2. [こちらのリンク](#) をクリックして、フィードバックをお寄せください。
3. **Summary** フィールドにわかりやすいタイトルを入力します。
4. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
5. 今後の更新に関する通知を受け取りたい場合は、**Reporter** としてご自身が割り当てられていることを確認してください。
6. ダイアログの下部にある **Create** をクリックします。

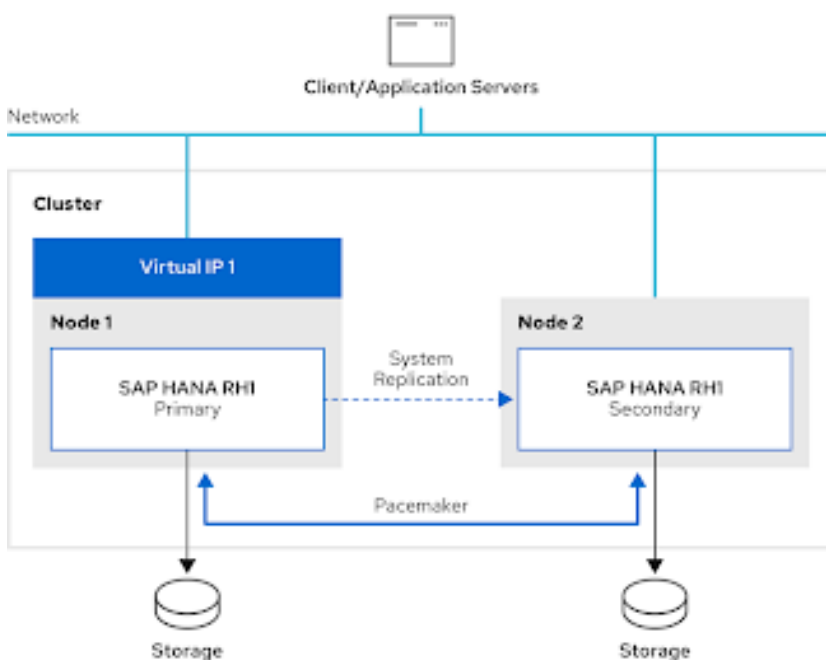


## 第1章 概要

このドキュメントでは、RHEL 9 で [RHEL HA Add-On](#) を使用して HA クラスタをセットアップし、'performance-optimized' SAP HANA スケールアップシステムレプリケーションのセットアップを自動化する方法について説明します。

'performance-optimized' とは、各ノード上のほとんどのリソース (CPU、RAM) を制御できる SAP HANA インスタンスが各ノード上で1つだけ実行されていることを意味します。これは、SAP HANA インスタンスが可能な限り高いパフォーマンスで実行できることを意味します。このシナリオでは、セカンダリー SAP HANA インスタンスがすべてのデータを事前にロードするように設定されているため、プライマリー SAP HANA インスタンスに障害が発生した場合のテイクオーバーは迅速に行われるはずですが。

次の図は、セットアップの概要を示しています。



2024 RHEL 9.2.2

'performance-optimized' SAP HANA システムレプリケーションセットアップでは、[Active/Active \(Read Enabled\)](#) SAP HANA システムレプリケーション設定を使用することもできます。これにより、セカンダリー SAP HANA インスタンス上のクライアントに read-only アクセスが許可されます。このドキュメントでは、'performance-optimized' SAP HANA スケールアップシステムレプリケーションを管理するための基本的なセットアップに加えて、[Active/Active \(Read Enabled\)](#) SAP HANA スケールアップシステムレプリケーション設定の管理に必要な追加のクラスタ設定に関するオプションの手順も提供します。

このドキュメントで説明されているセットアップに使用されるリソースエージェントとクラスタ設定は、[SAP Note 2063657 - SAP HANA System Replication Takeover Decision Guideline](#) で SAP が提供するガイドラインに基づいて開発されました。

このドキュメントでは、SAP HANA を実行するための RHEL 9 のインストールと設定、または SAP HANA のインストール手順については説明しません。各 HA クラスタノードで SAP HANA を実行するように RHEL 9 をインストールして設定する方法の詳細は、[RHEL 9 for SAP HANA2 インストールの設定](#) を参照してください。また、SAP HANA インスタンスのインストールについては、[SAP HANA Installation guide](#) およびハードウェアベンダー/クラウドプロバイダーのガイドラインを参照してください。

このドキュメントで説明されているセットアップは、オンプレミスの 'bare-metal' サーバーを使用して

行われました。AWS、Azure、GCP などのパブリッククラウド環境でこのようなセットアップの使用を予定する場合は、特定のプラットフォームのドキュメントを確認してください: ['performance optimized' SAP HANA スケールアップシステムレプリケーションの HA sソリューション - 設定ガイド](#)

## 1.1. サポートポリシー

[RHEL 高可用性クラスターのサポートポリシー - クラスター内の SAP HANA の管理](#) を参照してください。

## 1.2. 必要なサブスクリプションとリポジトリ

SAP Note [3108302 - SAP HANA DB: Recommended OS Settings for RHEL 9](#) に記載されているように、SAP HANA を実行するすべての RHEL 9 システムには [RHEL for SAP Solutions](#) サブスクリプションが必要です。RHEL 9 で SAP HANA を実行するための標準リポジトリに加えて、すべての HA クラスターノードで [RHEL HA Add-On](#) のリポジトリも有効にする必要があります。有効なリポジトリのリストは次のようになります。

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

各 HA クラスターノードで正しいサブスクリプションとリポジトリが有効になっていることを確認する方法の詳細は、[RHEL for SAP サブスクリプションおよびリポジトリ](#) を参照してください。

## 第2章 SAP HANA システムレプリケーションの設定

HA クラスターを設定する前に、SAP のガイドライン [SAP HANA System Replication: Configuration](#) に従って、SAP HANA システムレプリケーションを設定し、テストする必要があります。

次の例は、後で SAP HANA システムレプリケーションのセットアップを管理する HA クラスターの一部となるノードで SAP HANA システムレプリケーションを有効にする方法を示しています。

各 HA クラスターノードで正しいサブスクリプションとリポジトリが有効になっていることを確認する方法の詳細は、[RHEL for SAP サブスクリプションおよびリポジトリ](#) を参照してください。

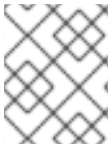
例で使用される SAP HANA 設定は、以下のとおりです。

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

### 2.1. 前提条件

両方のシステムが、問題なく両方のシステムの FQDN を解決できることを確認します。DNS がなくても FQDN を解決できるようにするには、以下の例のように FQDN を **/etc/hosts** に配置します。

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



#### 注記

[hostname](#) | [SAP Help Portal](#) に記載されているように、SAP HANA は、小文字のホスト名のみをサポートします。

システムレプリケーションを機能させるには、SAP HANA `log_mode` 変数を `normal` に設定する必要があります。これはデフォルト値でもあります。詳細は、[SAP Note 3221437 - System replication is failed due to "Connection refused: Primary has to run in log mode normal for system replication!"](#) を参照してください。これは、両方のノードで以下のコマンドを使用して、SAP HANA 管理ユーザーとして確認できます。

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

設定手順の多くは、インストール中に選択された SID の SAP HANA 管理ユーザーによって実行されます。このドキュメントで説明されているセットアップ例では、使用される SID が **RH1** であるため、ユーザー ID **rh1adm** が SAP HANA 管理ユーザーとして使用されます。

root ユーザーから SAP HANA 管理ユーザーに切り替えるには、次のコマンドを使用できます。

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

## 2.2. SAP HANA データベースの初期バックアップの実行

SAP HANA システムレプリケーションは、SAP HANA システムレプリケーションセットアップのプライマリーインスタンスとなる HANA インスタンスで初期バックアップが実行された後にのみ機能します。

以下に、**/tmp/foo** ディレクトリーに初期バックアップを作成する例を示します。

バックアップのサイズはデータベースのサイズによって異なるため、完了までに時間がかかる場合があります。バックアップが配置されるディレクトリーは、SAP HANA 管理ユーザーが書き込み可能である必要があります。

シングルテナントの SAP HANA セットアップでは、次のコマンドを使用して初期バックアップを作成できます。

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

マルチテナントの SAP HANA セットアップでは、**SYSTEMDB** とすべてのテナントデータベースをバックアップする必要があります。次の例は、**SYSTEMDB** のバックアップ方法を示しています。

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

テナントデータベースのバックアップ方法については、SAP HANA のドキュメントを確認してください。

## 2.3. SAP HANA プライマリーレプリケーションインスタンスの設定

初期バックアップが正常に完了したら、次のコマンドを使用して SAP HANA システムレプリケーションを初期化します。

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

初期化後、SAP HANA システムレプリケーションステータスに現在のノードが 'プライマリー' として表示されていることを確認します。

```
[rh1adm]#$ hdbnsutil -sr_state
checking for active or inactive nameserver ...
System Replication State
~~~~~
mode: primary
```

```
site id: 1
site name: DC1
Host Mappings:
```

## 2.4. SAP HANA セカンダリーレプリケーションインスタンスの設定

SAP HANA プライマリーインスタンスと同じ SID とインスタンス番号を使用して、他の HA クラスターノードにセカンダリー SAP HANA インスタンスをインストールした後、すでに実行中の SAP HANA プライマリーインスタンスに登録する必要があります。

セカンダリーレプリケーションインスタンスとなる SAP HANA インスタンスは、プライマリーインスタンスに登録できるようになる前に、まず停止する必要があります。

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

セカンダリー SAP HANA インスタンスが停止したら、SAP HANA システム PKI **SSFS\_RH1.KEY** および **SSFS\_RH1.DAT** ファイルをプライマリー SAP HANA インスタンスからセカンダリー SAP HANA インスタンスにコピーします。

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

詳細は、[SAP Note 2369981 - Required configuration steps for authentication with HANA System Replication](#) 参照してください。

これで、以下のコマンドを使用して、SAP HANA セカンダリーレプリケーションインスタンスを SAP HANA プライマリーレプリケーションインスタンスに登録できるようになります。

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

HANA システムレプリケーションの要件に応じて、`replicationMode` と `operationMode` の値を選択してください。詳細は、[Replication Modes for SAP HANA System Replication](#) および [Operation Modes for SAP HANA System Replication](#) を参照してください。

登録が成功すると、SAP HANA セカンダリーレプリケーションインスタンスを再度起動できます。

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

セカンダリーノードが実行中であり、`'mode'` が `hdbnsutil -sr_register` コマンドの `replicationMode` パラメーターに使用される値と一致していることを確認します。登録が成功した場合、SAP HANA セカンダリーレプリケーションインスタンスの SAP HANA システムレプリケーションステータスは、以下のようになります。

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

## 2.5. SAP HANA システムレプリケーションの状態の確認

SAP HANA システムレプリケーションの現在の状態を確認するには、現在のプライマリー SAP HANA ノードで SAP HANA 管理ユーザーとして SAP HANA によって提供される **systemReplicationStatus.py** Python スクリプトを使用できます。

シングルテナントの SAP HANA セットアップでは、出力は次のようになります。

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

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

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

マルチテナント SAP HANA セットアップでは、出力は次のようになります。

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

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

```
Local System Replication State
```

```
~~~~~
```

```
mode: PRIMARY
```

```
site id: 1
```

```
site name: DC1
```

いずれの場合も、戻りコードも確認してください。

```
echo $?
15
```

戻りコード 15 (アクティブ) は問題ありません。14 は同期、13 は初期化を意味します。

## 2.6. SAP HANA システムレプリケーションのテスト

テストフェーズは、KPI が満たされており、ランドスケープが設定どおりに機能するかどうかを検証する非常に重要なフェーズです。HA クラスターがないと SAP HANA システムレプリケーションのセットアップが期待どおりに動作しない場合、SAP HANA システムレプリケーションのセットアップを管理するために後で HA クラスターを設定するときに、予期しない動作が発生する可能性があります。

したがって、ガイドラインとして以下にいくつかのテストケースを提案しますが、特定の要件によって強化する必要があります。テストは、現実的なデータ負荷とサイズで実行する必要があります。

テストケース	説明
完全なレプリケーション	初回同期にかかる時間を開始時から測定します。
接続の切断	プライマリーとセカンダリーが接続されるまでにかかる時間を測定します。
テイクオーバー	セカンダリーシステムが完全に動作するまでにかかる時間を測定します。
データの整合性	データを作成または変更してから、テイクオーバーを実行して、データがまだ利用可能かどうかを確認します。
クライアントの再接続	テイクオーバー後にクライアントアクセスをテストし、DNS/仮想 IP スイッチが機能しているかどうかを確認します。
プライマリーがセカンダリーになる	テイクオーバー後に以前のプライマリーがセカンダリーになったときに、両方のシステムが同期するまでにかかる時間を測定します。

詳細は、[How To Perform System Replication for SAP HANA](#) の "9. Testing" を参照してください。

## 第3章 SAP HANA スケールアップシステムレプリケーションセッ トアップを管理するための HA クラスターの設定

RHEL でのペースメーカーベースの HA クラスターのセットアップに関する一般的なガイダンスについては、次のドキュメントを参照してください。

- [RHEL 9 での高可用性クラスターの設定と管理](#)
- [Support Policies for RHEL High Availability Clusters](#)

このガイドの残りの部分では、以下が設定され、適切に動作していることを前提とします。

- 基本的な HA クラスターは Red Hat の公式ドキュメントに従って設定されており、適切に機能するフェンシングを備えています (セットアップが実行されているプラットフォームに応じてどのフェンシングメカニズムを使用するかに関するガイドラインは、[フェンシング/STONITH](#) のサポートポリシーを参照してください。)



### 注記

HA クラスターによって管理される SAP HANA インスタンスによってアクセスされる共有ストレージがないため、このソリューションではフェンシング/STONITH メカニズムとしての fence\_scsi/fence\_mpath の使用はサポートされていません。

- SAP HANA システムレプリケーションが設定されており、SAP HANA インスタンス間の手動テイクオーバーが正しく機能していることが確認されています。
- SAP HANA インスタンスのブート時の自動起動は、すべての HA クラスターノードで無効になっています (SAP HANA インスタンスの開始と停止は HA クラスターによって管理されません)。



### 注記

HA クラスターによって管理される SAP HANA インスタンスが [systemd 対応](#) (SAP HANA 2.0 SPS07 以降) である場合は、systemd が HA クラスターによる SAP インスタンスの管理を妨げないようにするために、追加の設定変更が必要です。詳細は、[2.Red Hat HA Solutions for SAP \(The Systemd-Based SAP Startup Framework\)](#) を参照してください。

### 3.1. RHEL HA アドオンを使用した SAP HANA スケールアップシステムレプリケーションの管理に必要なリソースエージェントとその他のコンポーネントのインストール

SAP HANA スケールアップシステムレプリケーションのセットアップを管理するための HA クラスターのセットアップに必要なリソースエージェントおよびその他の SAP HANA 固有のコンポーネントは、“RHEL for SAP Solutions” リポジトリの resource-agents-sap-hana RPM パッケージで提供されません。

パッケージをインストールするには、次のコマンドを使用してください。

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

### 3.2. SAP HANA SRCONNECTIONCHANGED() フックの有効化



SAP の [Implementing a HA/DR Provider](#) に記載されているように、SAP HANA の最新バージョンでは、SAP HANA が特定のイベントの通知を送信できるようにするいわゆる "hooks" が提供されています。**srConnectionChanged()** フックを使用すると、HA クラスターのアクションを必要とする SAP HANA システムレプリケーションのステータスの変更が発生したことを検出する HA クラスターの機能を向上させ、偶発的なテイクオーバーが回避すべき状況でトリガーされないようにすることで、データ損失/データ破損を避けることができます。



### 注記

SAP HANA 2.0 SPS0 以降、および **srConnectionChanged()** フックをサポートするコンポーネントを提供する **resource-agents-sap-hana** パッケージのバージョンを使用する場合は、HA クラスターのセットアップを続行する前にフックを有効にすることが必須となります。

### 3.2.1. resource-agents-sap-hana パッケージのバージョンの確認

[How can the srConnectionChanged\(\) hook be used to improve the detection of situations where a takeover is required, in a Red Hat Pacemaker cluster managing HANA Scale-up or Scale-out System Replication?](#) に従って、RHEL 9 のバージョンで **srConnectionChanged()** フックを有効にするために必要なコンポーネントを提供する **resource-agents-sap-hana** パッケージの正しいバージョンがインストールされていることを確認してください。

### 3.2.2. すべての SAP HANA インスタンスで srConnectionChanged() フックのアクティブ化



### 注記

**srConnectionChanged()** フックをアクティブ化する手順は、すべての HA クラスターノード上の SAP HANA インスタンスごとに実行する必要があります。

1. 両方のノードの HA クラスターを停止します (このコマンドは1つの HA クラスターノードでのみ実行する必要があります)。

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

すべての SAP HANA インスタンスが完全に停止していることを確認します。

2. 各ノードの SAP HANA **global.ini** ファイルを更新して、(ファイル **/hana/shared/RH1/global/hdb/custom/config/global.ini** などの) 両方の SAP HANA インスタンスでフックスクリプトを使用できるようにします。

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

[trace]
ha_dr_saphanasr = info
```

3. 各 HA クラスターノードで、次のコマンドを実行し、以下のコンテンツを追加することで、ファイル **/etc/sudoers.d/20-saphana** を作成し、**srConnectionChanged()** フックが呼び出された際にフックスクリプトがノード属性を更新できるようにします。

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

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

**rh1** をSAP HANA インストールの小文字のSIDに置き換え、**DC1** と **DC2** をSAP HANA サイト名に置き換えます。

**Defaults** 設定が必要な理由の詳細は、[The srHook attribute is set to SFAIL in a Pacemaker cluster managing SAP HANA system replication, even though replication is in a healthy state](#) を参照してください。

- HA クラスタを起動せずに、両方の HA クラスタノードで SAP HANA インスタンスを手動で起動します。

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

- フックスクリプトが期待どおりに動作していることを確認します。SAP HANA インスタンスを停止するなど、フックをトリガーするための何らかのアクションを実行します。次に、以下のような方法を使用して、フックが何かを記録したかどうかを確認します。

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



### 注記

SAP HANA フックが正しく動作していることを確認する方法の詳細は、SAP ドキュメント [Install and Configure a HA/DR Provider Script](#) を参照してください。

- フックの機能が確認されたら、HA クラスタを再度起動できます。

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

## 3.3. 一般的な HA クラスタプロパティの設定

リソースの不必要なフェイルオーバーを回避するには、**resource-stickiness** パラメーターと **migration-threshold** パラメーターの次のデフォルト値を設定する必要があります (これは1つのノードでのみ行う必要があります)。

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



## 注記

RHEL 9 の時点で、上記のコマンドは非推奨となりました。代わりに次のコマンドを使用してください。

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

**resource-stickiness=1000** は、リソースが現在の場所で実行し続けることを奨励しますが、**migration-threshold=5000** は、5000 回の失敗後にのみ、リソースを新しいノードに移動させます。リソースが別のノードに早期にフェイルオーバーすることを防ぐには、通常は **5000** で十分です。これにより、リソースのフェイルオーバー時間が制御可能な制限内に収まるようになります。

### 3.4. クローンされた SAPHANATOPOLOGY リソースの作成

**SAPHanaTopology** リソースエージェントは、各ノード上の SAP HANA システムレプリケーションのステータスと設定に関する情報を収集します。さらに、SAP HANA インスタンスの起動、停止、監視に必要なローカル **SAP HostAgent** を起動して監視します。

**SAPHanaTopology** リソースエージェントには次の属性があります。

属性名	必須/オプション	デフォルト値	説明
SID	はい	null	SAP HANA インストールの SAP システム識別子 (SID) (すべてのノードで同一である必要があります)。例: RH1
InstanceNumber	はい	null	SAP HANA インストールのインスタンス番号 (すべてのノードで同一である必要があります)。例: 02

以下は、**SAPHanaTopology** のクローン作成されたリソースを作成するコマンドの例です。

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

結果として得られるリソースは次のようになります。

```
[root]# pcs resource config SAPHanaTopology_RH1_02-clone
Clone: SAPHanaTopology_RH1_02-clone
Meta Attrs: clone-max=2 clone-node-max=1 interleave=true Resource: SAPHanaTopology_RH1_02
(class=ocf provider=heartbeat type=SAPHanaTopology)
```

Attributes: SID=RH1 InstanceNumber=02

Operations: start interval=0s timeout=600 (SAPHanaTopology\_RH1\_02-start-interval-0s)

stop interval=0s timeout=300 (SAPHanaTopology\_RH1\_02-stop-interval-0s)

monitor interval=10 timeout=600 (SAPHanaTopology\_RH1\_02-monitor-interval-10s)



## 注記

リソース操作に対して示されているタイムアウトは単なる例であり、実際の SAP HANA セットアップに応じて調整する必要があるかもしれません (たとえば、大規模な SAP HANA データベースは起動に時間がかかる場合があるため、開始タイムアウトを増やす必要があります)。

リソースが開始されると、ノード属性の形式で保存された収集された情報が表示されます。この情報は **pcs status --full** コマンドで表示できます。以下は、**SAPHanaTopology** のみが開始された場合、属性がどのようになるかを示した例です。

```
[root]# pcs status --full
...
Node Attributes:
* Node node1:
+ hana_rh1_remoteHost      : node2
+ hana_rh1_roles           : 1:P:master1::worker:
+ hana_rh1_site            : DC1
+ hana_rh1_srmode          : syncmem
+ hana_rh1_vhost           : node1
* Node node2:
+ hana_rh1_remoteHost      : node1
+ hana_rh1_roles           : 1:S:master1::worker:
+ hana_rh1_site            : DC2
+ hana_rh1_srmode          : syncmem
+ hana_rh1_vhost           : node2
...
```

## 3.5. プロモーション可能な SAPHANA リソースの作成

**SAPHana** リソースエージェントは、SAP HANA スケールアップシステムレプリケーションの一部である SAP HANA インスタンスを管理し、SAP HANA システムレプリケーションのステータスも監視します。SAP HANA プライマリーレプリケーションインスタンスに障害が発生した場合、**SAPHana** リソースエージェントは、リソースエージェントパラメーターの設定方法に基づいて、SAP HANA システムレプリケーションのテイクオーバーをトリガーできます。

**SAPHana** リソースエージェントには、以下の属性があります。

属性名	必須/オプション	デフォルト値	説明
SID	はい	null	SAP HANA インストールの SAP システム識別子 (SID) (すべてのノードで同一である必要があります)。例: RH1

属性名	必須/オプション	デフォルト値	説明
InstanceNumber	はい	null	SAP HANA インストールのインスタンス番号 (すべてのノードで同一である必要があります)。例: 02
PREFER_SITE_TAKEOVER	いいえ	null	リソースエージェントは、プライマリインスタンスをローカルで再起動するのではなく、セカンダリーインスタンスに切り替えることを優先する必要がありますか? true: セカンダリーサイトへのテイクオーバーを優先します。false: ローカルで再起動することを優先します。never: いかなる状況でも、他のノードのテイクオーバーは行われません。
AUTOMATED_REGISTER	いいえ	false	テイクオーバーイベントが発生した場合、以前のプライマリインスタンスをセカンダリーとして登録する必要がありますか? ("false": いいえ、手動介入が必要です。 "true": はい、以前のプライマリはリソースエージェントによってセカンダリーとして登録されます)。

属性名	必須/オプション	デフォルト値	説明
DUPLICATE_PRIMARY_TIMEOUT	いいえ	7200	クラスターが反応する前にデュアルプライマリ状況が発生した場合は、2つのプライマリタイムスタンプ間に時間差が必要です。時間差が時間ギャップより小さい場合、クラスターは一方または両方のインスタンスを "WAITING" ステータスで保持します。これは、管理者にフェイルオーバーに対応する機会を与えるためです。以前のプライマリーのノード全体がクラッシュした場合、時間差が経過した後に以前のプライマリが登録されます。SAP HANA インスタンス "のみ" がクラッシュした場合は、以前のプライマリがすぐに登録されます。新しいプライマリへのこの登録後、すべてのデータがシステムレプリケーションによって上書きされます。

**PREFER\_SITE\_TAKEOVER**、**AUTOMATED\_REGISTER**、および **DUPLICATE\_PRIMARY\_TIMEOUT** パラメーターは、HA クラスターによって管理される SAP HANA システムレプリケーションの可用性とデータ保護の要件に従って設定する必要があります。

通常、新しい SAP HANA プライマリインスタンスが完全にアクティブになるまでの時間は、元の SAP HANA プライマリインスタンスが再起動して、すべてのデータをディスクからメモリーに再ロードする際にかかる時間よりも短いため、一般に、**PREFER\_SITE\_TAKEOVER** は true に設定し、プライマリ SAP HANA インスタンスの障害が検出された場合に HA クラスターがテイクオーバーをトリガーできるようにします。

HA クラスターによってトリガーされたテイクオーバーが発生した後に、新しいプライマリ SAP HANA インスタンス上のすべてのデータが正しいことを検証できるようにするには、**AUTOMATED\_REGISTER** を false に設定する必要があります。これにより、オペレーターは、テイクオーバーが偶然発生した場合に古いプライマリ SAP HANA インスタンスに戻るか、テイクオーバーが正しかった場合は、古いプライマリ SAP HANA インスタンスを新しいセカンダリー SAP HANA インスタンスとして登録し、SAP HANA システムレプリケーションが再び動作するようにするかのいずれかの可能性を提供します。

**AUTOMATED\_REGISTER** が true に設定されている場合、HA クラスターによるテイクオーバーが発生した後、古いプライマリ SAP HANA インスタンスが SAPHana リソースエージェントによって新しいセカンダリー SAP HANA インスタンスとして自動的に登録されます。これにより、SAP HANA システムレプリケーションセットアップの可用性が向上し、SAP HANA システムレプリケーション環境におけるいわゆる "dual-primary" 状況が阻止されます。ただし、セカンダリー SAP HANA インスタンス上

のデータが完全に同期していないにもかかわらず、HA クラスターによってテイクオーバーがトリガーされた場合、古いプライマリー SAP HANA インスタンスを新しいセカンダリー SAP HANA インスタンスとして自動登録すると、このインスタンス上のすべてのデータが削除されるため、テイクオーバーが発生する前に同期されていなかったデータは利用できなくなることから、データ損失やデータ破損のリスクが高まる可能性があります。

SAP HANA インスタンスと SAP HANA システムレプリケーションを管理するための昇格可能な **SAPHana** クラスターリソースは、次の例のように作成できます。

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

結果の HA クラスターリソースは以下ようになります。

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



## 注記

リソース操作のタイムアウトは単なる例であり、実際の SAP HANA セットアップに応じて調整する必要があるかもしれません (たとえば、大規模な SAP HANA データベースは起動に時間がかかる場合があるため、起動タイムアウトを増やす必要があります)。

リソースが開始され、HA クラスターが最初の監視操作を実行すると、以下に示すように、ノード上の SAP HANA データベースの現状を記述する追加のノード属性が追加されます。

```
[root]# pcs status --full
...
Node Attributes:
* Node node1:
+ hana_rh1_clone_state      : PROMOTED
+ hana_rh1_op_mode         : delta_datashipping
+ hana_rh1_remoteHost      : node2
+ hana_rh1_roles           : 4:P:master1:master:worker:master
+ hana_rh1_site            : DC1
```

```

+ hana_rh1_sync_state      : PRIM
+ hana_rh1_srmode         : syncmem
+ hana_rh1_version        : 2.00.064.00.1660047502
+ hana_rh1_vhost         : node1
+ lpa_rh1_lpt            : 1495204085
+ master-SAPHana_RH1_02  : 150
* Node node2:
+ hana_r12_clone_state    : DEMOTED
+ hana_rh1_op_mode       : delta_datashipping
+ hana_rh1_remoteHost    : node1
+ hana_rh1_roles         : 4:S:master1:master:worker:master
+ hana_rh1_site          : DC2
+ hana_rh1_srmode        : syncmem
+ hana_rh1_sync_state    : SOK
+ hana_rh1_version       : 2.00.064.00.1660047502
+ hana_rh1_vhost        : node2
+ lpa_rh1_lpt            : 30
+ master-SAPHana_RH1_02  : -INFINITY
...

```

### 3.6. 仮想 IP アドレスリソースの作成

クライアントが、現在実行されている HA クラスターノードから独立してプライマリー SAP HANA インスタンスにアクセスできるようにするには、仮想 IP アドレスが必要です。HA クラスターは、プライマリー SAP HANA インスタンスが実行されているノードでこのアドレスを有効にします。

HA クラスターが VIP を管理できるようにするには、IP **192.168.0.15** を使用して **IPAddr2** リソースを作成します。

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

HA クラスターが実行されているプラットフォームに基づいて、仮想 IP アドレスを管理するための適切なリソースエージェントを使用してください。

結果の HA クラスターリソースは次のようになります。

```

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

```

### 3.7. 制約の作成

正しく動作させるには、**SAPHana** リソースを開始する前に **SAPHanaTopology** リソースが開始されていることと、プライマリー SAP HANA インスタンスが実行されているノードに仮想 IP アドレスが存在することを確認する必要があります。

これを達成するには、次の制約が必要です。

#### 3.7.1. 制約 - **SAPHana** の前に **SAPHanaTopology** を開始します。



以下のコマンド例は、これらのリソースの **start** 順序を義務付ける制約を作成します。ここで言及すべきことが2つあります。

- **symmetrical=false** 属性は、リソースの開始のみを考慮し、リソースを逆の順序で停止する必要がないことを定義します。
- 両方のリソース (**SAPHana** と **SAPHanaTopology**) には、ノード上でこれらのリソースを並列開始できる属性 **interleave=true** があります。これにより、順序制約を設定しているにもかかわらず、すべてのノードが **SAPHanaTopology** を開始するのを待たずに、**SAPHanaTopology** が実行されたらすぐに任意のノードで **SAPHana** リソースを開始できるようになります。

制約を作成するためのコマンドは、以下のとおりです。

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

結果の制約は、次の例のようになります。

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

### 3.7.2. 制約 - IPAddr2 リソースを SAPHana リソースのマスターと同じ場所に配置します

以下は、マスターとして昇格された **SAPHana** リソースと **IPAddr2** リソースを同じ場所に配置するコマンドの例です。

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

制約では、デフォルトの INFINITY の代わりにスコア 2000 が使用されていることに注意してください。これにより、**SAPHana** リソースでマスターが昇格されない場合でも **IPAddr2** リソースがアクティブな状態を維持できるため、このアドレスを使用して SAP インスタンスに関するステータス情報クエリーできる SAP Management Console (MMC) や SAP Landscape Management (LaMa) などのツールを引き続き使用できます。

結果の制約は次のようになります。

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

## 3.8. ACTIVE/ACTIVE (READ-ENABLED) SAP HANA システムレプリケーションセットアップ用のセカンダリー仮想 IP アドレスの追加 (オプション)

SAP HANA 2.0 SPS1 以降、SAP HANA は SAP HANA システムレプリケーションの [Active/Active \(Read-Enabled\)](#) セットアップをサポートしており、SAP HANA システムレプリケーションセットアップのセカンダリーインスタンスを読み取り専用アクセスに使用できます。

このようなセットアップをサポートするには、クライアントがセカンダリー SAP HANA インスタンスにアクセスできるようにする 2 番目の仮想 IP アドレスが必要です。テイクオーバーが発生した後もセカンダリーレプリケーションサイトに確実にアクセスできるようにするために、HA クラスターは、昇格可能な SAPHana リソースのスレーブで仮想 IP アドレスを移動する必要があります。

SAP HANA で [Active/Active \(Read-Enabled\)](#) モードを有効にするには、セカンダリー SAP HANA インスタンスを登録するときに、`operationMode` を `logreplay_readaccess` に設定する必要があります。

### 3.8.1. セカンダリー仮想 IP アドレスを管理するリソースの作成

```
[root]# pcs resource create vip2_RH1_02 IPAddr2 ip="192.168.1.11"
```

HA クラスターが実行されているプラットフォームに基づいて、仮想 IP アドレスを管理するための適切なリソースエージェントを使用してください。

### 3.8.2. 場所の制約の作成

これは、セカンダリー仮想 IP アドレスが適切な HA クラスターノードに確実に配置されるようにするためです。

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

これらの場所の制約により、2 番目の仮想 IP リソースは確実に次の動作をします。

- プライマリー SAP HANA インスタンスとセカンダリー SAP HANA インスタンスが両方とも実行されていて、SAP HANA システムレプリケーションが同期している場合、セカンダリー SAP HANA インスタンスが実行されている HA クラスターノード上で 2 番目の仮想 IP がアクティブになります。
- セカンダリー SAP HANA インスタンスが実行されていない場合、または SAP HANA システムレプリケーションが同期していない場合、プライマリー SAP HANA インスタンスが実行されている HA クラスターノード上で 2 番目の仮想 IP がアクティブになります。セカンダリー SAP HANA インスタンスが実行中で、SAP HANA システムレプリケーションが再び同期されると、2 番目の仮想 IP はセカンダリー SAP HANA インスタンスが実行されている HA クラスターノードに戻ります。
- プライマリー SAP HANA インスタンスが実行されておらず、HA クラスターによって SAP HANA テイクオーバーがトリガーされた場合、他のノード上の SAP HANA インスタンスが新しいセカンダリーとして登録され、SAP HANA システムレプリケーションが再び同期されるまで、2 番目の仮想 IP は同じノード上で実行され続けます。

これにより、正常な SAP HANA インスタンスが実行されているノードに 2 番目の仮想 IP リソースが割り当てられる時間が最大化されます。

## 3.9. HDBINDEXSERVER プロセス失敗アクション (オプション) に対する SAP HANA SRSERVICESTATECHANGED() フックの有効化

HANA は、`indexserver` プロセスの問題を検出すると、SAP HANA に組み込まれている機能を使用して、`indexserver` プロセスを自動的に停止および再起動することで問題を回復します。

ただし、場合によっては、サービスの停止フェーズに非常に長い時間がかかることがあります。その間、システムレプリケーションは同期されなくなる可能性があります。HANA は引き続き動作し、新しい接続を受け入れます。最終的に、サービスは停止と再起動のプロセスを完了し、回復します。

データの一貫性にリスクをもたらすこの長時間実行される再起動を待つ代わりに、その間にインスタンスで何か他の障害が発生した場合、**ChkSrv.py** フックスクリプトが状況に反応し、HANA インスタンスを停止して、より迅速な回復を行うことができます。自動フェイルオーバーが有効になっているセットアップでは、セカンダリーノードが正常な状態であれば、インスタンスが停止するとテイクオーバーが開始します。それ以外の場合、リカバリーはローカルで続行しますが、強制インスタンスの再起動によってリカバリーが高速化されます。

**global.ini** 設定ファイルで設定していると、SAP HANA はインスタンス内のすべてのイベントに対して **ChkSrv.py** フックスクリプトを呼び出します。スクリプトはイベントを処理し、イベントの詳細に適用したフィルターの結果に基づいてアクションを実行します。これにより、障害後に HANA によって停止および再起動する HANA **indexserver** プロセスと、インスタンスのシャットダウンの一環として停止する同じプロセスを区別できます。

実行可能なさまざまなアクションを以下に示します。

- Ignore (無視): このアクションは、解析されたイベントと決定情報を専用のログファイルに書き込むだけなので、フックスクリプトが何を実行するかを確認するのに役立ちます。
- Stop (停止): このアクションは、**sapcontrol** コマンドを通じてインスタンスに対して正常な **StopSystem** を実行します。
- Kill (強制終了): このアクションは、設定可能なデフォルトのシグナル 9 を使用して **HDB kill-<signal>** コマンドを実行します。

停止アクションと強制終了アクションの両方で HANA インスタンスが停止しますが、最終的には強制終了の方が少し速くなることに注意してください。

この時点で、クラスターは HANA リソースの障害を認識し、設定された方法で対応します。通常はインスタンスを再起動し、有効になっている場合はテイクオーバーも行います。

### 3.9.1. resource-agents-sap-hana パッケージのバージョンの確認

[Pacemaker cluster does not trigger a takeover of HANA System Replication when the hdbindexserver process of the primary HANA instance hangs/crashes](#) に記載されているように、RHEL 9 のバージョンで **srServiceStateChanged()** フックを有効にするために必要なコンポーネントを提供する正しいバージョンの **resource-agents-sap-hana** パッケージがインストールされていることを確認してください。

### 3.9.2. すべての SAP HANA インスタンスで **srServiceStateChanged()** フックをアクティブ化する



#### 注記

**srServiceStateChanged()** フックをアクティブ化する手順は、すべての HA クラスターノード上の SAP HANA インスタンスごとに実行する必要があります。

1. 各ノードの SAP HANA **global.ini** ファイルを更新して、(ファイル **/hana/shared/RH1/global/hdb/custom/config/global.ini** などの) 両方の SAP HANA インスタンスでフックスクリプトを使用できるようにします。

```
[ha_dr_provider_chksrv]
provider = ChkSrv
```

```
path = /usr/share/SAPHanaSR/srHook
execution_order = 2
action_on_lost = stop

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

オプションのパラメーターを以下のように設定します。

- **action\_on\_lost** (デフォルト: ignore)
- **stop\_timeout** (デフォルト: 20)
- **kill\_signal** (デフォルト: 9)

以下は、**action\_on\_lost** で使用可能なオプションの説明です。

- **ignore**: この機能を有効にしますが、イベントのみをログに記録します。これは、設定された環境でのフックのアクティビティを監視するのに役立ちます。
  - **stop**: **sapcontrol -nr <nr> -function StopSystem** を正常に実行します。
  - **kill**: 最速の停止のために HDB **kill-<signal>** を実行します。
  - **stop\_timeout** は stop および kill アクションのコマンド実行に追加され、**kill\_signal** は **HDB kill-<signal>** コマンドの一部として kill アクションで使用されることに注意してください。
2. HANA の実行中に **HA/DR** プロバイダーを再ロードして、新しいフックをアクティブ化します。

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

3. 新しいトレースファイルをチェックしてフックの初期化を確認します。

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

## 第4章 セットアップのテスト

HA クラスターのセットアップを実稼働環境に移す前に、すべてが想定どおりに動作することを確認するために徹底的にテストする必要があります。また、オペレーターが、特定の状況での HA クラスターの動作方法や、障害が発生した場合にセットアップを健全な状態に戻す方法を経験できるようにする必要があります。

少なくとも次のテストを実行する必要があります。

- HA クラスターコマンドを使用して、プライマリー SAP HANA インスタンスの手動移動を実行します。  
予想される結果: SAP HANA 側でテイクオーバーがトリガーされ、セカンダリー SAP HANA インスタンスが新しいプライマリー SAP HANA インスタンスに昇格します。SAPHana リソースの **AUTOMATED\_REGISTER** パラメーターの設定に応じて、HA クラスターは以前のプライマリーインスタンスを新しいセカンダリーとして自動的に登録するか、オペレーターが以前のプライマリーインスタンスに何が起こるかを決定する必要があります。
- プライマリー SAP HANA インスタンスが実行されている HA クラスターノードをクラッシュします。  
予想される結果: HA クラスターノードがフェンスされ、SAP HANA 側でテイクオーバーがトリガーされ、セカンダリー SAP HANA インスタンスが新しいプライマリー SAP HANA インスタンスになるように昇格します。SAPHana リソースの **AUTOMATED\_REGISTER** パラメーターの設定に応じて、HA クラスターは以前のプライマリーインスタンスを新しいセカンダリーとして自動的に登録するか、オペレーターが以前のプライマリーインスタンスに何が起こるかを決定する必要があります。
- HA クラスター外のプライマリー SAP HANA インスタンスを手動で停止します。  
予想される結果: SAP HANA 側でテイクオーバーがトリガーされ、セカンダリー SAP HANA インスタンスが新しいプライマリー SAP HANA インスタンスに昇格します。SAPHana リソースの **AUTOMATED\_REGISTER** パラメーターの設定に応じて、HA クラスターは以前のプライマリーインスタンスを新しいセカンダリーとして自動的に登録するか、オペレーターが以前のプライマリーインスタンスに何が起こるかを決定する必要があります。
- セカンダリー SAP HANA インスタンスが実行されているノードをクラッシュします。  
期待される結果: HA クラスターノードがオンラインに戻り、SAP HANA システムレプリケーションが再開されると、HA クラスターノードはフェンスされ、セカンダリー SAP HANA インスタンスが起動されるはずですが。
- HA クラスター外のセカンダリー SAP HANA インスタンスを手動で停止します。  
予想される結果: セカンダリー SAP HANA インスタンスが HA クラスターによって再起動される必要があります。
- SAP HANA システムレプリケーションで使用されるネットワーク接続を無効にします。  
予想される結果: HA クラスターは、SAP HANA システムレプリケーションの障害が発生したことを検出しますが、両方のノードで SAP HANA インスタンスを実行し続ける必要があります。

## 第5章 メンテナンス手順

### 5.1. OS と HA クラスターコンポーネントの更新

詳細は、[Recommended Practices for Applying Software Updates to a RHEL High Availability or Resilient Storage Cluster](#) を参照してください。

### 5.2. SAP HANA インスタンスの更新

このドキュメントで説明されている HA クラスター設定を使用して SAP HANA システムレプリケーションのセットアップが管理されている場合、更新の前後に SAP HANA インスタンスを更新する実際のプロセスに加えて、追加の手順がいくつか必要になります。次の手順を実行します。

1. SAPHana リソースを管理対象外モードにします。

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

2. SAP が提供する手順を使用して、SAP HANA インスタンスを更新します。
3. SAP HANA インスタンスの更新が完了し、SAP HANA システムレプリケーションが再び動作していることが確認されたら、SAPHana リソースのステータスを更新して、クラスターが SAP HANA システムレプリケーションのセットアップの現状を認識していることを確認する必要があります。

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

4. HA クラスターが SAP HANA システムレプリケーションセットアップの現在のステータスを正しく取得した場合は、SAPHana リソースを管理対象モードに戻し、HA クラスターが SAP HANA システムレプリケーションセットアップの問題に再び対応できるようにします。

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

### 5.3. SAPHANA リソースの別のノードへの手動移動 (HA クラスターによる SAP HANA システムレプリケーションのテイクオーバー)

SAP HANA システムレプリケーションの手動テイクオーバーは、昇格可能なクローンリソースを移動することでトリガーできます。

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



## 注記

テイクオーバーが完了し、制約が削除された後に以前の SAP HANA プライマリーインスタンスに何が起こるかは、SAPHana リソースの **AUTOMATED\_REGISTER** パラメーターの設定によって異なります。

- **Automated\_REGISTER=true** の場合、以前の SAP HANA プライマリーインスタンスが新しいセカンダリーとして登録され、SAP HANA システムレプリケーションが再びアクティブになります。
- **AUTOMATED\_REGISTER=false** の場合、テイクオーバー後に以前の SAP HANA プライマリーインスタンスに何が起こるかは、オペレーターが決定します。

## 第6章 参考資料

### 6.1. RED HAT

- [RHEL 9 for SAP HANA2 インストールの設定](#)
- [RHEL 9 での高可用性クラスターの設定と管理](#)
- [Support Policies for RHEL High Availability Clusters](#)
- [Support Policies for RHEL High Availability Clusters - Fencing/STONITH](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP HANA in a Cluster](#)
- [SAP HANA、S/4HANA および NetWeaver ベースの SAP アプリケーション向け Red Hat HA ソリューション](#)

### 6.2. SAP

- [SAP HANA Server Installation and Update Guide](#)
- [SAP HANA System Replication](#)
- [Implementing a HA/DR Provider](#)
- [SAP Note 2057595 - FAQ: SAP HANA High Availability](#)
- [SAP Note 2063657 - SAP HANA System Replication Takeover Decision Guideline](#)
- [SAP Note 3007062 - FAQ: SAP HANA & Third Party Cluster Solutions](#)

### 6.3. その他

- [Be Prepared for Using Pacemaker Cluster for SAP HANA – Part 1: Basics](#)
- [Be Prepared for Using Pacemaker Cluster for SAP HANA – Part 2: Failure of Both Nodes](#)