



Red Hat Enterprise Linux for SAP Solutions 9

RHEL HA アドオンを使用して SAP NetWeaver
または SAP S/4HANA アプリケーションサー
バーインスタンスを管理するための HA クラス
ターの設定

Red Hat Enterprise Linux for SAP Solutions 9 RHEL HA アドオンを使用して SAP NetWeaver または SAP S/4HANA アプリケーションサーバーインスタンスを管理するための HA クラスターの設定

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

このガイドでは、RHEL HA アドオンを使用して SAP NetWeaver または SAP S/4HANA アプリケーションサーバーインスタンスを管理するように HA クラスタを設定するプロセスの概要を説明します。

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
第1章 概要	5
1.1. 概要	5
1.2. 対象者	5
1.3. 概念	5
1.4. リソースエージェント	8
1.5. マルチ SID サポート (オプション)	13
1.6. サポートポリシー	14
第2章 要件	15
2.1. サブスクリプションおよびリポジトリ	15
2.2. ストレージ要件	15
2.3. フェンシング/STONITH	16
2.4. クォーラム	17
第3章 SAP アプリケーションサーバーインスタンスのインストール	18
3.1. このドキュメントで使用する設定オプション	18
3.2. SAP インスタンスをインストールするためのクラスターノードの準備	18
3.3. SAP インスタンスのインストール	19
3.4. インストール後の設定	22
第4章 クラスターのセットアップ	24
4.1. 各ノードでの基本的なクラスターインストールの実行	24
4.2. 一般的なクラスタープロパティの設定	24
4.3. すべてのクラスターノードへの RESOURCE-AGENTS-SAP パッケージのインストール	24
4.4. 共有ファイルシステムへのアクセスの設定	25
4.5. (A)SCS リソースグループの設定	25
4.6. ERS リソースグループの設定	27
4.7. 制約の作成	28
4.8. データベースインスタンスを管理するためのクラスターリソースグループの設定 (オプション)	29
4.9. PRIMARY/ADDITIONAL APPLICATION SERVER (PAS/AAS) リソースグループの設定 (オプション)	31
4.10. STANDALONE ENQUEUE SERVER 2 (ENSA2) MULTI-NODE CLUSTER (オプション)	33
4.11. SAP HA インターフェイスを有効にして、クラスターが制御する SAP インスタンスを SAP 管理ツールで管理できるようにする方法 (オプション)	35
4.12. システム起動時のクラスターの自動起動を有効にする方法 (オプション)	35
第5章 クラスター設定のテスト	36
5.1. PCS コマンドを使用した ASCS インスタンスの手動移動	36
5.2. SAPCONTROL を使用した ASCS インスタンスの手動移動 (SAP HA インターフェイスが有効な場合)	37
5.3. ASCS インスタンスの障害のテスト	38
5.4. ERS インスタンスのテストの失敗	40
5.5. ノードのクラッシュによる ASCS インスタンスのフェイルオーバー	41
5.6. ノードのクラッシュによる ERS インスタンスの障害	42
5.7. ノードのクラッシュによる ASCS インスタンスの障害 (ENSA2)	43
第6章 メンテナンス手順	45
6.1. RHEL と RHEL HA アドオンの更新	45
第7章 参考資料	46
7.1. RED HAT	46
7.2. SAP	46

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

1.1. 概要

SAP NetWeaver または SAP S/4HANA ベースのシステムは、多くのビジネスプロセスで重要な役割を果たします。そのため、ビジネスに対してこれらのシステムの継続的かつ信頼性の高い可用性を確保することが重要です。これは、HA クラスタリングを使用して SAP NetWeaver または SAP S/4HANA システムのインスタンスを管理することで実現できます。

HA クラスタリングの基本的な考え方は非常に単純です。1台の大きなマシンがすべての負荷とリスクを負担するのではなく、サービスまたは障害が発生したマシンを一時的に完全に代替するものとして、1台以上のマシンが自動的に投入されます。最良の場合、この置き換えプロセスによってシステムユーザーが中断を強いられることはありません。

1.2. 対象者

SAP NetWeaver または SAP S/4HANA に基づいた高可用性ソリューションの設計と実装は非常に複雑になる場合があります。そのため、信頼性が高く、再現性があり、正確で迅速な自動アクションを確実に実行するには、インフラストラクチャーの各レイヤーとデプロイメントのあらゆる側面に関する深い知識が必要です。

このドキュメントは、RHEL HA アドオンまたはその他のクラスタリングソリューションを使用して SAP NetWeaver または S/4HANA アプリケーションサーバーインスタンスと HA クラスタをセットアップした経験を持つ、SAP および Red Hat の認定または訓練を受けた管理者およびコンサルタントを対象としています。ソフトウェアや追加のドキュメントをダウンロードするには、[SAP Support Portal](#) と [Red Hat カスタマーポータル](#) の両方にアクセスする必要があります。

お客様のデータセンター要件を満たすようにクラスタをセットアップし、ソリューションをカスタマイズする場合、Red Hat コンサルティングをご利用いただくことを強く推奨します。通常、そのような要件は、このドキュメントで説明するソリューションよりも複雑になるためです。

1.3. 概念

1.3.1. SAP NetWeaver または S/4HANA の高可用性

一般的な SAP NetWeaver または S/4HANA 環境は、次の3つの特徴的なコンポーネントで構成されます。

- SAP **(A)SCS** インスタンス
- SAP アプリケーションサーバーインスタンス (Primary Application Server (PAS) および Additional Application Server (AAS) インスタンス)
- データベースインスタンス

(A)SCS インスタンスとデータベースインスタンスは単一障害点 (SPOF) です。したがって、データの損失や破損、および SAP システムの不必要な停止を避けるために、それらのインスタンスを HA ソリューションで確実に保護することが重要です。SPOF の詳細は、[Single point of failure](#) を参照してください。

アプリケーションサーバーの場合、エンキューサーバーが管理するエンキューロックテーブルが最も重要なコンポーネントです。これを保護するために、SAP はエンキューロックテーブルのバックアップコピーを保持する "Enqueue Replication Server" (**ERS**) を開発しました。**(A)SCS** が1つのサーバーで実行

されている間、**ERS** は常に、現在のエンキューテーブルのコピーを別のサーバー上に保持する必要があります。

このドキュメントでは、SAP と Red Hat が確立した高可用性のガイドラインに準拠する **(A)SCS** および **ERS** インスタンスを管理するための HA クラスターソリューション (2 ノードまたは 3 ノード) をセットアップする方法について説明します。HA ソリューションは、通常 SAP NetWeaver で使用される "Standalone Enqueue Server" (ENSA1) と、SAP S/4HANA で使用される "Standalone Enqueue Server 2" (ENSA2) のいずれにも使用できます。

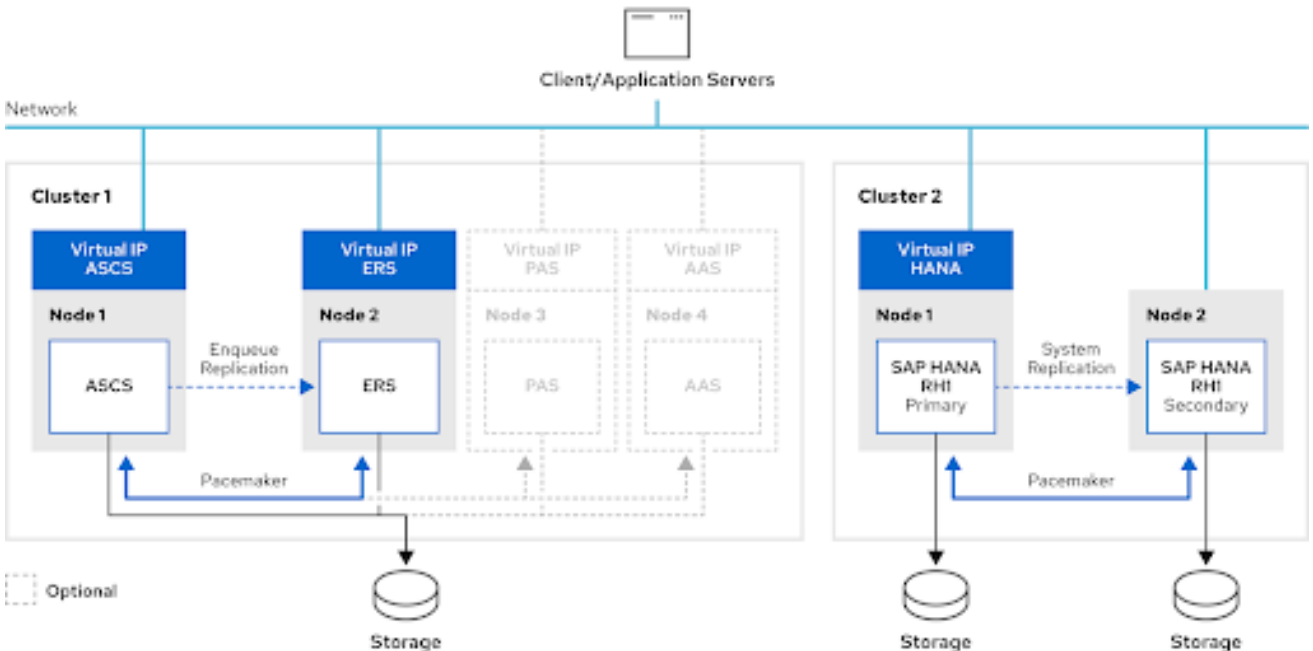
さらに、同じ HA クラスターの一部として管理できる、または別個の HA クラスター上で管理できる Primary Application Server (PAS) や Additional Application Server (AAS) インスタンスなど、他の SAP インスタンスタイプを管理できるように HA クラスターリソースをセットアップするにあたってのガイドラインも提供します。

1.3.2. ENSA1 vs. ENSA2

1.3.2.1. Standalone Enqueue Server (ENSA1)

Standalone Enqueue Server (ENSA1) の **(A)SCS** インスタンスに問題がある場合は、**(A)SCS** インスタンスが **ERS** インスタンスの "後に続く" 必要があります。つまり、HA クラスターは、**ERS** インスタンスが現在実行されているホストで **(A)SCS** インスタンスを起動する必要があります。**(A)SCS** インスタンスを実行していたホストがフェンスされるまで、両方のインスタンスが同じノード上で実行されたままになります。**(A)SCS** インスタンスを以前に実行していた HA クラスターノードがオンラインに戻ったら、エンキューレプリケーションを再開できるように、HA クラスターは **ERS** インスタンスをその HA クラスターノードに移動する必要があります。

次の図は、Standalone Enqueue Server (ENSA1) を使用して SAP NetWeaver セットアップを管理するための Pacemaker HA クラスターの一般的なアーキテクチャーを示しています。



この図では、必要に応じて Primary Application Server および Additional Application Server (PAS/AAS) インスタンスを別のサーバーで管理できることが示されています。ただし、これらのインスタンスを **(A)SCS** インスタンスや **ERS** インスタンスと同じ HA クラスターノードで実行し、当該クラスターで管理することもサポートされています。

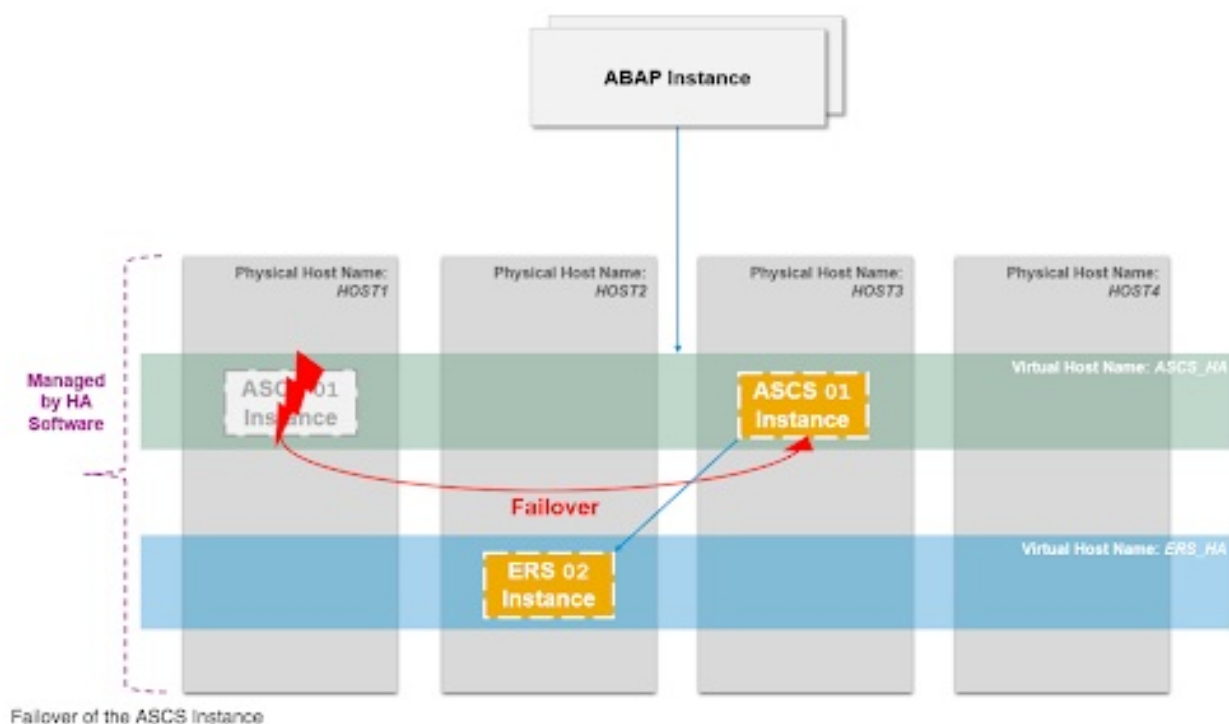
Standalone Enqueue Server (ENSA1) の動作の詳細は、SAP ドキュメントの [Standalone Enqueue Server](#) を参照してください。

1.3.2.2. Standalone Enqueue Server 2 (ENSA2)

上記で ENSA1 に関して説明したように、フェイルオーバーが発生した場合、Standalone Enqueue Server は Enqueue Replication Server の "後続く" 必要があります。つまり、HA ソフトウェアは、**ERS** インスタンスが現在実行されているホストで **(A)SCS** インスタンスを起動する必要があります。

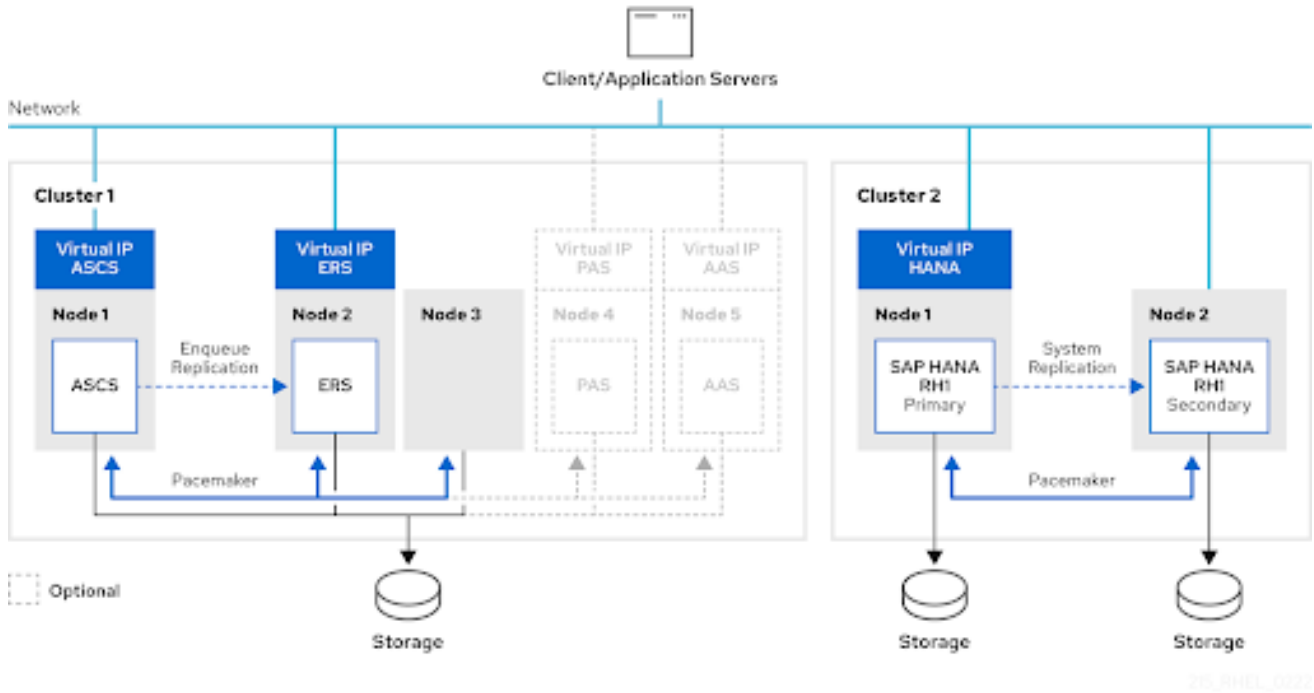
Standalone Enqueue Server (ENSA1) とは対照的に、新しい Standalone Enqueue Server 2 (ENSA2) および Enqueue Replicator 2 にはこれらの制限がなくなりました。つまり、障害が発生した場合、**ASCS** インスタンスを同じクラスターノードで再起動できます。または、別の HA クラスターノードに移動することもできます。移動先のノードは、**ERS** インスタンスが実行されている HA クラスターノードである必要はありません。これにより、Standalone Enqueue Server 2 (ENSA2) を使用する場合に、3 つ以上の HA クラスターノードを持つマルチノード HA クラスターセットアップを使用できるようになります。

3 つ以上の HA クラスターノードを使用する場合、次の図に示すように、**ASCS** はスペアノードにフェイルオーバーします。



ENSA2 の詳細は、SAP Note の [2630416 - Support for Standalone Enqueue Server 2](#) を参照してください。

次の図は、Standalone Enqueue Server 2 (ENSA2) で SAP S/4HANA セットアップを管理するために使用できる 3 ノードクラスターのアーキテクチャーを示しています。



26_RHEL_022

この図では、必要に応じて Primary Application Server および Additional Application Server (PAS/AAS) インスタンスを別のサーバーで管理できることが示されています。ただし、これらのインスタンスを **ASCS** インスタンスや **ERS** インスタンスと同じ HA クラスタードで実行し、当該クラスタで管理することもサポートされています。

SAP S/4HANA の場合、“コスト最適化”がなされた HA クラスタセットアップを使用することもできます。この場合、HANA システムレプリケーションセットアップの管理に使用するクラスタードは、**ASCS** および **ERS** インスタンスの管理にも使用されます。

1.4. リソースエージェント

RHEL 9 では、SAP 環境のさまざまなインスタンスタイプを管理するために、[resource-agents-sap RPM パッケージ](#) で以下のリソースエージェントが提供されています。

1.4.1. SAPInstance リソースエージェント

SAPInstance リソースエージェントは、SAP カーネルの一部である SAP Start Service を使用して、SAP アプリケーションサーバーインスタンスを管理するために使用できます。このリソースエージェントは、(A)SCS、ERS、PAS、および AAS インスタンスに加えて、他の SAP インスタンスタイプ (スタンドアロン SAP Web Dispatcher インスタンスや、スタンドアロン SAP Gateway インスタンスなど) の管理にも使用できます (このようなインスタンスを管理する pacemaker リソースを設定する方法は、[How to manage standalone SAP Web Dispatcher instances using the RHEL HA Add-On](#) を参照してください)。

SAPInstance リソースエージェントのすべての操作は、SAP Startup Framework が提供するコマンドで実行されます。これらのコマンドは、各 SAP インスタンスの **sapstartsrv** プロセスと通信します。**sapstartsrv** は、4 つのステータスカラーを使用します。

色	意味
緑	すべて問題ありません。

色	意味
黄	何か問題がありますが、サービスは動作しています。
赤	サービスが動作しません。
灰色	サービスが起動していません。

SAPInstance リソースエージェントは、緑と黄の場合は問題ないと解釈し、ステータスが赤またはグレーの場合は NOT_RUNNING としてクラスターに報告します。

RHEL 9 に同梱される SAPInstance リソースエージェントのバージョンは、systemd に対応した SAP Startup Framework によって管理される SAP インスタンスもサポートします (詳細は、[The Systemd-Based SAP Startup Framework](#) を参照してください)。

1.4.1.1. SAPInstance リソースエージェントの重要なパラメーター

属性名	必須	デフォルト値	説明
instanceName	はい	null	完全な SAP インスタンスプロファイル名 (<SAPSID>_<INSTANCE+INSTNO>_<virt hostname>) (例: S4H_ASCS20_s4asc s)。
START_PROFILE	いいえ	null	SAP スタートプロファイルへの完全パス (SAP NetWeaver 7.1 以降では、SAP スタートプロファイルはインスタンスプロファイルと同一です)。

属性名	必須	デフォルト値	説明
IS_ERS	いいえ	false	昇格可能なリソースを実装せず、リソース障害の後に別のクラスターノードで実行されている ERS を ASCS が検出できるようにになっていない ASCS/ERS SAP Netweaver インストールにのみ使用されます。SAP NetWeaver 7.50 HA 認定 (NW-HA-CLU-750; ENSA1) に従う実装の ERS インスタンスを管理するために使用するリソースでは、このパラメーターを true に設定する必要があります。これには、ENSA1 を使用する場合の NetWeaver 7.50 未満のシステムも含まれます。
DIR_EXECUTABLE	いいえ	null	sapstartsrv および sapcontrol バイナリーの場所の完全修飾パス (SAP カーネルバイナリーのデフォルトの場所が変更されている場合にのみ必要)。
DIR_PROFILE	いいえ	null	SAP スタートプロファイルの場所の完全修飾パス (インスタンスプロファイルのデフォルトの場所が変更されている場合にのみ必要)。

属性名	必須	デフォルト値	説明
AUTOMATIC_RECOVER	いいえ	false	SAPInstance リソースエージェントは、起動試行が失敗した場合、自動的に再起動を1回試みます。そのために、実行中のインスタンスプロセスを強制終了し、 kill.sap ファイルを削除して、 cleanipc を実行します。SAP インスタンスがクラッシュすると、一部のプロセスや共有メモリーセグメントが残ることがあります。このオプションを true に設定すると、起動操作中に残存物の削除を試みます。
MONITOR_SERVICES	いいえ	disp+work msg_server enserver enrepserver jcontrol jstart	インスタンスの健全性を判断するために監視する必要がある SAP インスタンスサービスのリスト。監視するサービスの増減や、 sapstartsrv がサポートする他のサービスの監視を行うには、このパラメーターを使用してリストを変更します。名前は、コマンド sapcontrol -nr [Instance-Nr] -function GetProcessList の出力で使用される文字列と一致する必要があり、(パイプ) 記号で区切って複数のサービスを指定できます (このパラメーターの値は常に、監視するサービスの完全なリストである必要があります)。

パラメーターの完全なリストは、**pcs resource describe SAPInstance** を実行することで取得できません。

1.4.2. SAPDatabase リソースエージェント

SAPDatabase リソースエージェントを使用すると、SAP NetWeaver ベースの HA クラスターセットアップの一部として、単一の Oracle、IBM DB2、SAP ASE、または MaxDB のデータベースインスタンスを管理できます。詳細は、[Support Policies for RHEL High Availability Clusters - Management of SAP](#)

[NetWeaver in a Cluster](#) を参照し、RHEL 9 でサポートされているデータベースバージョンのリストを確認してください。

SAPDatabase リソースエージェントは、データベースコマンドを直接実行しません。**SAP Host Agent** を使用してデータベースを制御します。したがって、SAP Host Agent を各クラスターノードにインストールする必要があります。

SAPDatabase リソースエージェントは、データベースインスタンスを管理するための基本機能のみを提供します。そのため、データベースインスタンスにさらに多くの HA 機能が必要な場合は、代わりにデータベースの HA 機能 (Oracle RAC や IBM DB2 HA/DR など) を使用することを推奨します。

S/4HANA HA セットアップの場合、HANA システムレプリケーションを使用して、HANA インスタンスの障害に対する堅牢性を高めることを推奨します。HANA システムレプリケーション HA セットアップは、別個のクラスターを使用して実行できます。あるいは、HANA システムレプリケーションセットアップの管理に使用するのと同じ HA クラスターが **ASCS** インスタンスと **ERS** インスタンスを管理する、“コスト最適化”された S/4HANA HA セットアップを使用することもできます。

1.4.2.1. SAPDatabase リソースエージェントの重要なパラメーター

属性名	必須	デフォルト値	説明
SID	はい	null	一意のデータベースシステム識別子 (通常は SAP SID と同一)。
DBTYPE	はい	null	管理するデータベースのタイプ。有効な値は、 ADA (SAP MaxDB)、 DB6 (IBM DB2)、 ORA (Oracle DB)、 SYB (SAP ASE) です。
DBINSTANCE	いいえ	null	データベースインスタンス名が SID と等しくない、特殊なデータベース実装に使用する必要があります (Oracle DataGuard など)。
DBOSUSER	いいえ	ADA=/etc/opt/sdb から取得、DB6= db2SID 、ORA= oraSID および oracle 、SYB= sybSID 、HDB= SIDadm	このパラメーターは、オペレーティングシステムレベルのデータベースプロセスが、使用しているデータベースタイプのデフォルトユーザーで実行されていない場合に設定できます。

属性名	必須	デフォルト値	説明
STRICT_MONITORING	いいえ	false	これは、リソースエージェントがデータベースを監視する方法を制御します。 true に設定すると、 saphostctrl -function GetDatabaseStatus を使用してデータベースの状態をテストします。 false に設定すると、オペレーティングシステムプロセスのみを監視します。
MONITOR_SERVICES	いいえ	Instance Database Listener	STRICT_MONITORING が true に設定されている場合に、 SAPDatabase リソースエージェントが監視するサービスを定義します。サービス名は、 saphostctrl -function GetDatabaseStatus コマンドの出力と一致する必要があります。
AUTOMATIC_RECOVER	いいえ	false	これを true に設定すると、 saphostctrl -function StartDatabase が常に -force オプションを使用して呼び出されます。

パラメーターの完全なリストは、**pcs resource describe SAPDatabase** を実行することで取得できます。

1.5. マルチ SID サポート (オプション)

このドキュメントで説明するセットアップは、同じ HA クラスター内の複数の SAP 環境 (マルチ SID) の **(A)SCS/ERS** インスタンスを管理するために使用することもできます。たとえば、ABAP と Java アプリケーションサーバーインスタンスの両方を含む SAP 製品 (SAP Solution Manager など) は、マルチ SID クラスターの候補となる可能性があります。

ただし、このようなセットアップでは、いくつかの追加の考慮事項に留意する必要があります。

1.5.1. 一意の SID とインスタンス番号

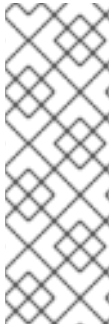
競合を避けるために、**(A)SCS/ERS** インスタンスの各ペアは異なる SID を使用する必要があります、異なる SID に属している場合でも、各インスタンスは一意のインスタンス番号を使用する必要があります。

1.5.2. サイジング

各 HA クラスターノードは、複数のインスタンスをサポートするためのサイジングに関する SAP 要件を満たす必要があります。

1.5.3. インストール

(A)SCS/ERS ペアごとに、セクション 4.5、4.6、および 4.7 に記載されているすべての手順を繰り返してください。各 (A)SCS/ERS ペアは、設定ルールに従って個別にフェイルオーバーします。



注記

RHEL 9 のデフォルトの pacemaker 設定では、リソースアクションの特定の失敗 (リソースの停止の失敗など) により、クラスターノードがフェンスされます。つまり、たとえば、HA クラスターノードの1つの (A)SCS インスタンスでリソースの停止に失敗すると、同じ HA クラスターノードで実行されている他のすべてのリソースが停止します。この動作を変更する方法については、[高可用性クラスターの設定および管理 - 第 21 章 リソースの監視操作](#) に記載されている、操作を監視するための on-fail プロパティの説明を参照してください。

1.6. サポートポリシー

- [Support Policies for RHEL High Availability Clusters - Management of SAP S/4HANA](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP NetWeaver in a Cluster](#)

第2章 要件

2.1. サブスクリプションおよびリポジトリ

すべてのクラスターノードでサブスクリプション、カーネル、およびパッチレベルを同一に保ち、確実に正しいリポジトリを有効にすることが重要です。

RHEL 9 で SAP NetWeaver または SAP S/4HANA アプリケーションサーバーを実行するために必要なサブスクリプションとリポジトリを有効にし、それらを RHEL HA アドオンで管理する方法のガイドラインについては、[RHEL for SAP サブスクリプションおよびリポジトリ](#)を確認してください。

2.2. ストレージ要件

クラスターが管理する SAP S/4HANA インストールで使用するディレクトリは、SAP が提供するガイドラインに従ってセットアップする必要があります。詳細は、[SAP Directories](#)を参照してください。

2.2.1. ローカルディレクトリ

[SAP のガイドンス](#)に従って、`/usr/sap/`、`/usr/sap/SYS/`、および `/usr/sap/<SAPSID>/` ディレクトリを、各ノードでローカルに作成する必要があります。SAP システムのインストール後、`/usr/sap/`には、ノードに固有のファイルおよびディレクトリ (`/usr/sap/sapservices` および `/usr/sap/hostctrl` など) がいくつか追加されます。一方で、`/usr/sap/SYS/`には他のファイルおよびディレクトリへのシンボリックリンクのみが含まれ、`/usr/sap/<SAPSID>/`は、主にインスタンス固有のディレクトリのマウントポイントとして使用されます。

2.2.2. インスタンス固有のディレクトリ

(A)SCS、**ERS**、およびクラスターが管理するその他のアプリケーションサーバーインスタンスの場合、インスタンスを実行するノードのローカルディレクトリとしてクラスターがマウントできる別の SAN LUN または **NFS** エクスポート上に、インスタンス固有のディレクトリを作成する必要があります。以下に例を示します。

- **(A)SCS:** `/usr/sap/<SAPSID>/ASCS<Ins#>/`
- **ERS:** `/usr/sap/<SAPSID>/ERS<Ins#>/`
- アプリケーションサーバー: `/usr/sap/<SAPSID>/D<Ins#>/`

クラスター設定は、インスタンスと仮想 IP の管理に使用するリソースグループの一部として、インスタンスディレクトリのファイルシステムを管理するリソースを含む必要があります。これにより、クラスターが、インスタンスを実行すべきノードに自動的にファイルシステムをマウントできるようになります。

インスタンス固有のディレクトリに SAN LUN を使用する場合は、[HA-LVM](#)を使用して、インスタンスディレクトリを一度に1つのノードにしかマウントできないようにする必要があります。

クラスターがインスタンス自体の起動を試みた時点でファイルシステムが確実にマウントされているように、論理ボリュームを管理するリソース (SAN LUN を使用する場合は) およびファイルシステムは必ず、SAP インスタンスの管理に使用するリソースよりも前に設定する必要があります。

NFSを除き、共有ファイルシステム (GFS2 など) を使用してすべてのインスタンス固有ディレクトリをホストし、それらをすべてのクラスターノードで同時に使用可能にすることは、このドキュメントで説明する解決策ではサポートされません。

特定のディレクトリーに対して **NFS** エクスポートを使用する際に、**NFS** ファイルサーバー (Azure NetApp Files (ANF) や Amazon EFS など) の同じディレクトリーツリーにディレクトリーが作成されている場合は、Filesystem リソースを設定するときにオプション **force_unmount=safe** を使用する必要があります。このオプションを使用すると、クラスターは、エクスポートを作成したディレクトリーツリーで実行中の全プロセスを停止するのではなく、特定の **NFS** エクスポートで実行中のプロセスのみを停止します (詳細は、[During failover of a pacemaker resource, a Filesystem resource kills processes not using the filesystem](#) を参照してください)。

2.2.3. 共有ディレクトリー

次のディレクトリーは、SAP システムの SAP インスタンスを実行するすべてのサーバーで利用可能とする必要があります。

- `/sapmnt/`
- `/usr/sap/trans/`

`/sapmnt/` ディレクトリーは、SAP システムに含まれるサービスを実行している他のすべてのサーバー (たとえば、HANA DB インスタンスをホストしているサーバーや、クラスターが管理していない追加のアプリケーションサーバーをホストしているサーバー) からもアクセスする必要があります。

同じ SAP システムのサービスをホストしているすべてのサーバー間で `/sapmnt/` ディレクトリーと `/usr/sap/trans/` ディレクトリーを共有するには、次のいずれかの方法を使用できます。

- 外部 **NFS** サーバーの使用 ([Support Policies for RHEL High Availability Clusters - Management of Highly Available Filesystem Mounts](#) に記載されているように、同じホストを **NFS** サーバーとしても、その **NFS** サーバーから同じ **NFS** エクスポートをマウント ("ループバックマウント") する **NFS** クライアントとしても使用することはサポートされません)。
- GFS2 ファイルシステムの使用 (この場合、すべてのノードに [Resilient Storage Add-on](#) サブスクリプションが必要です。ノードは、クラスターによって管理されていない SAP インスタンスを実行しているサーバーを含みます)。

共有ディレクトリーは、`/etc/fstab` 経由で静的にマウントすることも、クラスターによってマウントを管理することもできます (この場合、適切な制約を設定して、SAP インスタンスを起動する前に、クラスターが `/sapmnt/` ディレクトリーをクラスターノードにマウントしていることを確認する必要があります)。

2.3. フェンシング/STONITH

[Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH](#) に記載されているように、RHEL HA アドオンを使用した HA クラスターのセットアップが完全にサポートされるようにするには、動作するフェンシング/STONITH デバイスを各クラスターノードで有効にする必要があります。

どのフェンシング/STONITH デバイスを使用するかは、クラスターが実行されているプラットフォームによって異なります。フェンスエージェントに関する推奨事項については、[Support Policies for RHEL High Availability Clusters](#) の [Fencing/STONITH](#) セクションを確認するか、ハードウェアプロバイダーまたはクラウドプロバイダーに問い合わせ、プラットフォームで使用するフェンスデバイスを確認してください。



注記

SAP NetWeaver/S/4HANA アプリケーションサーバーインスタンスを管理するための HA クラスターセットアップのフェンスデバイスとして **fence_scsi/fence_mpath** を使用することは、サポートされません。これらのフェンスデバイスは、読み取りと書き込みのために複数のクライアントが同時にアクセスする共有ストレージを管理するクラスターセットアップにしか使用できないためです ([Support Policies for RHEL High Availability Clusters - fence_scsi and fence_mpath](#) を参照してください)。SAP NetWeaver/S/4HANA を管理する HA クラスターの主な目的は、そのような環境で必要な共有ディレクトリーではなく、SAP アプリケーションサーバーインスタンスを管理することです。そのため、**fence_scsi/fence_mpath** を使用すると、ノードをフェンスする必要がある場合に SAP インスタンスが停止しない可能性があります (**fence_scsi/fence_mpath** は通常、クラスターが管理するストレージデバイスへのアクセスのみをブロックするため)。

2.4. クォーラム

pacemaker は、クラスターが定足数を満たしているかどうかを判断するためのいくつかのビルトインメカニズムを提供します。ただし、状況によっては、“スプリットブレイン”状態が発生した場合にクラスターのどちら側が稼働状態を維持すべきかをクラスターが判断できるように、クラスターのセットアップに“クォーラムデバイス”を追加することが望ましい場合があります。

SAP アプリケーションサーバーインスタンスの管理に使用する HA クラスターセットアップの場合、デフォルトではクォーラムデバイスは必要ありませんが、必要に応じてそのようなセットアップにクォーラムデバイスを追加することができます。

クォーラムデバイスをセットアップするための方法は、設定によって異なります。詳細は、次のガイドラインを参照してください。

- [Design Guidance for RHEL High Availability Clusters - Considerations with qdevice Quorum Arbitration](#)

第3章 SAP アプリケーションサーバーインスタンスのインストール

3.1. このドキュメントで使用する設定オプション

以下は、このドキュメントのインスタンスに使用される設定オプションです。これらのオプションは、ローカルの要件に応じて調整してください。

HA クラスターノードと、HA クラスターによって管理される **(A)SCS** および **ERS** インスタンスの場合、次の値を使用します。

```
1st HA cluster node name:  node1
2nd HA cluster node name:  node2

SID:          S4H

ASCS Instance number:  20
ASCS virtual hostname: s4ascs
ASCS virtual IP address: 192.168.200.101

ERS Instance number:  29
ERS virtual hostname: s4ers
ASCS virtual IP address: 192.168.200.102
```

オプションのプライマリーアプリケーションサーバー (PAS) および追加のアプリケーションサーバー (AAS) インスタンスの場合は、次の値を使用します。

```
PAS Instance number:  21
PAS virtual hostname: s4pas
PAS virtual IP address: 192.168.200.103

AAS Instance number:  22
AAS virtual hostname: s4aas
AAS virtual IP address: 192.168.200.104
```

3.2. SAP インスタンスをインストールするためのクラスターノードの準備

インストールを開始する前に、次のことを確認してください。

- RHEL 9 での SAP アプリケーションサーバーインスタンスの実行に関する SAP および Red Hat の推奨事項に従って、すべての HA クラスターノードに RHEL 9 がインストールおよび設定されている。
- [RHEL for SAP サブスクリプションおよびリポジトリ](#) に記載されているように、RHEL for SAP Applications または RHEL for SAP Solutions サブスクリプションがアクティベートされ、必要なリポジトリがすべての HA クラスターノードで有効になっている。
- 共有ストレージとインスタンスディレクトリが正しいマウントポイントに存在する。
- SAP インスタンスが使用する仮想ホスト名と IP アドレスが双方向で解決でき、仮想 IP アドレスにアクセスできる。

- SAP インスタンスをインストールする各 HA クラスターノードで、SAP インストールメディアにアクセスできる。

これらのセットアップ手順は、Ansible および **rhel-system-roles-sap** システムロールを使用して、部分的に自動化できます。詳細は、[Red Hat Enterprise Linux System Roles for SAP](#) を参照してください。

3.3. SAP インスタンスのインストール

Software Provisioning Manager (SWPM) を使用して、次の順序でインスタンスをインストールします。

- **(A)SCS** インスタンス
- **ERS** インスタンス
- DB インスタンス
- PAS インスタンス
- AAS インスタンス

次のセクションでは、このドキュメントで説明している HA クラスターセットアップで管理する SAP インスタンスをインストールする際に従う必要がある、具体的な推奨事項をいくつか示します。SAP NetWeaver または S/4HANA アプリケーションサーバーインスタンスのインストール方法の詳細は、公式の SAP インストールガイドを確認してください。

3.3.1. node1 への (A)SCS のインストール

SAP インスタンスに必要なローカルディレクトリーとマウントポイントは、**(A)SCS** インスタンスをインストールする HA クラスターノードに作成する必要があります。

```
/sapmnt/  
/usr/sap/  
/usr/sap/SYS/  
/usr/sap/trans/  
/usr/sap/S4H/ASCS20/
```

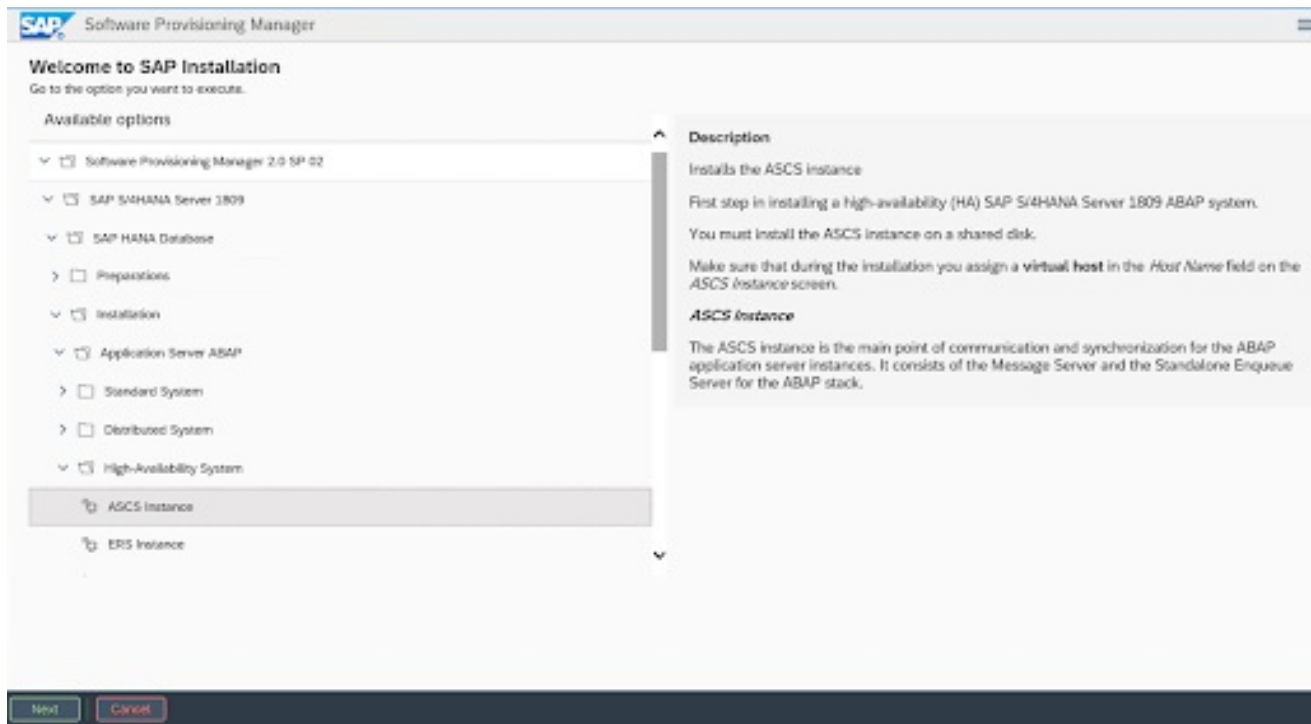
インストールを開始する前に、共有ディレクトリーとインスタンスディレクトリーを手動でマウントする必要があります。

また、**(A)SCS** インスタンスの仮想 IP アドレスを node1 で有効にする必要があります、**ERS** インスタンスの仮想ホスト名が仮想 IP アドレスに解決されることを検証しておく必要があります。

SAP インストーラーを実行するときは、**(A)SCS** インスタンスに使用する仮想ホスト名を必ず指定してください。

```
[root@node1]# ./sapinst SAPINST_USE_HOSTNAME=s4ascs
```

(A)SCS インスタンスのインストールには、**High-Availability System** オプションを選択します。



3.3.2. node2 への ERS のインストール

SAP インスタンスに必要なローカルディレクトリーとマウントポイントは、**ERS** インスタンスをインストールする HA クラスターノードに作成する必要があります。

```
/sapmnt/
/usr/sap/
/usr/sap/SYS/
/usr/sap/trans/
/usr/sap/S4H/ERS29
```

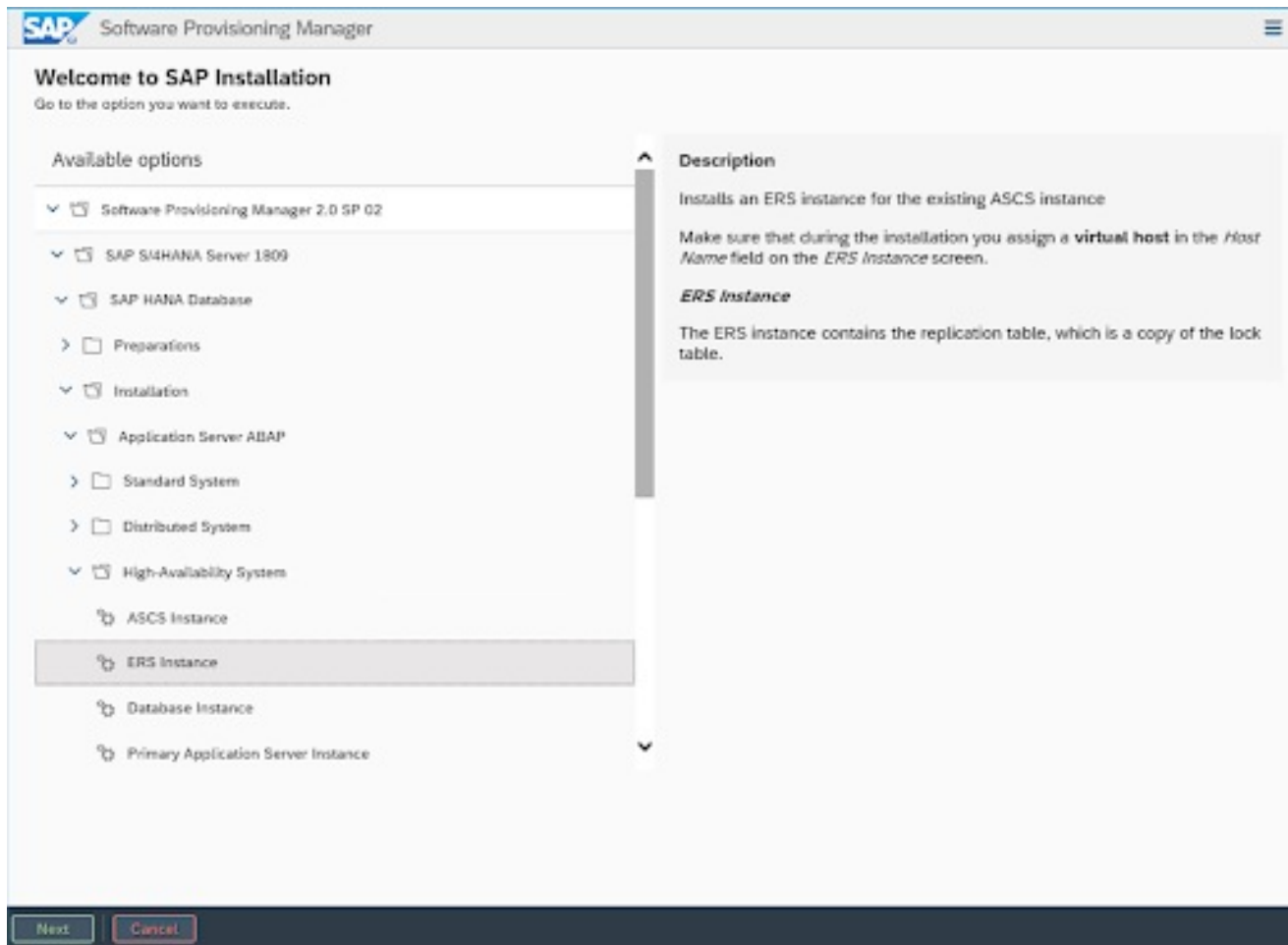
インストールを開始する前に、共有ディレクトリーとインスタンスディレクトリーを手動でマウントする必要があります。

また、**ERS** インスタンスの仮想 IP アドレスを node 2 で有効にする必要があります、**ERS** インスタンスの仮想ホスト名が仮想 IP アドレスに解決されることを検証しておく必要があります。

インストールを開始するときに、**ERS** インスタンスの仮想ホスト名を必ず指定してください。

```
[root@node2]# ./sapinst SAPINST_USE_HOSTNAME=s4ers
```

ERS インスタンスのインストールには、High-Availability System オプションを選択します。



3.3.3. プライマリー/追加のアプリケーションサーバーインスタンスのインストール

SAP インスタンスに必要なローカルディレクトリーとマウントポイントは、プライマリーまたは追加のアプリケーションサーバーインスタンスをインストールする HA クラスターノードに作成する必要があります。

```
/sapmnt/
/usr/sap/
/usr/sap/SYS/
/usr/sap/trans/
/usr/sap/S4H/
/usr/sap/S4H/D<Ins#>
```

インストールを開始する前に、共有ディレクトリーとインスタンスディレクトリーを手動でマウントする必要があります。

また、アプリケーションサーバーインスタンスの仮想 IP アドレスを HA クラスターノードで有効にする必要があります。アプリケーションサーバーインスタンスの仮想ホスト名が仮想 IP アドレスに解決されることを検証しておく必要があります。

インストーラーの起動時にインスタンスの仮想ホスト名を指定します。

```
[root@node<X>]# ./sapinst SAPIINST_USE_HOSTNAME=<virtual hostname of instance>
```

アプリケーションサーバーインスタンスのインストールには、High-Availability System オプションを選択します。

3.4. インストール後の設定

3.4.1. (A)SCS プロファイルの変更

インスタンスはクラスターによって管理されるため、**(A)SCS** インスタンスプロファイルを変更して、インスタンスの **sapstartsrv** プロセスによるエンキューサーバードプロセスの自動再起動を防ぐ必要があります。

(A)SCS インスタンスプロファイルを変更するには、次のコマンドを実行します。

```
[root@node1]# sed -i -e 's/Restart_Program_01/Start_Program_01/'
/sapmnt/S4H/profile/S4H_ASCS20_s4ascs
```

3.4.2. ERS プロファイルの変更

ERS インスタンスはクラスターによって管理されるため、**ERS** インスタンスプロファイルを変更して、インスタンスの **sapstartsrv** によるエンキューレプリケーションサーバードプロセスの自動再起動を防ぐ必要があります。

ERS インスタンスプロファイルを変更するには、次のコマンドを実行します。

```
[root@node2]# sed -i -e 's/Restart_Program_00/Start_Program_00/'
/sapmnt/S4H/profile/S4H_ERS29_s4ers
```

3.4.3. /usr/sap/sapservices ファイルの更新

HA クラスターが管理する SAP インスタンスが HA クラスターの制御外で起動しないようにするには、すべてのクラスターノードの **/usr/sap/sapservices** ファイルで次の行がコメントアウトされていることを確認してください。

```
#LD_LIBRARY_PATH=/usr/sap/S4H/ERS29/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH; /usr/sap/S4H/ERS29/exe/sapstartsrv
pf=/usr/sap/S4H/SYS/profile/S4H_ERS29_s4ers -D -u s4hadm

#LD_LIBRARY_PATH=/usr/sap/S4H/SCS20/exe:$LD_LIBRARY_PATH; export
LD_LIBRARY_PATH; /usr/sap/S4H/SCS20/exe/sapstartsrv
pf=/usr/sap/S4H/SYS/profile/S4H_ASCS20_s4ascs -D -u s4hadm

#LD_LIBRARY_PATH=/usr/sap/S4H/D21/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/S4H/D21/exe/sapstartsrv pf=/usr/sap/S4H/SYS/profile/S4H_D21_s4hpas -D -u s4hadm

#LD_LIBRARY_PATH=/usr/sap/S4H/D22/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/S4H/D22/exe/sapstartsrv pf=/usr/sap/S4H/SYS/profile/S4H_D22_s4haas -D -u s4hadm
```

3.4.4. フェイルオーバーノードでのインスタンス固有ディレクトリーのマウントポイントの作成

インスタンス固有のディレクトリーをマウントするマウントポイントを作成する必要があります。ユーザーとグループの所有権は、すべての HA クラスターノードで **<sid>adm** ユーザーと **sapsys** グループに設定する必要があります。

```
[root@node1]# mkdir /usr/sap/S4H/ERS29/
```

```
[root@node1]# chown s4hadm:sapsys /usr/sap/S4H/ERS29/

[root@node2]# mkdir /usr/sap/S4H/ASCS20
[root@node2]# chown s4hadm:sapsys /usr/sap/S4H/ASCS20

[root@node<x>]# mkdir /usr/sap/S4H/D<Ins#>
[root@node<x>]# chown s4hadm:sapsys /usr/sap/S4H/D<Ins#>
```

3.4.5. すべてのクラスターノードで SAP インスタンスを起動および停止できることの確認

`sapcontrol` を使用して (A)SCS インスタンスと ERS インスタンスを停止し、インスタンス固有のディレクトリーをアンマウントして、他のノードにマウントします。

```
/usr/sap/S4H/ASCS20/
/usr/sap/S4H/ERS29/
/usr/sap/S4H/D<Ins#>/
```

`sapcontrol` を使用したすべての SAP インスタンスの手動での起動および停止が、すべての HA クラスターノードで機能すること、および SAP が提供するツールを使用して SAP インスタンスが正しく実行されていることを確認します。

3.4.6. 正しいバージョンの SAP Host Agent がすべての HA クラスターノードにインストールされていることの確認

各クラスターノードで次のコマンドを実行して、SAP Host Agent のバージョンが同じであり、最小バージョン要件を満たしていることを確認します。

```
[root@node<x>]# /usr/sap/hostctrl/exe/saphostexec -version
```

SAP Host Agent を更新する必要がある場合は、SAP Note [1031096-Installing Package SAPHOSTAGENT](#) を確認してください。

3.4.7. 恒久的な SAP ライセンスキーのインストール

フェイルオーバー後も SAP インスタンスが確実に実行し続けるようにするには、各クラスターノードのハードウェアキーに基づいて複数の SAP ライセンスキーをインストールすることが必要な場合があります。詳細は、SAP Note [1178686 - Linux: Alternative method to generate a SAP hardware key](#) を参照してください。

3.4.8. systemd 対応の SAP インスタンスを使用する場合に必要な追加の変更

クラスターが管理する SAP インスタンスが `systemd` 対応の場合、`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、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。

第4章 クラスターのセットアップ

4.1. 各ノードでの基本的なクラスターインストールの実行

まず pacemaker クラスターをセットアップするには、[RHEL 9 での高可用性クラスターの設定と管理](#)を参照してください。

フェンシング/STONITH セットアップについては、[Support Policies for RHEL High Availability Clusters - General Requirements for Fencing/STONITH](#) のガイドラインに従ってください。さまざまなプラットフォームでサポートされているフェンシング/STONITH エージェントに関する情報は、[Cluster Platforms and Architectures](#) で入手できます。

このガイドの残りの部分では、以下が適切に機能していることを前提とします。

- Pacemaker クラスターがドキュメントに従って設定されており、適切かつ機能するフェンシングを備えている (フェンシングが適切に機能していることを確認する手順は、[How to test fence devices and fencing configuration in a Red Hat High Availability cluster?](#) を参照してください)。
- [Setting up Enqueue Replication Server failover](#) で説明されているように、(A)SCS インスタンスと ERS インスタンス間のエンキューレプリケーションを手動でテストしている。
- すべての HA クラスターノードが [RHEL for SAP Applications](#) または [RHEL for SAP Solutions](#) にサブスクライブされており、[RHEL for SAP サブスクリプションおよびリポジトリ](#) で説明されているように、必要なリポジトリが有効になっている。

4.2. 一般的なクラスタープロパティの設定

リソースの不必要なフェイルオーバーを回避するには、1つのクラスターノードで次のコマンドを実行して、**resource-stickiness** パラメーターと **migration-threshold** パラメーターのデフォルト値を設定します。

```
[root@node1]# pcs resource defaults update resource-stickiness=1
[root@node1]# pcs resource defaults update migration-threshold=3
```



注記

resource-stickiness=1 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=3** は、3回の失敗後にリソースを新しいノードに移動させます。リソースが時期尚早に別のノードにフェイルオーバーするのを防ぐには、通常は3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。

4.3. すべてのクラスターノードへの RESOURCE-AGENTS-SAP パッケージのインストール

SAPInstance リソースエージェントおよび **SAPDatabase** リソースエージェントは、別個の **resource-agents-sap** パッケージで提供されます。次のコマンドを実行して、各 HAcluster ノードにインストールします。

```
[root@node<x>]# dnf install resource-agents-sap
```

4.4. 共有ファイルシステムへのアクセスの設定

SAP インスタンスが動作するには、次の共有ファイルシステムがすべてのクラスターノードで利用可能である必要があります。

```
/sapmnt
/usr/sap/trans
```

共有ファイルシステムは、クラスターによって管理することも、各クラスターノードの `/etc/fstab` に追加することで静的にマウントすることもできます。

4.4.1. クラスターによって管理される共有ファイルシステムの設定

外部 **NFS** サーバーからすべてのクラスターノードに共有ファイルシステムをマウントするには、以下に示すように、Filesystem クラスターリソースのクローンを作成します。

```
[root@node1]# pcs resource create s4h_fs_sapmnt Filesystem device='<NFS_Server>:
<sapmnt_nfs_share>' directory='/sapmnt' fstype='nfs' clone interleave=true
[root@node1]# pcs resource create s4h_fs_sap_trans Filesystem device='<NFS_Server>:
<sap_trans_nfs_share>' directory='/usr/sap/trans' fstype='nfs' clone interleave=true
```

4.4.2. クラスターの外部で管理される共有ファイルシステムの設定

共有ファイルシステムがクラスターによって管理されない場合は、pacemaker サービスを起動する前に、共有ファイルシステムが利用可能であることを確認する必要があります。

共有ファイルシステムがクラスターの外部で管理される場合に、共有ファイルシステムへのアクセスを必要とするリソースをクラスターが起動する前に、共有ファイルシステムを確実にマウントする方法は、[第13章 クラスターリソースの実行順序の決定](#) を参照してください。

4.5. (A)SCS リソースグループの設定

4.5.1. (A)SCS インスタンスの仮想 IP アドレスを管理するリソースの作成

アプリケーションサーバーやその他のクライアントが HA クラスターノードで現在実行中の **(A)SCS** インスタンスに接続できるようにするには、ある HA クラスターノードから別の HA クラスターノードに **(A)SCS** インスタンスが移動する際に、クラスターが、そのインスタンスに割り当てられている仮想 IP アドレスを移動させる必要があります。

このためには、仮想 IP を管理するリソースを、**(A)SCS** インスタンスの管理に使用するリソースグループの一部として作成する必要があります。

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。

物理サーバーまたは仮想マシンでは、**IPAddr2** リソースエージェントを使用してリソースを作成できません。

```
[root@node1]# pcs resource create s4h_vip_asc20 IPAddr2 ip=192.168.200.101 --group
s4h_ASCS20_group
```

4.5.2. (A)SCS インスタンスディレクトリーを管理するリソースの作成

SAP では、インスタンスディレクトリーが、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のみ使用可能であることが必要であるため、インスタンスディレクトリーに使用されるファイルシステムを管理するための HA クラスターリソースをセットアップする必要があります。



注記

インスタンスディレクトリーが **NFS** に格納されている場合でも、HA クラスターが、SAP インスタンスを実行すべき HA クラスターノードにのみ **NFS** エクスポートをマウントできるように、リソースを作成する必要があります。

4.5.2.1. NFS

(A)SCS インスタンスのインスタンスディレクトリーが **NFS** にある場合は、**(A)SCS** インスタンスを管理するリソースグループの一部としてそのインスタンスディレクトリーを管理するリソースを、次のコマンドで作成できます。

```
[root@node1]# pcs resource create s4h_fs_ascscs20 Filesystem device='<NFS_Server>:
<s4h_ascscs20_nfs_share>' directory=/usr/sap/S4H/ASCS20 fstype=nfs force_unmount=safe --group
s4h_ASCS20_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.5.2.2. HA-LVM

HA-LVM を使用して **(A)SCS** インスタンスのインスタンスディレクトリーを管理する場合は、記事 [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#) のガイドラインに従って設定する必要があります。

まず、**LVM-activate** クラスターリソースを追加し、次に Filesystem クラスターリソースを追加する必要があります。

```
[root@node1]# pcs resource create s4h_fs_ascscs20_lvm LVM-activate
volgrpname='<ascscs_volume_group>' vg_access_mode=system_id --group s4h_ASCS20_group

[root@node1]# pcs resource create s4h_fs_ascscs20 Filesystem
device='/dev/mapper/<ascscs_logical_volumes>' directory=/usr/sap/S4H/ASCS20 fstype=ext4 --group
s4h_ASCS20_group
```

4.5.3. (A)SCS インスタンスを管理するためのリソースの作成

```
[root@node1]# pcs resource create s4h_ascscs20 SAPInstance
InstanceName="S4H_ASCS20_rhascs"
START_PROFILE=/sapmnt/S4H/profile/S4H_ASCS20_rhascs AUTOMATIC_RECOVER=false \
  meta resource-stickiness=5000 migration-threshold=1 \
  --group s4h_ASCS20_group \
  op monitor interval=20 on-fail=restart timeout=60 \
  op start interval=0 timeout=600 \
  op stop interval=0 timeout=600
```

resource-stickiness=5000 は、**ERS** リソースによるフェイルオーバー制約のバランスをとるために使用され、リソースが起動したノードに留まり、クラスター内で制御不能に移行しないようにします。

migration-threshold=1 を指定すると、**(A)SCS** インスタンスは、問題が検出されたときに同じ HA クラスターノードで再起動するのではなく、別のノードにフェイルオーバーします。**ENSA2** では同じ HA クラスターノードでの **(A)SCS** インスタンスの再起動が許可されるため、**ENSA2** セットアップでは、このオプションは必要ありません。

リソースグループのすべてのリソースを作成したら、リソーススティックネスをグループに追加して、可能であれば **(A)SCS** インスタンスが HA クラスターノードに留まるようにします。

```
[root@node1]# pcs resource meta s4h_ASCS20_group resource-stickiness=3000
```

4.6. ERS リソースグループの設定

4.6.1. ERS インスタンスの仮想 IP アドレスを管理するためのリソースの作成

ERS インスタンスは、アプリケーションサーバーから直接アクセスされるわけではありませんが、現在 HA クラスターノードで実行中の **ERS** インスタンスに SAP 管理ツールが接続できるように、仮想 IP が必要です。したがって、ある HA クラスターノードから別の HA クラスターノードに **(A)SCS** インスタンスが移動する際は、インスタンスに割り当てられた仮想 IP アドレスをクラスターが移動する必要があります。

このためには、仮想 IP を管理するリソースを、**ERS** インスタンスの管理に使用するリソースグループの一部として作成する必要があります。

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。

物理サーバーまたは仮想マシンでは、**IPaddr2** リソースエージェントを使用してリソースを作成できます。

```
[root@node1]# pcs resource create s4h_vip_ers29 IPaddr2 ip=192.168.200.102 --group s4h_ERS29_group
```

4.6.2. ERS インスタンスディレクトリーを管理するためのリソースの作成

SAP では、インスタンスディレクトリーが、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のみ使用可能であることが必要であるため、インスタンスディレクトリーに使用されるファイルシステムを管理するための HA クラスターリソースをセットアップする必要があります。



注記

インスタンスディレクトリーが **NFS** に格納されている場合でも、HA クラスターが、SAP インスタンスを実行すべき HA クラスターノードにのみ **NFS** エクスポートをマウントできるように、リソースを作成する必要があります。

4.6.2.1. NFS

ERS インスタンスのインスタンスディレクトリーが **NFS** にある場合は、ERS インスタンスを管理するためのリソースグループの一部としてそのインスタンスディレクトリーを管理するリソースを、次のコマンドで作成できます。

```
[root@node1]# pcs resource create s4h_fs_ers29 Filesystem device='<NFS_Server>:  
<s4h_ers29_nfs_share>' directory=/usr/sap/S4H/ERS29 fstype=nfs force_unmount=safe --group
```

```
s4h_ERS29_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.6.2.2. HA-LVM

HA-LVM を使用して **ERS** インスタンスのインスタンスディレクトリーを管理する場合は、記事 [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#) のガイドラインに従って設定する必要があります。

まず、**LVM-activate** クラスタリソースを追加し、次に Filesystem クラスタリソースを追加する必要があります。

```
[root@node1]# pcs resource create s4h_fs_ers29_lvm LVM-activate
volgrpname='<ers_volume_group>' vg_access_mode=system_id --group s4h_ERS29_group

# pcs resource create s4h_fs_ers29 Filesystem device='/dev/mapper/<ers_logical_volume>'
directory=/usr/sap/S4H/ERS29 fstype=ext4 --group s4h_ERS29_group
```

4.6.3. ERS インスタンスを管理するためのリソースの作成

ERS インスタンスクラスタリソースを作成します。

```
[root@node1]# pcs resource create s4h_ers29 SAPInstance InstanceName="S4H_ERS29_rhers"
START_PROFILE=/sapmnt/S4H/profile/S4H_ERS29_rhers AUTOMATIC_RECOVER=false
IS_ERS=true --group s4h_ERS29_group \
  op monitor interval=20 on-fail=restart timeout=60 \
  op start interval=0 timeout=600 \
  op stop interval=0 timeout=600
```



注記

ENSA1 デプロイメントの場合、**IS_ERS=true** 属性は必須です。**IS_ERS** の詳細は、[How does the IS_ERS attribute work on a SAP NetWeaver cluster with Standalone Enqueue Server \(ENSA1 and ENSA2\)?](#) を参照してください。

4.7. 制約の作成

4.7.1. (A)SCS および ERS リソースグループのコロケーション制約を作成する

リソースグループ **s4h_ASCS20_group** と **s4h_ERS29_group** は、同じノードで実行されないようにする必要があります。グループの順序は重要です。

```
[root@node1]# pcs constraint colocation add s4h_ERS29_group with s4h_ASCS20_group -5000
```

4.7.2. (A)SCS リソースの場所の制約を作成する (ENSA1 のみ)

ENSA1 を使用する場合、フェイルオーバーの発生時に、**ERS** インスタンスを実行しているノードに **(A)SCS** インスタンスを確実に移動する必要があります。


```
[root@node1]# pcs constraint location s4h_ascs20 rule score=2000 runs_ers_S4H eq 1
```

4.7.3. (A)SCS および ERS リソースグループの順序の制約を作成する

pacemaker が **ERS** インスタンスを管理するためのリソースグループを停止すると同時に **(A)SCS** インスタンスを管理するためのリソースグループを開始することを決定した場合は、**(A)SCS** インスタンスを管理するためのリソースグループが起動した後に、**ERS** インスタンスを管理するためのリソースグループを停止します。

```
[root@node1]# pcs constraint order start s4h_ASCS20_group then stop s4h_ERS29_group
symmetrical=false kind=Optional
```



注記

symmetrical=false および **kind=Optional** が使用されるため、この制約が有効にならない状況が発生する場合があります。詳細は、[クラスターリソースの実行順序の決定](#) を参照してください。

4.7.4. クラスターが管理する /sapmnt リソースの順序の制約を作成する

共有ファイルシステム **/sapmnt** がクラスターによって管理されている場合、次の制約を使用すると、**(A)SCS** および **ERS** インスタンスの管理に使用するリソースグループは、**/sapmnt** ファイルシステムが使用可能になった後にのみ起動するようになります。

```
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ASCS20_group
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_ERS29_group
```

4.8. データベースインスタンスを管理するためのクラスターリソースグループの設定 (オプション)

レガシーデータベース (Oracle、IBM DB2、SAP ASE、SAP MaxDB など) を引き続き使用している SAP NetWeaver ベースの SAP 製品の管理に HA クラスターを使用する場合、データベースインスタンスをクラスターで管理することもできます。

この章では、**SAPDatabase** リソースエージェント、仮想 IP アドレス、それに必要なファイルシステムを使用して、単一のデータベースインスタンスを管理するためのリソースグループをセットアップする方法の例を示します。

SAPDatabase リソースエージェントは S/4HANA セットアップでは使用できないため、この章で説明するセットアップ例では、S4H ではなく **SAPSID RH1** を使用します。

4.8.1. データベースインスタンスの仮想 IP アドレスを管理するリソースの作成

rh1_SAPDatabase_group の一部となるデータベースインスタンスにアクセスするための仮想 IP アドレスを管理するリソースを作成するには、次のコマンドを使用します。

```
[root]# pcs resource create rh1_vip_db IPAddr2 ip=192.168.200.115 --group
rh1_SAPDatabase_group
```

4.8.2. データベースインスタンスが使用するディレクトリーを管理するリソースの作成

データベースインスタンスが別のシステムで誤って同時に起動してデータが破損するのを避けるために、データベースインスタンスが使用するディレクトリーは、データベースインスタンスを実行する HA クラスターノードにのみマウントできます。

データベースインスタンスが使用するディレクトリーを管理するストレージのセットアップ方法に応じて、データベースディレクトリーを管理するリソースをさまざまな方法で作成する必要があります。



注記

インスタンスディレクトリーが **NFS** に格納されている場合でも、HA クラスターが、データベースインスタンスを実行すべき HA クラスターノードにのみ **NFS** エクスポートをマウントできるように、リソースを作成する必要があります。

4.8.2.1. NFS

データベースインスタンスが使用するディレクトリーが **NFS** にある場合、データベースインスタンスを管理するリソースグループの一部としてそれらのディレクトリーを管理するリソースを、ディレクトリーごとに作成する必要があります。これには、次のコマンドを使用します。

```
[root@node1]# pcs resource create rh1_fs_db Filesystem device='<NFS_Server>:
<rh1_db_nfs_share>' directory=/sapdb/RH1 fstype=nfs force_unmount=safe --group
rh1_SAPDatabase_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.8.2.2. HA-LVM

HA-LVM を使用してデータベースインスタンスが使用するディレクトリーを管理する場合は、記事 [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#) のガイドラインに従って設定する必要があります。

まず、**LVM-activate** クラスターリソースを追加し、次に Filesystem クラスターリソースを追加する必要があります。

```
[root]# pcs resource create rh1_lvm_db LVM-activate volgrpname=vg_db
vg_access_mode=system_id --group rh1_SAPDatabase_group
[root]# pcs resource create rh1_fs_db Filesystem device=/dev/vg_db/lv_db directory=/sapdb/RH1
fstype=xfs --group rh1_SAPDatabase_group
```

データベースディレクトリーに複数のファイルシステムが使用されている場合は、それぞれに個別の Filesystem クラスターリソースを作成する必要があります。

4.8.3. SAPDatabase クラスターリソースの設定

データベースインスタンスに必要な仮想 IP アドレスとファイルシステムのリソースを追加した後、データベースインスタンスを管理する **SAPDatabase** クラスターリソースをリソースグループに追加できます。

```
[root]# pcs resource create rh1_SAPDatabase SAPDatabase DBTYPE="ADA" SID="RH1"
STRICT_MONITORING="TRUE" AUTOMATIC_RECOVER="TRUE" --group
rh1_SAPDatabase_group
```

4.9. PRIMARY/ADDITIONAL APPLICATION SERVER (PAS/AAS) リソースグループの設定 (オプション)

このセクションでは、PAS インスタンスも HA クラスターによって管理する必要がある場合に備えて、Primary Application Server (PAS) インスタンスと、インスタンスディレクトリーの関連する仮想 IP およびファイルシステムとを管理するリソースグループを設定する方法について説明します。HA クラスターが管理する必要のある Additional Application Server (AAS) インスタンスにも、同じ設定を使用できます。

4.9.1. PAS/AAS インスタンスの仮想 IP アドレス を管理するリソースの作成

HA クラスターが管理する PAS/AAS インスタンスに他のアプリケーションサーバーやクライアントが接続できるようにするには、ある HA クラスターノードから別の HA クラスターノードに PAS/AAS インスタンスが移動するときに、クラスターが、そのインスタンスに割り当てられている仮想 IP アドレスを移動させる必要があります。

このためには、仮想 IP を管理するリソースを、PAS/AAS インスタンスの管理に使用するリソースグループの一部として作成する必要があります。

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。

物理サーバーまたは仮想マシンでは、**IPAddr2** リソースエージェントを使用してリソースを作成できます。

```
[root@node1]# pcs resource create s4h_vip_pas_d21 IPAddr2 ip=192.168.200.103 --group
s4h_PAS_D21_group
```

4.9.2. PAS/AAS インスタンスディレクトリーのファイルシステムを管理するリソースの作成

SAP では、インスタンスディレクトリーが、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のみ使用可能であることが必要であるため、インスタンスディレクトリーに使用されるファイルシステムを管理するための HA クラスターリソースをセットアップする必要があります。



注記

インスタンスディレクトリーが **NFS** に格納されている場合でも、HA クラスターが、SAP インスタンスを実行すべき HA クラスターノードにのみ **NFS** エクスポートをマウントできるように、リソースを作成する必要があります。

4.9.2.1. NFS

PAS/AAS インスタンスのインスタンスディレクトリーが **NFS** にある場合は、PAS/AAS インスタンスを管理するリソースグループの一部としてそのインスタンスディレクトリーを管理するリソースを、次のコマンドで作成できます。

```
[root@node1]# pcs resource create s4h_fs_pas_d21 Filesystem device='<NFS_Server>:
<s4h_pas_d21_nfs_share>' directory=/usr/sap/S4H/D21 fstype=nfs force_unmount=safe --group
s4h_PAS21_D21_group \
  op start interval=0 timeout=60 \
  op stop interval=0 timeout=120 \
  op monitor interval=200 timeout=40
```

4.9.2.2. HA-LVM

HA-LVM を使用して PAS/AAS インスタンスのインスタンスディレクトリーを管理する場合は、記事 [What is a Highly Available LVM \(HA-LVM\) configuration and how do I implement it?](#) のガイドラインに従って設定する必要があります。

まず、LVM-activate クラスタリソースを追加し、次に Filesystem クラスタリソースを追加する必要があります。

```
[root@node1]# pcs resource create s4h_lvm_pas_d21 LVM-activate volgrpname=vg_d21
vg_access_mode=system_id --group s4h_PAS_D21_group
[root@node1]# pcs resource create s4h_fs_pas_d21 Filesystem device=/dev/vg_d21/lv_d21
directory=/usr/sap/S4H/D21 fstype=xfst --group s4h_PAS_D21_group
```

4.9.3. PAS/AAS SAPInstance クラスタリソースの設定

pacemaker が PAS または AAS インスタンスを管理できるようにするには、**(A)SCS/ERS** インスタンスと同じ **SAPInstance** リソースエージェントを使用できます。PAS/AAS インスタンスは、**(A)SCS/ERS** インスタンスセットアップと比較して単純なインスタンスで、設定が必要な属性は少なくなります。

以下のコマンドは、**D21** インスタンスの PAS インスタンスを作成し、それを **s4h_PAS_D21_group** リソースグループの最後に配置する方法の例です。

```
[root@node1]# pcs resource create s4h_pas_d21 SAPInstance InstanceName="S4H_D21_s4h-pas"
DIR_PROFILE=/sapmnt/S4H/profile START_PROFILE=/sapmnt/S4H/profile/S4H_D21_s4h-pas --
group s4h_PAS_D21_group
```

4.9.4. 制約の設定

4.9.4.1. PAS/AAS リソースグループの順序の制約を設定する

PAS/AAS インスタンスが適切に起動するには、起動前に **(A)SCS** およびデータベースインスタンスが実行されている必要があります。次のセクションでは、SAP NetWeaver / S/4HANA で使用できるさまざまなタイプのデータベースインスタンスに基づいて、必要な制約をセットアップする方法を示します。

4.9.4.1.1. s4h_SAPDatabase_group を使用したデプロイメント

クラスタリソースグループが1つである設定の場合、データベースに必要なすべてのリソースを起動します。たとえば、ここでは **SAPDatabase** リソースエージェントがデータベースの管理に使用されており、データベースグループ **rh1_SAPDatabase_group** に含まれているとします。以下のコマンドは、**(A)SCS** インスタンスがプロモートされ、データベースグループ **rh1_SAPDatabase_group** が実行されているときにのみ **rh1_PAS_D21_group** 全体を起動する制約を作成します。

```
[root@node1]# pcs constraint order rh1_SAPDatabase_group then rh1_PAS_D21_group
kind=Optional symmetrical=false
[root@node1]# pcs constraint order start rh1_ASCS20_group then rh1_PAS_D21_group
kind=Optional symmetrical=false
```

4.9.4.1.2. データベースとしてシステムレプリケーションを使用する SAP HANA を使用したデプロイメント

クラスタが管理するシステムレプリケーション (SR) 用に設定された SAP HANA データベースを使用

する場合、次の制約を作成すると、**(A)SCS** インスタンスがプロモートされ、SAP HANA **SAPhana_S4H_02-master** がプロモートされたときにのみ、**s4h_PAS_D21_group** グループ全体が起動するようになります。

```
[root@node1]# pcs constraint order promote SAPhana_S4H_02-master then s4h_PAS_D21_group
Kind=Optional symmetrical=false
[root@node1]# pcs constraint order start s4h_ASCS20_group then s4h_PAS_D21_group
Kind=Optional symmetrical=false
```

4.9.4.2. PAS および AAS SAPInstance クラスターリソースのコロケーション制約の設定 (オプション)

両方のノードの実行中に PAS インスタンスと AAS インスタンスが同じノードで実行されないようにするには、以下のコマンドを使用して負のコロケーション制約を追加します。

```
[root@node1]# pcs constraint colocation add s4h_AAS_D22_group with s4h_PAS_D21_group
score=-1000
```

スコアを **-1000** に指定すると、使用可能なノードが1つだけの場合でも、当該残りの1つのノードで PAS/AAS インスタンスが実行され続けます。このような状況で、AAS インスタンスを停止しておく必要がある場合は、**score=-INFINITY** を使用してこの条件を強制できます。

4.9.4.3. クラスターが管理する /sapmnt リソースの順序の制約を作成する

共有ファイルシステム **/sapmnt** がクラスターによって管理されている場合、次の制約を使用すると、PAS/AAS インスタンスの管理に使用するリソースグループは、**/sapmnt** ファイルシステムが使用可能になった後にのみ起動するようになります。

```
[root@node1]# pcs constraint order s4h_fs_sapmnt-clone then s4h_PAS_D21_group
```

4.10. STANDALONE ENQUEUE SERVER 2 (ENSA2) MULTI-NODE CLUSTER (オプション)

ENSA2 を備えた SAP S/4HANA の場合、3 つ以上の HA クラスターノードを使用して **ASCS** および **ERS** インスタンスを管理できます。インスタンスを実行中のノードに問題が発生した際にインスタンスのフェイルオーバーをより柔軟に行うために、クラスターノードを追加する必要がある場合は、次のセクションのガイドラインを使用してください。

4.10.1. OS 設定

リソース、サブスクリプション、OS 設定などの点で、最初の 2 つのノードと同一のノードを作成します。

この例では、ノードのホスト名は node3 です。各クラスターノードの `/etc/hosts` ファイルに、すべてのクラスターノードのホスト名と IP アドレス、さらに HA クラスターが管理するすべての SAP インスタンスの仮想ホスト名と仮想 IP アドレスが含まれていることを確認してください。

`/etc/services` の SAP 関連エントリを、最初の 2 つのノードの 1 つから 3 番目のノードに必ずコピーしてください。

4.10.2. ユーザーとグループの作成

SAP インスタンスに必要なユーザーとグループを作成します。ユーザーとグループは、他のノードで使用されているものと同じです。以下に例を示します。

```
Groups in /etc/group:
sapsys:x:1010:
sapinst:x:1011:root,s4hadm

Users in /etc/passwd:
s4hadm:x:1020:1010:SAP System Administrator:/home/s4hadm:/bin/csh
sapadm:x:1001:1010:SAP System Administrator:/home/sapadm:/bin/false
```

4.10.3. 共有ファイルシステムのローカルディレクトリーとマウントポイントの作成

追加の HA クラスターノードで実行可能にする必要のある全インスタンスに必要な、すべてのマウントポイントを作成します。

```
/sapmnt
/usr/sap/
/usr/sap/SYS/
/usr/sap/trans/
/usr/sap/S4H/
/usr/sap/S4H/ASCS20/
/usr/sap/S4H/ERS29/
/usr/sap/S4H/D<Ins#>/
```

すべてのディレクトリーのユーザーおよびグループの所有権は、必ず他のクラスターノードと同じユーザーおよびグループに設定し、ローカルディレクトリー (**/usr/sap/SYS** など) の内容は、必ず他のクラスターノードの1つからコピーしてください。

/sapmnt および **/usr/sap/trans** が既存の HA クラスターノードに **/etc/fstab** 経由で静的にマウントされている場合、これらのファイルシステムも追加の HA クラスターノードの **/etc/fstab** に追加する必要があります。また、ファイルシステムは後でマウントする必要があります。

/sapmnt および **/usr/sap/trans** がクラスターによって管理されている場合は、ファイルシステムが追加の HA クラスターノードにもマウントされるようにクラスター設定を更新する必要があります。

4.10.4. SAP インスタンスを管理するための RHEL HA アドオンとリソースエージェントのインストール

ノードをクラスターの一部とし、SAP インスタンスを管理できるようにするには、必要なパッケージをインストールします。

```
[root@node3]# dnf install pcs pacemaker resource-agents-sap
```

4.10.5. クラスターへのノードの追加

既存のクラスターの1つのノードに3番目のノードを追加します。

```
[root@node1]# pcs cluster auth node3
Username: hacluster
Password:

[root@node1]# pcs cluster node add node3
```

4.10.6. 3番目のノードを含めるためのフェンシング/STONITH設定の更新

STONITHセットアップによっては、3番目のHAクラスターノードを含むようにSTONITHリソースを更新する必要がある場合があります。

新しいHAクラスターノードにリソースを移動する前に、次のコマンドを使用して、既存のHAクラスターノードの1つからHAクラスターの新しいノードをフェンスできることを確認してください。

```
[root@node1]# pcs stonith fence node3
```

4.10.7. ERS リソース設定の更新

ERS インスタンスが起動したノードに留まり、クラスター内で制御不能に移行しないようにするには、リソースの **resource-stickiness** を設定します。

```
[root@node1]# pcs resource meta s4h_ers29 \ resource-stickiness=3000
```

4.11. SAP HA インターフェイスを有効にして、クラスターが制御する SAP インスタンスを SAP 管理ツールで管理できるようにする方法 (オプション)

SAP 管理者が、[SAP Landscape Management \(LaMa\)](#) などのツールを使用して、このドキュメントで説明している HA クラスターセットアップが制御する SAP アプリケーションサーバーインスタンスを管理できるようにするには、HA クラスターが管理する各 SAP アプリケーションサーバーインスタンスで [SAP HA インターフェイス](#) を有効にする必要があります。こうすることで、SAP 管理ツールで行われる操作であって、SAP インスタンスの管理に使用するクラスターリソースに影響する操作について、HA クラスターが認識できるようにします (たとえば、HA クラスターは、管理している SAP アプリケーションサーバーインスタンスが SAP LaMa 経由で起動または停止する場合、通知を受け取る必要があります)。

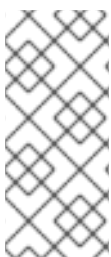
SAP HA インターフェイスの設定方法については、[How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#) を参照してください。

4.12. システム起動時のクラスターの自動起動を有効にする方法 (オプション)

デフォルトでは、HA クラスターは OS の起動時に自動起動できません。クラスターノードがフェンスされて再起動した後は、HA クラスターを手動で起動する必要があります。

すべてのクラスターノードのすべてのクラスターコンポーネントの自動起動は、次のコマンドで有効にできます。

```
[root@node1]# pcs cluster enable --all
```



注記

状況によっては、ノードの再起動後にクラスターを自動起動しない方が有益な場合があります。たとえば、クラスターリソースに必要なファイルシステムに問題がある場合、ファイルシステムを再度使用するには、まずそのファイルシステムを修復する必要があります。クラスターを自動起動しようとした際に、ファイルシステムが機能せず起動に失敗すると、さらに問題が発生する可能性があります。

第5章 クラスター設定のテスト

HA クラスターのセットアップを実稼働環境に導入する前に、次のテストを実行して、HA クラスターのセットアップが期待どおりに機能することを確認するよう推奨します。

また、これらのテストは、定期的な HA/DR 訓練の一環として後から繰り返し行う必要があります。それにより、クラスターが引き続き期待どおりに機能していること、また通常の動作中に問題が発生した場合や、セットアップの手動メンテナンスが必要な場合に、管理者がセットアップを健全な状態に戻すために必要な手順を熟知していることを確認します。

5.1. PCS コマンドを使用した ASCS インスタンスの手動移動

pacemaker クラスターが、インスタンスを他の HA クラスターノードにオンデマンドで移動できることを確認します。

- テストの前提条件
 - 両方のクラスターノードが稼働しており、**ASCS** と **ERS** のリソースグループが異なる HA クラスターノードで実行されている。

```
* Resource Group: S4H_ASCS20_group:
* S4H_lvm_ascs20 (ocf:heartbeat:LVM-activate): Started node1
* S4H_fs_ascs20 (ocf:heartbeat:Filesystem): Started node1
* S4H_vip_ascs20 (ocf:heartbeat:IPAddr2): Started node1
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* Resource Group: S4H_ERS29_group:
* S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node2
* S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node2
* S4H_vip_ers29 (ocf:heartbeat:IPAddr2): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。
- テストの手順
 - 任意のノードから次のコマンドを実行して、他の HA クラスターノードへの **ASCS** インスタンスの移動を開始します。

```
[root@node1]# pcs resource move S4H_ascs20
```

- モニタリング
 - テスト中に別のターミナルで次のコマンドを実行します。

```
[root@node2]# watch -n 1 pcs status
```

- 予想される動作
 - **ASCS** リソースグループが他のノードに移動されます。
 - その後、**ERS** リソースグループが停止し、**ASCS** リソースグループが以前に実行されていたノードに移動します。
- テスト結果

- **ASCS** リソースグループが他のノード (このシナリオではノード node2) に移動し、**ERS** リソースグループがノード node1 に移動します。

```
* Resource Group: S4H_ASCS20_group:
* S4H_lvm_asc20 (ocf:heartbeat:LVM-activate): Started node2
* S4H_fs_asc20 (ocf:heartbeat:Filesystem): Started node2
* S4H_vip_asc20 (ocf:heartbeat:IPAddr2): Started node2
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node2
* Resource Group: S4H_ERS29_group:
* S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node1
* S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node1
* S4H_vip_ers29 (ocf:heartbeat:IPAddr2): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- 復元の手順:
 - 場所の制約がある場合は、それを削除します。

```
[root@node1]# pcs resource clear S4H_asc20
```

5.2. SAPCONTROL を使用した ASCS インスタンスの手動移動 (SAP HA インターフェイスが有効な場合)

インスタンスに対して **SAP HA インターフェイス** が有効になっている場合に、**sapcontrol** コマンドがインスタンスを他の HA クラスタードに移動できることを確認します。

- テストの前提条件
 - SAP インスタンスに対して **SAP HA インターフェイス** が有効になっている。
 - 両方のクラスタードが稼働しており、**ASCS** と **ERS** のリソースグループが実行されている。

```
[root@node2: ~]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。
- テストの手順
 - <sid>adm ユーザーとして、**sapcontrol** の **HAFailoverToNode** 関数を実行して、**ASCS** インスタンスを他のノードに移動します。
- モニタリング
 - テスト中に別のターミナルで次のコマンドを実行します。

```
[root@node2]# watch -n 1 pcs status
```

- 予想される動作
 - **ASCS** インスタンスは他の HA クラスタードに移動し、移動を完了するために一時的な場所の制約を作成します。

- テスト

```
[root@node2]# su - s4hadm
node2:s4hadm 52> sapcontrol -nr 20 -function HAFailoverToNode ""

06.12.2023 12:57:04
HAFailoverToNode
OK
```

- テスト結果

- **ASCS** と **ERS** は、両方とも他のノードに移動します。

```
[root@node2]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- 制約は次のように作成されます。

```
[root@node1]# pcs constraint
Location Constraints:
Resource: S4H_ASCS20_group
Constraint: cli-ban-S4H_ASCS20_group-on-node2
Rule: boolean-op=and score=-INFINITY
Expression: #uname eq string node1
Expression: date lt xxxx-xx-xx xx:xx:xx +xx:xx
```

- 復元の手順

- 上記の制約は、Expression で指定された **date lt** に達すると自動的に消去されます。
- あるいは、次のコマンドを使用して制約を削除することもできます。

```
[root@node1]# pcs resource clear S4H_asc20
```

5.3. ASCS インスタンスの障害のテスト

ASCS インスタンスのエンキューサーまたは **ASCS** インスタンス全体に障害が発生したときに、pacemaker クラスタが必要なアクションを実行することを確認します。

- テストの前提条件

- 両方のクラスターノードが稼働しており、**ASCS** と **ERS** のリソースグループが実行されています。

```
[root@node2]# pcs status | egrep -e "S4H_asc20|S4H_ers29"
* S4H_asc20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。

- テストの手順

- **ASCS** が実行されているノードのエンキューサーバーの **PID** を特定します。
- 識別されたプロセスに **SIGKILL** シグナルを送信します。
- モニタリング
 - テスト中に別のターミナルで次のコマンドを実行します。

```
[root@node2]# watch -n 1 pcs status
```

- 予想される動作
 - エンキューサーブプロセスが強制終了されます。
 - pacemaker クラスタは、設定に従って必要なアクションを実行します。この場合、**ASCS** を他のノードに移動します。
- テスト
 - **ASCS** が実行されているノード上の **<sid>adm user** に切り替えます。

```
[root@node1]# su - s4hadm
```

- en.sap(NetWeaver) enq.sap(S/4HANA) の PID を特定します。

```
node1:s4hadm 51> pgrep -af "(en|enq).sap"
31464 enq.sapS4H_ASCS20 pf=/usr/sap/S4H/SYS/profile/S4H_ASCS20_s4ascs
```

- 特定されたプロセスを強制終了します。

```
node1:s4hadm 52> kill -9 31464
```

- クラスタの **Failed Resource Actions** に注目してください。

```
[root@node2]# pcs status | grep "Failed Resource Actions" -A1
Failed Resource Actions:
* S4H_ascs20 2m-interval monitor on node1 returned 'not running' at Wed Dec 6
15:37:24 2023
```

- **ASCS** と **ERS** は、他のノードに移動します。

```
[root@node2]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ascs20 2m-interval monitor on node1 returned 'not running' at Wed Dec 6
15:37:24 2023
```

- 復元の手順
 - 失敗したアクションをクリアします。

```
[root@node2]# pcs resource cleanup S4H_ascs20
...
Waiting for 1 reply from the controller
... got reply (done)
```

■

5.4. ERS インスタンスのテストの失敗

ASCS インスタンスのエンキューレプリケーションサーバー (**ERS**) に障害が発生したときに、pacemaker クラスターが必要なアクションを実行することを確認します。

- テストの前提条件
 - 両方のクラスターノードが稼働しており、**ASCS** と **ERS** のリソースグループが実行されています。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。
- テストの手順
 - **ERS** インスタンスが実行されているノードのエンキューレプリケーションサーバープロセスの PID を特定します。
 - 識別されたプロセスに SIGKILL シグナルを送信します。
- モニタリング
 - テスト中に別のターミナルで次のコマンドを実行します。

```
[root@node2]# watch -n 1 pcs status
```

- 予想される動作
 - エンキューレプリケーションサーバープロセスが強制終了します。
 - Pacemaker クラスターは、設定に従って必要なアクションを実行します。この場合、同じノード上で **ERS** インスタンスを再起動します。

- テスト
 - <sid>adm ユーザーに切り替えます。

```
[root@node1]# su - s4hadm
```

- **enqr.sap** の PID を特定します。

```
node1:s4hadm 56> pgrep -af enqr.sap
532273 enqr.sapS4H_ERS29 pf=/usr/sap/S4H/SYS/profile/S4H_ERS29_s4ers
```

- 特定されたプロセスを強制終了します。

```
node1:s4hadm 58> kill -9 532273
```

- クラスターの "Failed Resource Actions" に注目してください。

```
[root@node1]# pcs status | grep "Failed Resource Actions" -A1
Failed Resource Actions:
* S4H_ers29 2m-interval monitor on node1 returned 'not running' at Thu Dec 7
13:15:02 2023
```

- **ERS** は、他のノードですでに実行されている **ASCS** を妨げることなく、同じノードで再起動します。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 2m-interval monitor on node1 returned 'not running' at Thu Dec 7
13:15:02 2023
```

- 復元の手順

- 失敗したアクションをクリアします。

```
[root@node1]# pcs resource cleanup S4H_ers29
...
Waiting for 1 reply from the controller
... got reply (done)
```

5.5. ノードのクラッシュによる **ASCS** インスタンスのフェイルオーバー

ノードがクラッシュした場合に **ASCS** インスタンスが正しく移動することを確認します。

- テストの前提条件

- 両方のクラスターノードが稼働しており、**ASCS** と **ERS** のリソースグループが実行されています。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。

- テストの手順

- **ASCS** が実行されているノードをクラッシュします。

- モニタリング

- テスト中に、他のノードの別のターミナルで次のコマンドを実行します。

```
[root@node1]# watch -n 1 pcs status
```

- 予想される動作

- **ASCS** が実行されているノードがクラッシュし、設定に従ってシャットダウンまたは再起動されます。
- その間、**ASCS** は他のノードに移動します。

- **ERS** は、以前にクラッシュしたノードがオンラインに戻った後、そのノード上で起動します。
- テスト
 - **ASCS** が実行されているノードで、root ユーザーとして次のコマンドを実行します。

```
[root@node2]# echo c > /proc/sysrq-trigger
```

- **ASCS** は他のノードに移動します。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node1
```

- **ERS** は停止し、オンラインに戻ると以前にクラッシュしたノードに移動します。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Stopped
```

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- 復元の手順
 - 失敗したアクションがある場合はクリーンアップします。

```
[root@node1]# pcs resource cleanup
```

5.6. ノードのクラッシュによる ERS インスタンスの障害

ERS インスタンスが同じノードで再起動されることを確認します。

- テストの前提条件
 - 両方のクラスターノードが稼働しており、**ASCS** と **ERS** のリソースグループが実行されています。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。
- テストの手順
 - **ERS** が実行されているノードをクラッシュします。
- モニタリング
 - テスト中に、他のノードの別のターミナルで次のコマンドを実行します。

```
[root@nod1]# watch -n 1 pcs status
```

- 予想される動作
 - **ERS** が実行されているノードがクラッシュし、設定に従ってシャットダウンまたは再起動します。
 - その間、**ASCS** は、他のノードに対して実行を続けます。**ERS** は、クラッシュしたノードがオンラインに戻った後、そのノード上で再起動します。

- テスト

- **ERS** が実行されているノードで、root ユーザーとして次のコマンドを実行します。

```
[root@node2]# echo c > /proc/sysrq-trigger
```

- **ERS** は、テスト中に **ASCS** インスタンスを妨げることなく、クラッシュしたノードがオンラインに戻った後、そのノード上で再起動します。

```
[root@node1]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- 復元の手順

- 失敗したアクションがあればクリーンアップします。

```
[root@node2]# pcs resource cleanup
```

5.7. ノードのクラッシュによる ASCS インスタンスの障害 (ENSA2)

3 ノードの ENSA 2 クラスタ環境の場合、任意のインスタンスのフェイルオーバーイベント中に 3 番目のノードが考慮されます。

- テストの前提条件

- **ASCS** および **ERS** のリソースグループが実行されている 3 ノードの SAP S/4HANA クラスタ。
- 3 番目のノードはすべてのファイルシステムにアクセスでき、最初の 2 つのノードと同じ方法で、必要なインスタンス固有の IP アドレスをプロビジョニングできます。
- このセットアップ例では、基礎となる共有 **NFS** ファイルシステムは次のとおりです。

```
Node List:
* Online: [ node1 node2 node3 ]

Active Resources:
* s4r9g2_fence (stonith:fence_rhevm): Started node1
* Clone Set: s4h_fs_sapmnt-clone [fs_sapmnt]:
* Started: [ node1 node2 node3 ]
* Clone Set: s4h_fs_sap_trans-clone [fs_sap_trans]:
* Started: [ node1 node2 node3 ]
* Clone Set: s4h_fs_sap_SYS-clone [fs_sap_SYS]:
* Started: [ node1 node2 node3 ]
```

```
* Resource Group: S4H_ASCS20_group:
* S4H_lvm_ascs20 (ocf:heartbeat:LVM-activate): Started node1
* S4H_fs_ascs20 (ocf:heartbeat:Filesystem): Started node1
* S4H_vip_ascs20 (ocf:heartbeat:IPAddr2): Started node1
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node1
* Resource Group: S4H_ERS29_group:
* S4H_lvm_ers29 (ocf:heartbeat:LVM-activate): Started node2
* S4H_fs_ers29 (ocf:heartbeat:Filesystem): Started node2
* S4H_vip_ers29 (ocf:heartbeat:IPAddr2): Started node2
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- リソースおよびリソースグループのすべての障害がクリアされ、failcount がリセットされている。
- テストの手順
 - **ASCS** が実行されているノードをクラッシュします。
- モニタリング
 - テスト中に、**ASCS** グループが現在実行されていないノードの1つで、別のターミナルから次のコマンドを実行します。

```
[root@node2]# watch -n 1 pcs status
```

- 予想される動作
 - **ASCS** は、3番目のノードに移動します。
 - **ERS** は、すでに実行されているのと同じノード上で引き続き実行されます。
- テスト
 - **ASCS** グループが現在実行されているノードをクラッシュさせます。

```
[root@node1]# echo c > /proc/sysrq-trigger
```

- **ASCS** は、2番目のノードですすでに実行されている **ERS** インスタンスを妨げることなく、3番目のノードに移動します。

```
[root@node2]# pcs status | egrep -e "S4H_ascs20|S4H_ers29"
* S4H_ascs20 (ocf:heartbeat:SAPInstance): Started node3
* S4H_ers29 (ocf:heartbeat:SAPInstance): Started node2
```

- 復元の手順
 - 失敗したアクションがあればクリーンアップします。

```
[root@node2]# pcs resource cleanup
```


第6章 メンテナンス手順

6.1. RHEL と RHEL HA アドオンの更新

詳細は、[Recommendations: Applying package updates in a RHEL High Availability cluster](#) を参照してください。



注記

2 ノードのクラスターセットアップの場合、HA クラスターノードをスタンバイモードにする前に、他の HA クラスターノードにリソースを手動で移動する必要はありません (HA クラスターノードをスタンバイモードにすることで、HA クラスター設定に基づいて HA クラスターノード上で実行されているリソースの移動または停止が行われます)。

また、SAP システムのダウンタイムを最小限に抑えるために、“それほど重要ではない” リソース (**ERS** インスタンスなど) を実行する HA クラスターノードを最初に更新することを推奨します。その HA クラスターノードが更新され、更新の開始前にノードで実行されていたリソースが再び実行されていると確認された場合、“重要な” リソース (**(A)SCS** インスタンスなど) を実行する他の HA クラスターノードも更新できます。

第7章 参考資料

7.1. RED HAT

- [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 S/4HANA](#)
- [Support Policies for RHEL High Availability Clusters - Management of SAP NetWeaver 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](#)
- [How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#)
- [How to manage standalone SAP Web Dispatcher instances using the RHEL HA Add-On](#)
- [The Systemd-Based SAP Startup Framework](#)

7.2. SAP

- [SAP Note 1552925 - Linux: High Availability Cluster Solutions](#)
- [SAP Note 1693245 - SAP HA Script Connector Library](#)
- [SAP Note 1908655 - Support details for Red Hat Enterprise Linux HA Add-On](#)
- [SAP Note 2630416 - Support for Standalone Enqueue Server 2](#)
- [SAP Note 2641322 - Installation of ENSA2 and update from ENSA1 to ENSA2 when using the Red Hat HA solutions for SAP](#)
- [SAP Note 3108316 - Red Hat Enterprise Linux 9.x: Installation and Configuration](#)
- [Standalone Enqueue Server | SAP Help Portal](#)
- [Setting up Enqueue Replication Server Fail over | SAP Blogs](#)
- [High Availability with the Standalone Enqueue Server](#)
- [Evolution of ENSA2 and ERS2... | SAP Blogs](#)