



# Red Hat Enterprise Linux for SAP Solutions 8

RHEL HA アドオンを使用して、コスト最適化された SAP S/4HANA HA クラスタ (HANA システムレプリケーション + ENSA2) を設定する



Red Hat Enterprise Linux for SAP Solutions 8 RHEL HA アドオンを使用して、コスト最適化された SAP S/4HANA HA クラスタ (HANA システムレプリケーション + ENSA2) を設定する

---

## 法律上の通知

Copyright © 2023 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

SAP S/4HANA 1809 以降の HANA システムレプリケーションとスタンドアロンエンキューサーバー 2 (ENSA2) の両方を管理するための 2 ノード HA クラスターのセットアップ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. 概要 .....	5
1.2. 対象者 .....	6
1.3. コンセプト .....	6
1.4. サポートポリシー .....	6
<b>第2章 要件</b> .....	<b>7</b>
2.1. サブスクリプション .....	7
2.2. PACEMAKER リソースエージェント .....	7
2.3. 2 ノードクラスター環境 .....	7
2.4. ストレージ要件 .....	8
<b>第3章 SAP S/4HANA をインストールする</b> .....	<b>10</b>
3.1. このドキュメントで使用する設定オプション .....	10
3.2. ホストの準備 .....	10
3.3. S/4HANA をインストールする .....	10
3.4. インストール後の設定 .....	13
<b>第4章 PACEMAKER をインストールする</b> .....	<b>15</b>
4.1. 一般的なクラスタープロパティを設定する .....	15
4.2. すべてのクラスターノードに RESOURCE-AGENTS-SAP をインストールします .....	15
4.3. 共有ファイルシステムのクラスターリソースを設定する .....	16
4.4. ASCS リソースグループの設定 .....	16
4.5. ERS リソースグループの設定 .....	17
4.6. 制約を作成する .....	18
<b>第5章 クラスター設定をテストする</b> .....	<b>20</b>
5.1. 制約を確認してください .....	20
5.2. ノードクラッシュによるフェイルオーバー ASCS .....	20
5.3. ERS は以前に障害が発生したノードに移動します .....	21
<b>第6章 再起動後にクラスターを自動起動できるようにする</b> .....	<b>22</b>
<b>第7章 フェイルオーバーのテスト</b> .....	<b>23</b>
7.1. ノードクラッシュによるフェイルオーバー ASCS .....	23
7.2. ERS は現在のノードのまま .....	23
7.3. ERS クラッシュのテスト .....	23
7.4. コストが最適化された SAP S/4HANA クラスター環境は、次のようになります。 .....	23
<b>第8章 オプション - SAP 管理ツールを使用して、クラスター制御の ASCS/ERS インスタンスを管理するための SAP HA インターフェイスを有効にします。</b> .....	<b>25</b>



## 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. [このリンク](#) をクリックしてフィードバックを提供します。
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. 今後の更新について通知を受けたい場合は、必ず **Reporter** として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

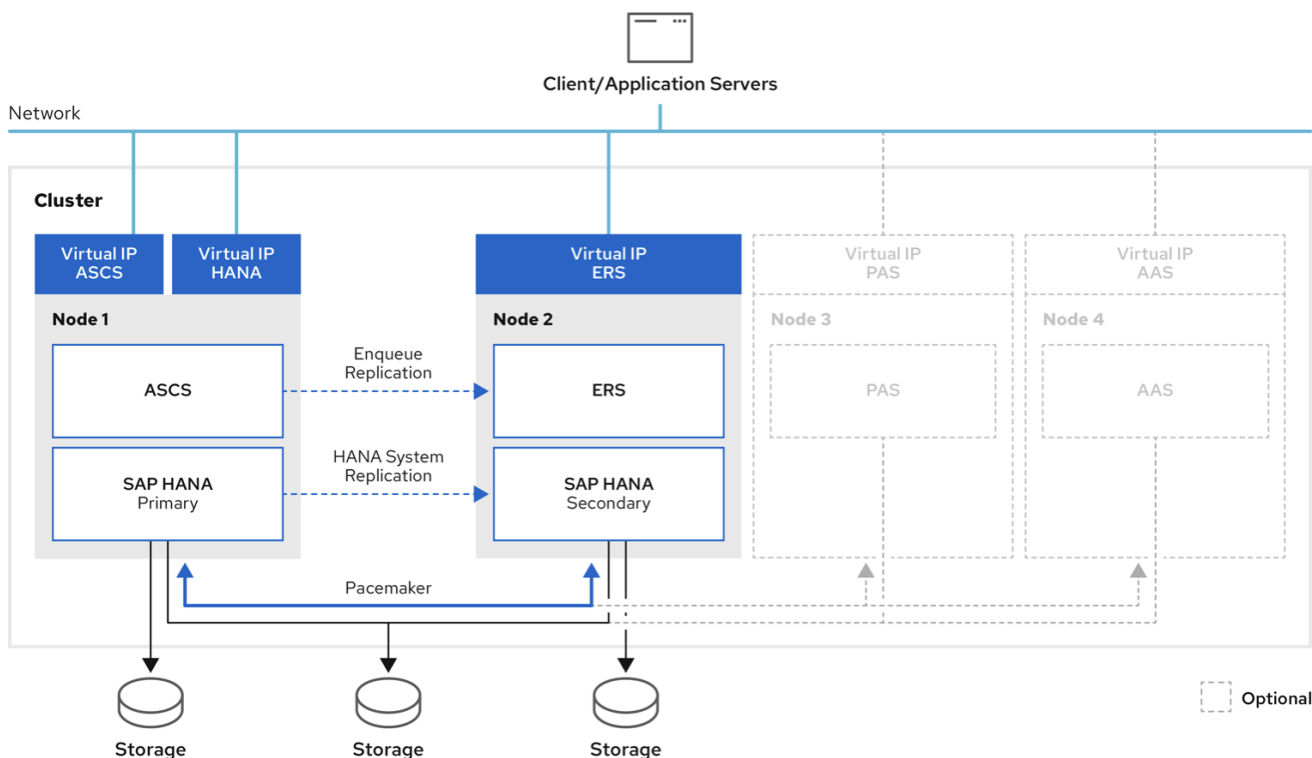
SAP S/4HANA システムのコスト最適化デプロイメントは、S/4HANA 移行シナリオで重要な役割を果たし、特にノードの追加に関連するコストを節約します。このようなシステムを高可用性にすることも重要です。その場合、制約を正しく設定する必要があります。

SAP S/4HANA High-Availability の典型的なコスト最適化セットアップは、2つの異なるコンポーネントで設定されます。

- SAP S/4HANA ASCS/ERS クラスタリソース
- データベース: システムレプリケーションを使用する SAP HANA
  - 詳細は、[Pacemaker での自動化された HANA システムレプリケーションの設定](#) を参照してください。

この記事では、SAP HANA システムレプリケーションと ASCS および ERS インスタンスの両方が単一のクラスターによって管理される SAP S/4HANA HA 環境のセットアップに焦点を当てています。これは、RHEL HA アドオンと、RHEL for SAP ソリューションの一部として利用可能な HA solutions for SAP を使用して行われます。

**注記:** 以下は、この記事で焦点を当てている 2 ノードクラスターセットアップのインストール例のアーキテクチャ図です。システムレプリケーションを使用した追加の SAP HANA インスタンスの設計と設定に関する別のセクションがあります。ASCS または SAP HANA のプライマリインスタンスは、互いに独立して他のノードにフェイルオーバーできることに注意してください。



## 1.2. 対象者

このドキュメントは、Red Hat Enterprise Linux (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高可用性ソリューションを設定した経験がある、SAP および Red Hat 認定またはトレーニング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。ソフトウェアと追加のドキュメントをダウンロードするには、SAP Service Marketplace と Red Hat Customer Portal 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ーをセットアップし、ソリューションをカスタマイズするには、Red Hat プロフェッショナルサービスを強く推奨します。

## 1.3. コンセプト

このドキュメントでは、SAP と Red Hat によって確立された高可用性ガイドラインに準拠する、コストが最適化された 2 ノードクラスターソリューションをセットアップする方法について説明します。これは、SAP S/4HANA 1809 以降のデフォルトインストールである Standalone Enqueue Server 2 (ENSA2) に基づいており、RHEL 8 for SAP Solutions 以降に加えて、完全に自動化されたフェイルオーバーをサポートするスケールアップ SAP HANA インスタンスを強調しています。SAP HANA システムレプリケーションを使用します。SAP によると、ENSA2 は Standalone Enqueue Server 1 (ENSA1) の後継です。これは SAP ロックコンセプトのコンポーネントであり、ロックテーブルを管理します。この原則により、ABAP システム内のデータの一貫性が確保されます。ENSA1 でのフェイルオーバー中、ASCS インスタンスはエンキューレプリケーションサーバー (ERS) をフォローする必要があります。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 ASCS インスタンスを開始する必要がありました。ENSA1 とは対照的に、新しい ENSA2 モデルと Enqueue Replicator 2 にはこれらの制限がなくなりました。ENSA2 の詳細は、SAP OSS Note [2630416 - Support for Standalone Enqueue Server 2](#) を参照してください。

さらに、このドキュメントでは、SAP HANA System Replication を使用して完全に自動化されたフェイルオーバーを備えた SAP HANA Scale-Up インスタンスについても取り上げます。このインスタンスでは、設定された制約に従って、SAP HANA の昇格可能なクローンリソースが各ノードで実行されます。この記事では、SAP HANA をインストールするための RHEL システムの準備や、SAP HANA のインストール手順については説明しません。SAP S/4HANA および SAP HANA 用のシステムを迅速かつエラーなく準備するには、[RHEL System Roles for SAP](#) を使用することを推奨します。

両方の設定は、自動化された SAP HANA スケールアップシステムレプリケーション環境を備えた、コストが最適化された SAP S/4HANA と見なされます。

## 1.4. サポートポリシー

詳細は、[RHEL High Availability クラスターのサポートポリシー - SAP S/4HANA の管理](#) および [RHEL High Availability 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。

このソリューションは、上記のポリシーを満たすことを条件としてサポートされます。

## 第2章 要件

### 2.1. サブスクリプション

サブスクリプション、カーネル、およびパッチレベルをすべてのクラスターノードで同一に保つことが重要です。

この HA ソリューションを使用するには、[RHEL for SAP Solutions](#) (パブリッククラウド環境でのオンプレミスまたは BYOS セットアップ用) または [RHEL for SAP with High Availability and Update Services](#) (パブリッククラウド環境で PAYG を使用する場合) サブスクリプションが必要です。すべてのクラスターノードに対して。また、各クラスターノードで SAP NetWeaver、SAP Solutions、および High Availability リポジトリを有効にする必要があります。

この [kbase 記事](#) に従って、この環境に必要な両方のノードでリポジトリを有効にします。

### 2.2. PACEMAKER リソースエージェント

Pacemaker ベースの HA クラスターが SAP HANA システムレプリケーションと ENSA2 の両方を管理するには、次のリソースエージェントが必要です。

#### 2.2.1. SAP インスタンス

この例では、**SAPInstance** リソースエージェントを使用して、ASCS および ERS リソースを管理します。**SAPInstance** リソースエージェントのすべての操作は、SAP スタートアップサービスフレームワーク **sapstartsrv** を使用して行われます。

#### 2.2.2. SAPHanaTopology (複製されたリソース)

このリソースエージェントは、各クラスターノードの SAP HANA システムレプリケーションの状態と設定を収集しています。SAPHana リソースエージェントが正しく機能するためには、このエージェントからのデータがクラスターノード属性に存在することが不可欠です。

#### 2.2.3. SAPHana (昇格可能な複製リソース)

このリソースは、SAP HANA データベースの開始、停止、および再配置 (フェイルオーバー) を担当します。このリソースエージェントは、SAPHanaTopology によって収集された情報を取得し、それに基づいて SAP HANA データベースと対話して処理を行います。また、クラスターノードの SAP HANA ステータスに関する追加情報をクラスターノード属性に追加します。

#### 2.2.4. Filesystem

ファイルシステム用の Pacemaker クラスターリソースエージェント。NFS や iSCSI などエクスポートされた共有記憶媒体上のファイルシステムを管理します。

#### 2.2.5. IPAddr2 (または CCSP で VIP を管理するための他の RA)

仮想 IPv4 および IPv6 アドレスとエイリアスを管理します。

### 2.3.2 ノードクラスター環境

これはコスト最適化シナリオであるため、2 ノードクラスター環境のみに焦点を当てます。ENSA1 は、ERS が実行されている他のノードに ASCS がフェイルオーバーできる 2 ノードクラスターでのみ設定

できます。一方、ENSA2 は、クラスター内で 2 つを超えるノードの実行をサポートします。ただし、SAP HANA スケールアップインスタンスは 2 ノードクラスターのみ制限されているため、このコスト最適化ドキュメントでは、クラスターで 2 ノードのみを使用することですべてをシンプルにしています。

## 2.4. ストレージ要件

S/4HANA のインストール用に作成されたディレクトリーは、以下のルールに従って共有ストレージに配置する必要があります。

### 2.4.1. インスタンス固有のディレクトリー

各ノードのクラスターによってマウントできる ASCS および ERS インスタンス用に、個別の SAN LUN または NFS エクスポートが必要です。

たとえば、以下に示すように、ASCS インスタンスと ERS インスタンスはそれぞれ、インスタンス固有のディレクトリーが対応するノードに存在する必要があります。

- ASCS node: `/usr/sap/SID/ASCS<Ins#>`
- ERS ノード: `/usr/sap/SID/ERS<Ins#>`
- 両方のノード: `/hana/`
  - **注記:** システムレプリケーションがあるため、`/hana/`ディレクトリーはローカルであり、各ノードで共有されません。

**注記:** アプリケーションサーバーの場合、アプリケーションサーバーインスタンスが実行されるノードで次のディレクトリーを使用できるようにする必要があります。

- App Server Node(s) (D<Ins#>): `/usr/sap/SID/D<Ins#>`

インスタンスディレクトリーに SAN LUN を使用する場合、お客様は [HA-LVM](#) を使用して、インスタンスディレクトリーを一度に 1 つのノードにのみマウントできるようにする必要があります。

NFS エクスポートを使用する場合、Azure NetApp Files や Amazon EFS などの NFS ファイルサーバー上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。このオプションにより、クラスターは、エクスポートが作成されたディレクトリーツリーで実行されているすべての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されているプロセスのみを停止するようになります。

### 2.4.2. 共有ディレクトリー

次のマウントポイントは、ASCS、ERS、HANA、およびアプリケーションサーバーノードで使用する必要があります。

```
/sapmnt
/usr/sap/trans
/usr/sap/SID/SYS
```

共有ストレージは、次の方法で実現できます。

- 外部 NFS サーバーを使用する (NFS サーバーは、共有がマウントされるクラスター内のどのノードでも実行できません。この制限の詳細は、[Hangs occur if a Red Hat Enterprise Linux system is used as both NFS server and NFS client for the same mount](#) というソリューションを

参照してください)。

- GFS2 ファイルシステムを使用する (これには、すべてのノードに **Resilient Storage Add-on** が必要です)
- glusterfs ファイルシステムを使用する ([Can glusterfs be used for the SAP NetWeaver shared filesystems?](#) のソリューションの追加の注意事項を確認してください)

これらのマウントポイントは、クラスターによって管理されるか、クラスターが開始される前に **マウントされている** 必要があります。

## 第3章 SAP S/4HANA をインストールする

### 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。

両方のノードは、クラスター内の自動システムレプリケーションを使用して、ASCS/ERS および HDB インスタンスを実行します。

```
1st node hostname: s4node1
2nd node hostname: s4node2

SID: S4H

ASCS Instance number: 20
ASCS virtual hostname: s4ascs

ERS Instance number: 29
ERS virtual hostname: s4ers
```

HANA データベース:

```
SID: S4D
HANA Instance number: 00
HANA virtual hostname: s4db
```

### 3.2. ホストの準備

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 8 for SAP Solutions をインストールします (SAP HANA の最新の認定バージョンを推奨します)。
- システムを Red Hat Customer Portal またはサテライトに登録する
- RHEL for SAP Applications および RHEL for SAP Solutions リポジトリを有効にする
- 高可用性アドオンチャンネルを有効にする
- 共有ストレージとファイルシステムを正しいマウントポイントに配置する
- インスタンスが使用する仮想 IP アドレスを存在させ、到達可能にする
- インスタンスで使用されるホスト名を IP アドレスに解決して戻す
- インストールメディアを利用可能にする
- SAP S/4HANA を実行するための推奨事項に従ってシステムを設定する
  - 詳細については、Red Hat Enterprise Linux 8.x: インストールと設定 - [SAP OSS ノート 2772999](#) を参照してください。

### 3.3. S/4HANA をインストールする

SAP の 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。

- ASCS インスタンス
- ERS インスタンス
- システムレプリケーションを使用する両方のノードの SAP HANA DB インスタンス

### 3.3.1. S/4 を s4node1 にインストールします

次のファイルシステムは、ASCS がインストールされる s4node1 にマウントする必要があります。

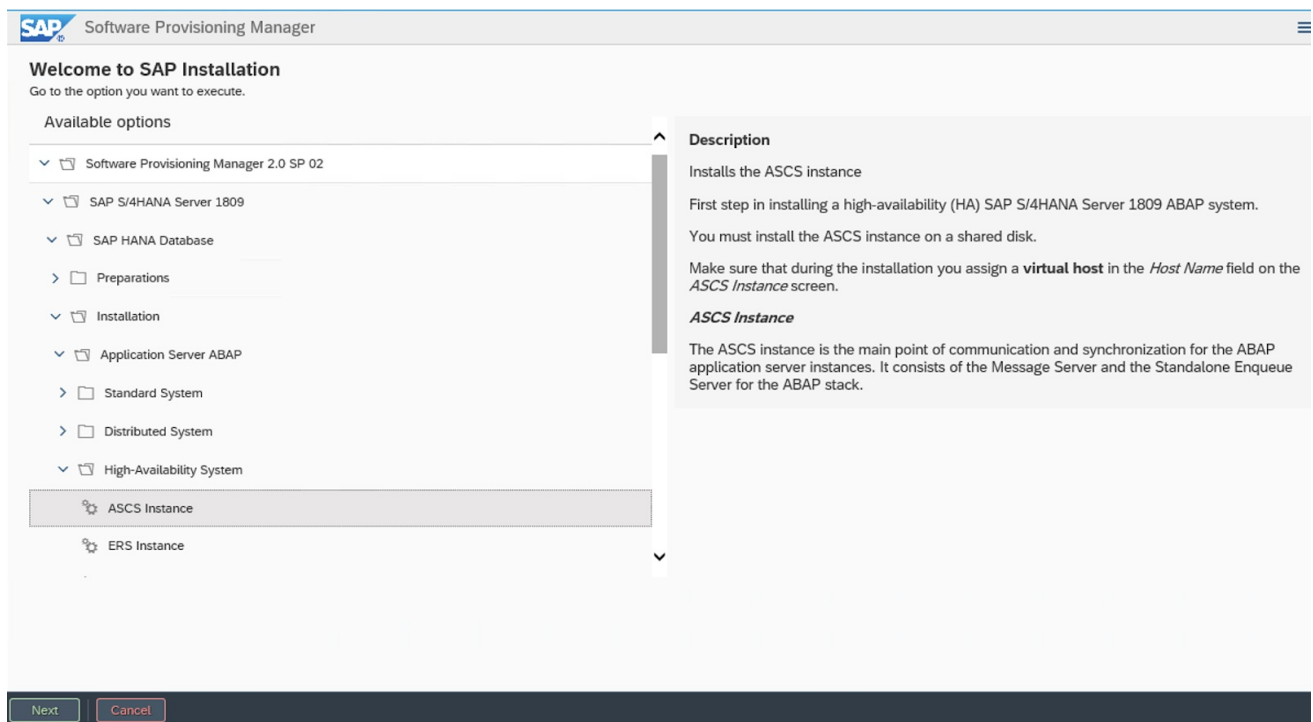
```
/usr/sap/S4H/ASCS20
/usr/sap/S4H/SYS
/usr/sap/trans
/sapmnt
```

s4ascs の仮想 IP は、**s4node1** で有効にする必要があります。

インストーラーを実行します。

```
[root@s4node1]# ./sapinst SAPINST_USE_HOSTNAME=s4ascs
```

**High-Availability System** オプションを選択します。



### 3.3.2. s4node2 に ERS をインストールする

次のファイルシステムは、ERS がインストールされる s4node2 にマウントする必要があります。

■

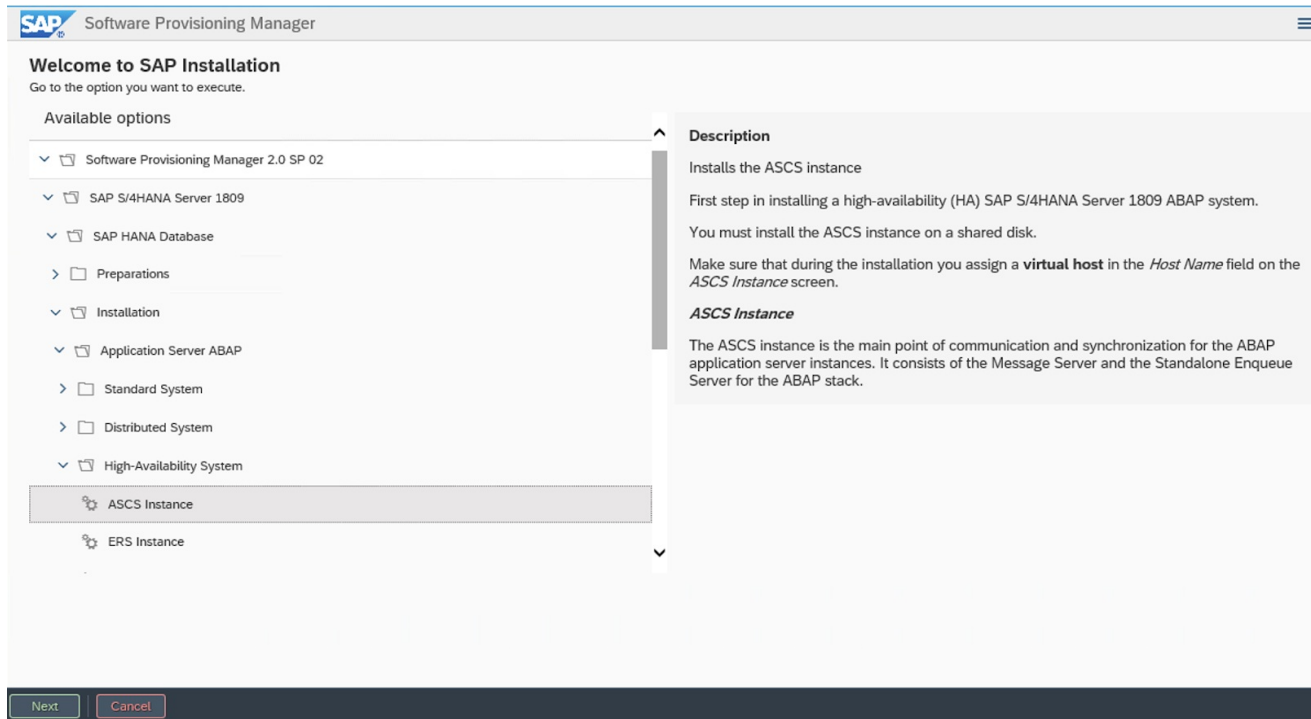
```
/usr/sap/S4H/ERS29 /usr/sap/S4H/SYS /usr/sap/trans /sapmnt
```

s4node2 で **s4ers** の仮想 IP を有効にする必要があります。

インストーラーを実行します。

```
[root@s4node2]# ./sapinst SAPINST_USE_HOSTNAME=s4ers
```

**High-Availability System** オプションを選択します。



### 3.3.3. SAP HANA

この例では、次の設定で SAP HANA を使用します。 [support policies](#) に従って、他のサポートされているデータベースを使用することもできます。

```
SAP HANA SID: S4D
SAP HANA Instance number: 00
```

この例では、hdblcm コマンドラインツールを使用して SAP HANA データベースサーバーを両方のノードにインストールできます。その後、次のアークティクルで説明されているのと同じ方法で、自動化された HANA システムレプリケーションが確立されます。 [SAP HANA system replication in pacemaker cluster](#)。

このセットアップでは、ASCS と HANA のプライマリーインスタンスが互いに独立してフェイルオーバーできることに注意してください。したがって、ASCS とプライマリー SAP HANA インスタンスが同じノードで実行を開始する状況が発生します。したがって、ASCS/ERS とプライマリー SAP HANA インスタンスが他のノードで障害が発生した場合に一方のノードで実行できるように、両方のノードに十分なリソースと空きメモリーがあることを確認することも重要です。



これを実現するために、SAP HANA インスタンスに特定のメモリー制限/制限を設定できます。SAP HANA 環境ごとに利用可能なオプションを調べるために、SAP に連絡することを強く推奨します。関連するリンクは次のとおりです。

- [Change the Global Memory Allocation Limit](#)
- [HANA: How to set allocation limit for tenant databases](#)

## 3.4. インストール後の設定

### 3.4.1. ASCS プロファイルの変更

ASCS インスタンスは、クラスターによって管理されるため、サーバーインスタンスの自動再起動を防ぐために、そのプロファイルに次の変更を加える必要があります。変更を適用するには、ASCS プロファイル `/sapmnt/S4H/profile/S4H_ASCS20_s4ascs` で次のコマンドを実行します。

```
[root]# 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、エンキューサーバーの自動再起動を防ぐために、そのプロファイルに次の変更を加える必要があります。変更を適用するには、ERS プロファイル `/sapmnt/S4H/profile/S4H_ERS29_s4ers` で次のコマンドを実行します。

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

### 3.4.3. `/usr/sap/sapservices` ファイルを更新します

s4node1 と s4node2 の両方で、次の 2 行が `/usr/sap/sapservices` ファイルでコメント化されていることを確認します。

```
#LD_LIBRARY_PATH=/usr/sap/S4H/ERS29/exe:$LD_LIBRARY_PATH; export  
LD_LIBRARY_PATH; /usr/sap/S4H/ERS29/exe/sapstartsrv  
pf=/usr/sap/S4H/SYS/profile/S4H_ERS29_s4ers -D -u s4hadm  
#LD_LIBRARY_PATH=/usr/sap/S4H/ASCS20/exe:$LD_LIBRARY_PATH; export  
LD_LIBRARY_PATH; /usr/sap/S4H/ASCS20/exe/sapstartsrv  
pf=/usr/sap/S4H/SYS/profile/S4H_ASCS20_s4ascs -D -u s4hadm
```

### 3.4.4. フェイルオーバーノードで ASCS と ERS のマウントポイントを作成する

```
[root@s4node1 ~]# mkdir /usr/sap/S4H/ERS29/  
[root@s4node1 ~]# chown s4hadm:sapsys /usr/sap/S4H/ERS29/  
  
[root@s4node2 ~]# mkdir /usr/sap/S4H/ASCS20  
[root@s4node2 ~]# chown s4hadm:sapsys /usr/sap/S4H/ASCS20
```

### 3.4.5. 他のノードでインスタンスを手動でテストする

ASCS および ERS インスタンスを停止します。インスタンス固有のディレクトリーを他のノードに移動します。

```
[root@s4node1 ~]# umount /usr/sap/S4H/ASCS20
[root@s4node2 ~]# mount /usr/sap/S4H/ASCS20

[root@s4node2 ~]# umount /usr/sap/S4H/ERS29/
[root@s4node1 ~]# mount /usr/sap/S4H/ERS29/
```

他のクラスターノードで ASCS インスタンスと ERS インスタンスを手動で開始し、それぞれ手動で停止します。

### 3.4.6. すべてのノードで **SAP HostAgent** を確認する

すべてのノードで、SAP HostAgent のバージョンが同じで、最小バージョン要件を満たしているかどうかを確認します。

```
[root]# /usr/sap/hostctrl/exe/saphostexec -version
```

SAP HostAgent をアップグレード/インストールします。[SAP OSS Note 1031096](#) に従ってください。

### 3.4.7. 恒久的な **SAP** ライセンスキーをインストールする

高可用性シナリオでの SAP ハードウェアキーの決定が改善されました。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必要がある場合があります。詳細については、SAP OSS Note [1178686](#) - Linux: Alternative method to generate a SAP hardware key を参照してください。

## 第4章 PACEMAKER をインストールする

次のドキュメントを参照して、最初に Pacemaker クラスターをセットアップしてください。

- [Red Hat Enterprise Linux 7 の高可用性アドオンの参照ドキュメント](#)
- [RHEL 8 での高可用性クラスターの設定と管理](#)

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。

このガイドでは、次のことが適切に機能していることを前提としています。

- Pacemaker クラスターはドキュメントに従って設定されており、適切で機能するフェンシングがあります
- (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は、[エンキューレプリケーションサーバーのフェイルオーバーの設定](#) で説明されているように、手動でテストされています。
- ノードは、[RHEL for SAP Repositories and How to Enable Them](#) で説明されているように、必要なチャンネルにサブスクライブされます。

### 4.1. 一般的なクラスタープロパティを設定する

初期テスト中および運用後のリソースの不要なフェイルオーバーを回避するには、`resource-stickiness` および `migration-threshold` パラメーターに次のデフォルト値を設定します。デフォルトは、独自に定義された値で上書きするリソースには適用されないことに注意してください。

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

**警告:** RHEL 8.4 (pcs-0.10.8-1.el8) の時点で、上記のコマンドは非推奨です。以下のコマンドを使用してください: `+[source,text]`

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

#### 注記:

1.上記のコマンドをクラスターの1つのノードで実行するだけで十分です。

2.コマンド **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。

### 4.2. すべてのクラスターノードに RESOURCE-AGENTS-SAP をインストールします

```
[root]# yum install resource-agents-sap
```

### 4.3. 共有ファイルシステムのクラスターリソースを設定する

すべてのクラスターノードで次のマウントポイントを提供するように共有ファイルシステムを設定します。

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

#### 4.3.1. クラスターによって管理される共有ファイルシステムを設定する

以下に示すように、複製された **Filesystem** クラスターリソースを使用して、外部 NFS サーバーからの共有をすべてのクラスターノードにマウントできます。

```
[root]# pcs resource create s4h_fs_sapmnt Filesystem \
device='<NFS_Server>:<sapmnt_nfs_share>' directory='/sapmnt' \
fstype='nfs' --clone interleave=true
[root]# pcs resource create s4h_fs_sap_trans Filesystem \
device='<NFS_Server>:<sap_trans_nfs_share>' directory='/usr/sap/trans' \
fstype='nfs' --clone interleave=true
[root]# pcs resource create s4h_fs_sap_sys Filesystem \
device='<NFS_Server>:<s4h_sys_nfs_share>' directory='/usr/sap/S4H/SYS' \
fstype='nfs' --clone interleave=true
```

**Filesystem** リソースを作成した後、それらがすべてのノードで適切に開始されていることを確認します。

```
[root]# pcs status
...
Clone Set: s4h_fs_sapmnt-clone [s4h_fs_sapmnt]
Started: [ s4node1 s4node2 ]
Clone Set: s4h_fs_sap_trans-clone [s4h_fs_sap_trans]
Started: [ s4node1 s4node2 ]
Clone Set: s4h_fs_sys-clone [s4h_fs_sys]
Started: [ s4node1 s4node2 ]
...
```

#### 4.3.2. クラスター外で管理される共有ファイルシステムを設定する

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** サービスを開始する前にそれらが利用可能であることを確認する必要があります。

RHEL 7 では、**systemd** の並列化により、共有ファイルシステムが **resource-agents-deps** ターゲットで開始されていることを確認する必要があります。詳細は、ドキュメントセクションを参照してください。 [9.6.Pacemaker で管理されていないリソースの依存関係の起動順の設定 \(Red Hat Enterprise Linux 7.4 以降\)](#)

### 4.4. ASCS リソースグループの設定

#### 4.4.1. 仮想 IP アドレスのリソースを作成する

```
[root]# pcs resource create s4h_vip_ascs20 IPAddr2 ip=192.168.200.201 \
--group s4h_ASCS20_group
```

#### 4.4.2. ASCS ファイルシステムのリソースを作成します。

以下は、NFS ファイルシステムのリソースを作成する例です。

```
[root]# pcs resource create s4h_fs_asc20 Filesystem \
device='<NFS_Server>:<s4h_asc20_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
```

以下は、HA-LVM ファイルシステムのリソースを作成する例です。

```
[root]# pcs resource create s4h_fs_asc20_lvm LVM \
volgrpname='<asc20_volume_group>' exclusive=true \
--group s4h_ASCS20_group

[root]# pcs resource create s4h_fs_asc20 Filesystem \
device='/dev/mapper/<asc20_logical_volume>' \
directory=/usr/sap/S4H/ASCS20 fstype=ext4 \
--group s4h_ASCS20_group
```

#### 4.4.3. ASCS インスタンスのリソースを作成する

```
[root]# pcs resource create s4h_asc20 SAPInstance \
InstanceName="S4H_ASCS20_s4asc20" \
START_PROFILE=/sapmnt/S4H/profile/S4H_ASCS20_s4asc20 \
AUTOMATIC_RECOVER=false \
meta resource-stickiness=5000 \
--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
```

**注記:** `meta resource-stickiness=5000` は、フェイルオーバーの制約と ERS のバランスを取るためにここにあるため、リソースは開始したノードに留まり、制御不能にクラスター内を移動することはありません。

グループにリソーススティッキネスを追加して、可能であれば ASCS がノードに留まるようにします。

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

### 4.5. ERS リソースグループの設定

#### 4.5.1. 仮想 IP アドレスのリソースを作成する

```
[root]# pcs resource create s4h_vip_ers29 IPAddr2 ip=192.168.200.202 \
--group s4h_ERS29_group
```

#### 4.5.2. ERS ファイルシステムのリソースを作成する

以下は、NFS ファイルシステムのリソースを作成する例です。

```
[root]# 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
```

以下は、HA-LVM ファイルシステムのリソースを作成する例です。

```
[root]# pcs resource create s4h_fs_ers29_lvm LVM \
volgrpname='<ers_volume_group>' exclusive=true --group s4h_ERS29_group

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

### 4.5.3. ERS インスタンスのリソースを作成する

ERS インスタンスのクラスターリソースを作成します。

注記: ENSA2 デプロイメントでは、**IS\_ERS** 属性は任意です。**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。

```
[root]# pcs resource create s4h_ers29 SAPInstance \
InstanceName="S4H_ERS29_s4ers" \
START_PROFILE=/sapmnt/S4H/profile/S4H_ERS29_s4ers \
AUTOMATIC_RECOVER=false \
--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
```

## 4.6. 制約を作成する

### 4.6.1. ASCS 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]# pcs constraint colocation add s4h_ERS29_group with s4h_ASCS20_group \
-5000
```

### 4.6.2. ASCS リソースの場所の制約を作成する

ASCS20 インスタンス **rh2\_ascs20** は、フェイルオーバー前に ERS が実行されていたノードでの実行を優先します。

```
# pcs constraint location rh2_ascs20 rule score=2000 runs_ers_RH2 eq 1
```

### 4.6.3. ASCS および ERS リソースグループの順序制約を作成する

**s4h\_ERS29\_group** の前に **s4h\_ASCS20\_group** を開始することを推奨します

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

#### 4.6.4. クラスターが管理する /sapmnt リソースの順序制約を作成する

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、次の制約により、ASCS および ERS の SAPInstance リソースを含むリソースグループは、ファイルシステムが使用可能になったときにのみ開始されます。

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

## 第5章 クラスター設定をテストする

### 5.1. 制約を確認してください

```
[root]# pcs constraint
Location Constraints:
Ordering Constraints:
start s4h_ASCS20_group then start s4h_ERS29_group (kind:Optional) (non-symmetrical)
start s4h_ASCS20_group then stop s4h_ERS29_group (kind:Optional) (non-symmetrical)
Colocation Constraints:
s4h_ERS29_group with s4h_ASCS20_group (score:-5000) Ticket Constraints:
```

### 5.2. ノードクラッシュによるフェイルオーバー ASCS

クラッシュの前に、ASCS は s4node1 で実行され、ERS は s4node2 で実行されていました。

```
[root@s4node1]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node1
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2 ...
```

s4node2 で、次のコマンドを実行して、クラスター内のステータスの変化を監視します。

```
[root@s4node2 ~]# crm_mon -Arf
```

次のコマンドを実行して、s4node1 をクラッシュさせます。コマンド実行後、s4node1 への接続が失われることに注意してください。

```
[root@s4node1 ~]# echo c > /proc/sysrq-trigger
```

s4node2 で、フェイルオーバープロセスをモニターします。フェイルオーバー後、クラスターはこのような状態になり、ASCS と ERS の両方が s4node2 にあるはずで

```
[root@s4node2 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node2
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
...
```



### 5.3. ERS は以前に障害が発生したノードに移動します

s4node1 をオンラインに戻し、クラスターを開始します。

```
[root@s4node1 ~]# pcs cluster start
```

ERS は s4node1 に移動する必要がありますが、ASCS は s4node2 に残ります。ERS が移行を完了するまで待ちます。最後に、クラスターは次のような状態になります。

```
[root@node1 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started
s4node2 s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node2
s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node2
Resource Group: s4h_ERS29_group
s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started
s4node1 s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node1
s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node1
...
```

## 第6章 再起動後にクラスターを自動起動できるようにする

クラスターは、再起動後に自動起動するようにまだ有効化されていません。システム管理者は、ノードがフェンシングされて再起動された後、クラスターを手動で開始する必要があります。

前のセクションをテストした後、すべてが正常に機能したら、再起動後にクラスターを自動起動できるようにします。

```
[root@s4node1]# 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。例えば、クラスターリソースが必要とするファイルシステムに問題があり、ファイルシステムを修復しないと再利用できない場合、ファイルシステムが動作しないため、クラスターを自動起動させることは失敗となる可能性があります。これにより、さらに問題が発生する可能性があります。

ここで、前のセクションのテストを再実行して、クラスターがまだ正常に機能することを確認してください。セクション 5.3 では、ノードの再起動後にコマンド **pcs cluster start** を実行する必要がないことに注意してください。クラスターは、再起動後に自動的に開始する必要があります。

この時点で、ENSA2 の 2 ノードクラスターが正常に設定されました。集中的なテストを続行して本番環境の準備を整えるか、オプションでクラスターにノードを追加することができます。

## 第7章 フェイルオーバーのテスト

### 7.1. ノードクラッシュによるフェイルオーバー ASCS

クラッシュの前に、ASCS は s4node1 で実行され、ERS は s4node2 で実行されていました。s4node2 で、次のコマンドを実行して、クラスター内のステータスの変化を監視します。

```
[root@s4node2 ~]# crm_mon -Arf
```

次のコマンドを実行して、s4node1 をクラッシュさせます。コマンド実行後、s4node1 への接続が失われることに注意してください。

```
[root@s4node1 ~]# echo c > /proc/sysrq-trigger
```

s4node2 で、フェイルオーバープロセスをモニターします。フェイルオーバー後、クラスターはこのような状態になり、ASCS は s4node3 で実行され、ERS は s4node2 に残ります。

```
[root@s4node2 ~]# pcs status
...
Resource Group: s4h_ASCS20_group
  s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
  s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node1
  s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
Resource Group: s4h_ERS29_group
  s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
  s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
  s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
...
```

### 7.2. ERS は現在のノードのまま

s4node1 をオンラインに戻します。ERS は、s4node1 に戻るのではなく、現在のノードに留まる必要があります。

### 7.3. ERS クラッシュのテスト

同様に、ERS が実行されているノードをテストしてクラッシュさせます。ASCS が現在のノードにそのまま残っている間、ERS グループはスペアノードにフェイルオーバーする必要があります。クラッシュしたノードが戻った後、ERS グループは戻ってはいけません。

### 7.4. コストが最適化された SAP S/4HANA クラスター環境は、次のようになります。

```
[root@s4node1 ~]# pcs status
Cluster name: SAP-S4-HANA
....
Node List:
  * Online: [ s4node1 s4node2 ]
....
Full List of Resources:
  * s4-fence (stonith:fence_rhevm): Started s4node1
```

```
* Clone Set: fs_sapmnt-clone [fs_sapmnt]:
* Started: [ s4node1 s4node2 ]
* Clone Set: fs_sap_trans-clone [fs_sap_trans]:
* Started: [ s4node1 s4node2 ]
* Clone Set: fs_sap_SYS-clone [fs_sap_SYS]:
* Started: [ s4node1 s4node2 ]
* Resource Group: s4h_ASCS20_group:
* s4h_lvm_ascs20 (ocf::heartbeat:LVM-activate): Started s4node1
* s4h_fs_ascs20 (ocf::heartbeat:Filesystem): Started s4node1
* s4h_ascs20 (ocf::heartbeat:SAPInstance): Started s4node1
* s4h_vip_ascs20 (ocf::heartbeat:IPAddr2): Started s4node1
* Resource Group: s4h_ERS29_group:
* s4h_lvm_ers29 (ocf::heartbeat:LVM-activate): Started s4node2
* s4h_fs_ers29 (ocf::heartbeat:Filesystem): Started s4node2
* s4h_ers29 (ocf::heartbeat:SAPInstance): Started s4node2
* s4h_vip_ers29 (ocf::heartbeat:IPAddr2): Started s4node2
* Clone Set: SAPHanaTopology_S4D_00-clone [SAPHanaTopology_S4D_00]:
* Started: [ s4node1 s4node2 ]
* Clone Set: SAPHana_S4D_00-clone [SAPHana_S4D_00] (promotable):
  * Masters: [ s4node2 ]
* Slaves: [ s4node1 ]
* vip_S4D_00 (ocf::heartbeat:IPAddr2): Started s4node2
```

## 第8章 オプション - SAP 管理ツールを使用して、クラスター制御の ASCS/ERS インスタンスを管理するための SAP HA インターフェイスを有効にします。

システム管理者が手動で、または SAP 管理コンソール (MC/MMC) などのツールを使用して、Pacemaker クラスター内で実行されている SAP インスタンスを制御する場合、HA クラスターソフトウェアによって提供される HA インターフェイスを介して変更を行う必要があります。SAP Start Service **sapstartsrv** は SAP インスタンスを制御し、HA インターフェイスを介して Pacemaker クラスターソフトウェアと通信するように設定する必要があります。

HALib を設定するには、**kbase** の記事に従ってください: [How to enable the SAP HA Interface for SAP ABAP application server instances managed by the RHEL HA Add-On?](#)