

SUSE Linux Enterprise High Availability Extension

11 SP1

www.novell.com

2010 年4 月22 日

High Availability ガイド



High Availability ガイド

Copyright © 2006- 2010 Novell, Inc.

この文書は、フリーソフトウェア財団発行のGNUフリー文書ライセンス バージョン 1.2 またはそれ以降に定める条項に従って、複製、頒布、あるいは改変が許可されています。ただし、変更不可部分であるこの著作権表示およびライセンスを変更せずに記載すること。「GNUフリー文書ライセンス」と記載されたセクションにライセンスのコピーが含まれています。

SUSE®, openSUSE®, openSUSE®のロゴ、Novell®, Novell®のロゴ、N®のロゴは、米国およびその他の国におけるNovell, Inc.の登録商標です。*LinuxはLinus Torvalds氏の登録商標です。他のすべての第三者の商標は、各所有者が所有権を有しています。商標記号(®、™など)は、Novellの商標を示します。アスタリスク(*)は、サードパーティの商標を示します。

本書のすべての情報は、細心の注意を払って編集されています。しかし、このことは絶対に正確であることを保証するものではありません。Novell, Inc.、Suse Linux Products GmbH、著者、翻訳者のいずれも誤りまたはその結果に対して一切責任を負いかねます。

目次

このガイドについて	ix
パート I インストールおよび管理	1
1 製品の概要	3
1.1 主な機能	4
1.2 利点	7
1.3 クラスタ構成: ストレージ	10
1.4 アーキテクチャ	12
2 はじめに	17
2.1 ハードウェア要件	17
2.2 ソフトウェアの必要条件	18
2.3 共有ディスクのシステム要件	18
2.4 準備作業	19
2.5 概要: クラスタのインストールとセットアップ	19
3 YaSTによるインストールと基本設定	21
3.1 High Availability Extensionのインストール	21
3.2 クラスタの初期セットアップ	22
3.3 クラスタをオンラインにする	30
3.4 AutoYaSTによる大量展開	31

パート II 設定および管理	35
4 設定および管理の基本事項	37
4.1 グローバルクラスタオプション	37
4.2 クラスタリソース	39
4.3 リソース監視	51
4.4 リソースの制約	52
4.5 詳細情報	58
5 クラスタリソースの設定と管理(GUI)	61
5.1 Pacemaker GUI - 概要	62
5.2 グローバルクラスタオプションの設定	65
5.3 クラスタリソースの設定	66
5.4 クラスタリソースの管理	88
6 クラスタリソースの設定と管理(コマンドライン)	95
6.1 crmコマンドラインツール - 概要	95
6.2 グローバルクラスタオプションの設定	102
6.3 クラスタリソースの設定	103
6.4 クラスタリソースの管理	116
7 Webインターフェイスによるクラスタリソースの管理	119
7.1 HA Web Konsoleの起動とログイン	120
7.2 HA Web Konsoleの使用	121
7.3 トラブルシューティング	122
8 リソースエージェントの追加または変更	125
8.1 STONITHエージェント	125
8.2 OCFリソースエージェントの作成	126
8.3 OCF戻りコードと障害回復	127
9 フェンシングとSTONITH	131
9.1 フェンシングのクラス	131
9.2 ノードレベルのフェンシング	132
9.3 STONITHの構成	134
9.4 フェンシングデバイスの監視	139
9.5 特殊なフェンシングデバイス	140
9.6 詳細情報	142

10	Linux Virtual Serverによる負荷分散	143
10.1	概念の概要	143
10.2	YaSTによるIP負荷分散の設定	146
10.3	追加設定	152
10.4	詳細情報	153
11	ネットワークデバイスボンディング	155
11.1	YaSTによるボンディングデバイスの設定	155
11.2	詳細情報	157
	パート III ストレージおよびデータレプリケーション	159
12	Oracle Cluster File System 2	161
12.1	特長と利点	161
12.2	OCFS2のパッケージと管理ユーティリティ	162
12.3	OCFS2サービスの設定	163
12.4	OCFS2ボリュームの作成	166
12.5	OCFS2ボリュームのマウント	169
12.6	詳細情報	170
13	Distributed Replicated Block Device (DRBD)	171
13.1	概念の概要	171
13.2	DRBDサービスのインストール	173
13.3	DRBDサービスの設定	174
13.4	DRBDサービスのテスト	178
13.5	DRBDのチューニング	180
13.6	DRBDのトラブルシュート	180
13.7	詳細情報	182
14	クラスタLVM	183
14.1	概念の概要	183
14.2	clVMの環境設定	184
14.3	有効なLVM2デバイスの明示的な設定	192
14.4	詳細情報	193
15	ストレージ保護	195
15.1	ストレージベースのフェンシング	196
15.2	排他的ストレージアクティベーションの確保	201

16 Sambaクラスタリング	205
16.1 概念の概要	205
16.2 基本的な設定	207
16.3 クラスタ対応Sambaのデバッグとテスト	209
16.4 詳細情報	211
 パート IV トラブルシューティングと参照情報	 213
 17 トラブルシューティング	 215
17.1 インストールの問題	215
17.2 HAクラスタの「デバッグ」	216
17.3 FAQ	218
17.4 その他の情報	219
 18 クラスタ管理ツール	 221
 19 HA OCF Agents	 275
 パート V 付録	 367
 A 単純なテストリソースのセットアップ例	 369
A.1 GUIによるリソースの構成	369
A.2 リソースの手動設定	371
 B クラスタの最新製品バージョンへのアップグレード	 373
B.1 SLES 10からSLEHA 11へのアップグレード	373
B.2 SLEHA 11からSLEHA 11 SP1へのアップグレード	378
 C 新機能	 381
C.1 バージョン10 SP3からバージョン11への変更点	381
C.2 バージョン11からバージョン11 SP1への変更点	385
 D GNU利用許諾契約書	 389
D.1 GNU General Public License	389
D.2 GNU Free Documentation License	392

このガイドについて

SUSE® Linux Enterprise High Availability Extensionはオープンソースクラスタリングテクノロジーの統合スイートで、高可用性を備えた物理および仮想Linuxクラスタを実装できます。構成と管理をすばやく効率的に行うため、High Availability Extensionにはグラフィカルユーザインタフェース(GUI)とコマンドラインインタフェース(CLI)の両方が備わっています。さらに、HA Web Konsoleも標準装備しているので、WebインターフェイスからでもLinuxクラスタを管理できます。

このガイドは、High Availability(HA)クラスタのセットアップ、構成、保守を行う必要がある管理者向けに作成されています。両方のアプローチ(GUIとCLI)について詳細に記述し、管理者が主要タスクの実行に必要な、適切なツールを選択できるよう支援します。

このガイドは、次のパートに分かれています。

インストールおよび管理

このパートでは、クラスタのインストールと設定を開始する前に、クラスタの基本とアーキテクチャをよく把握し、主要な機能と利点の概要を理解します。必要なハードウェア/ソフトウェア要件と、以降の手順を実行する前に必要な準備作業について学習します。YaSTを使用してHAクラスタのインストールおよび基本セットアップを実行します。

設定および管理

このパートでは、グラフィックユーザインタフェース(Pacemaker GUI)またはcrmコマンドラインインタフェースを使用して、リソースを追加、設定、管理します。Webインターフェイスを介してクラスタを監視する場合は、HA Web Konsoleを使用します。負荷分散およびフェンシングの使用方法を学習します。独自のリソースエージェントの作成、または既存のエージェントの変更を検討している場合、別の種類のリソースエージェントを作成する方法について背景情報を取得できます。

ストレージおよびデータレプリケーション

SUSE Linux Enterprise High Availability Extensionには、クラスタ対応型ファイルシステムのOCF2 (Oracle Cluster File System)とボリュームマネージャであるcLVM(clustered Logical Volume Manager)が標準装備されています。データのレプリケーションでは、DRBD (Distributed Replicated Block Device)を使用して、High Availabilityサービスのデータをクラスタのアクティブ

ノードからスタンバイノードへミラーリングします。さらに、クラスタ化したSambaサーバにより、異種混合環境にもHigh Availabilityソリューションが提供されます。

トラブルシューティングと参照情報

独自のクラスタの管理には、一定量のトラブルシューティングを実行する必要があります。このパートでは、よくある問題とその解決方法を学習します。High Availability Extensionから提供されているコマンドラインツールの総合的なリファレンスを参照し、クラスタ管理に役立ててください。

付録

このパートには、最新リリース以降のHigh Availability Extensionの新機能と動作変更が一覧されています。クラスタを最新リリースバージョンに移行する方法を学び、単純なテストリソースの設定例を参照してください。

このマニュアル中の多くの章に、他の資料やリソースへのリンクが記載されています。それらは、システム上で参照できる追加ドキュメントやインターネットから入手できるドキュメントなどです。

ご使用製品の利用可能なマニュアルと最新のドキュメントアップデートの概要については、<http://www.novell.com/documentation>を参照してください。

1 フィードバック

次のフィードバックチャンネルがあります:

バグと機能拡張の要求

ご使用の製品に利用できるサービスとサポートのオプションについては、<http://www.novell.com/services/>を参照してください。

製品コンポーネントのバグの報告には、<http://support.novell.com/additional/bugreport.html>をご利用ください。

機能拡張の要求は、<https://secure-www.novell.com/rms/rmsTool?action=ReqActions.viewAddPage&return=www>から送信してください。

ユーザからのコメント

私たちは、このマニュアルおよびこの製品に含まれている他のドキュメントについて、皆さんのコメントや提案をうかがいたいと思っています。オンラインドキュメントの各ページの下にあるユーザコメント機能を使用するか、または<http://www.novell.com/documentation/feedback.html>にアクセスして、コメントを入力してください。

2 マニュアルの表記規則

本書では、次の書体を使用しています：

- `/etc/passwd`:ディレクトリ名とファイル名
- `placeholder:placeholder`は、実際の値で置き換えられます
- `PATH`:環境変数`PATH`
- `ls`、`--help`:コマンド、オプション、およびパラメータ
- `user`:ユーザまたはグループ
- `<Alt>`、`Alt+F1`:キー:押すためのキーまたはキーの組み合わせ、キーはキーボードと同様に、大文字で表示されます
- `[ファイル]`、`[ファイル] > [名前を付けて保存]`:メニュー項目、ボタン
- **► amd64 em64t**: この項は、指定されたアーキテクチャにのみ関連しています。矢印は、テキストブロックの先頭と終わりを示します。 **◄**
- *Dancing Penguins* (*Penguins*章、↑他のマニュアル):他のマニュアルの章への参照です。

パート I. インストールおよび管 理

製品の概要

SUSE® Linux Enterprise High Availability Extensionはオープンソースクラスタ化技術の統合スイートで、可用性の高い物理的および仮想Linuxクラスタを実装し、SPOF (単一障害点)をなくします。データ、アプリケーション、サービスなどの重要なネットワークリソースの高度な可用性と管理のしやすさを実現します。その結果、ミッションクリティカルなLinuxワークロードに対してビジネスの継続性維持、データ整合性の保護、予期せぬダウンタイムの削減を行います。

基本的な監視、メッセージング、およびクラスタリソース管理の機能を標準装備し、個々の管理対象クラスタリソースのフェールオーバー、フェールバック、および移行(負荷分散)をサポートします。High Availability ExtensionはSUSE Linux Enterprise Server 11 SP1へのアドオンとして提供されています。

この章では、High Availability Extensionの主な製品機能と利点を紹介します。ここには、いくつかのクラスタ例が記載されており、クラスタを構成するコンポーネントについて学ぶことができます。最後のセクションでは、アーキテクチャの概要を示し、クラスタ内の個々のアーキテクチャ層とプロセスについて説明します。

High Availabilityクラスタのコンテキストでよく使用される用語については、用語集 (397 ページ)を参照してください。

1.1 主な機能

SUSE® Linux Enterprise High Availability Extensionを使用することで、ネットワークリソースの可用性を確保し、維持することができます。以降のセクションでは、いくつかの主要機能に焦点を合わせて説明します。

1.1.1 広範なクラスタリングシナリオ

High Availability Extensionは、次のシナリオをサポートします。

- アクティブ/アクティブ構成
- アクティブ/パッシブ構成: N+1、N+M、Nから1、NからM
- ハイブリッド物理仮想クラスタ。仮想サーバを物理サーバとともにクラスタ化できます。これによって、サービスの可用性とリソースの使用状況が向上します。

クラスタには、最大16のLinuxサーバを含めることができます。クラスタ内のどのサーバも、クラスタ内の障害が発生したサーバのリソース(アプリケーション、サービス、IPアドレス、およびファイルシステム)を再起動することができます。

1.1.2 柔軟性

High Availability Extensionには、Corosync/OpenAISメッセージングおよびメンバーシップ層のほか、Pacemakerクラスタリソースマネージャが標準装備されています。Pacemakerの使用によって、管理者は継続的にリソースのヘルスとステータスを監視し、依存関係を管理し、柔軟に設定できるルールとポリシーに基づいてサービスを自動的に開始および停止できます。High Availability Extensionではユーザの組織に合わせて特定のアプリケーションおよびハードウェアインフラストラクチャに応じたクラスタのカスタマイズが可能です。時間依存設定を使用して、サービスを特定の時刻に修復済みのノードに自動的にフェールバックさせることができます。

1.1.3 ストレージとデータレプリケーション

High Availability Extensionでは必要に応じてサーバストレージを自動的に割り当て、再割り当てすることができます。ファイバチャネルまたはiSCSIストレージエリアネットワーク(SAN)をサポートしています。共有ディスクもサポートされていますが、必要要件ではありません。SUSE Linux Enterprise High Availability Extensionには、クラスタ対応のファイルシステムとボリュームマネージャ(OCFS2(Oracle Cluster File System)、cLVM (clustered Logical Volume Manager))も含まれています。データのレプリケーションでは、DRBD (Distributed Replicated Block Device)を使用して、High Availabilityサービスのデータをクラスタのアクティブノードからスタンバイノードへミラーリングできます。さらに、SUSE Linux Enterprise High Availability Extensionは、SambaクラスタリングのテクノロジーであるCTDB (Clustered Trivial Database)もサポートします。

1.1.4 仮想化環境のサポート

SUSE Linux Enterprise High Availability Extensionは物理的および仮想Linuxサーバの両方が混在したクラスタリングをサポートしています。SUSE Linux Enterprise Server 11 SP1は、オープンソース仮想化ハイパーバイザであるXenと、ハードウェア仮想化拡張機能に基づく、Linuxの仮想化ソフトウェアであるKVM (Kernel-based Virtual Machine)を標準装備しています。High Availability Extension内のクラスタリソースマネージャは、Xenで作成された仮想サーバで実行中のサービスと物理サーバで実行中のサービスを認識、監視、および管理できます。ゲストシステムは、クラスタにサービスとして管理されます。

1.1.5 リソースエージェント

SUSE Linux Enterprise High Availability Extensionには、Apache、IPv4、IPv6、その他多数のリソースを管理するための膨大な数のリソースエージェントが含まれています。またIBM WebSphere Application Serverなどの一般的なサードパーティアプリケーション用のリソースエージェントも含まれています。ご利用の製品に含まれているOpen Cluster Framework (OCF)リソースエージェントのリストは、第19章 *HA OCF Agents* (275 ページ)を参照してください。

1.1.6 ユーザフレンドリな管理ツール

High Availability Extensionは、クラスタの基本的なインストールとセットアップのほか、効果的な設定および管理に使用できる強力なツールセットを標準装備しています。

YaST

一般的なシステムインストールおよび管理用グラフィックユーザインターフェイス。これを使用して、3.1項「High Availability Extensionのインストール」(21 ページ)に説明されているように、SUSE Linux Enterprise Server上にHigh Availability Extensionをインストールします。YaSTには、クラスタまたは個々のコンポーネントの設定に役立つ次のモジュールがHigh Availabilityカテゴリに含まれてます。

- ・ クラスタ: 基本的なクラスタセットアップ。詳細については、3.2項「クラスタの初期セットアップ」(22 ページ)を参照してください。
- ・ DRBD: Distributed Replicated Block Deviceの設定。
- ・ IP負荷分散: Linux Virtual Serverによる負荷分散の設定。詳細については、第10章 *Linux Virtual Server*による負荷分散(143 ページ)を参照してください。

Pacemaker GUI

クラスタの構成と管理を容易にするインストール可能なグラフィックユーザインターフェイス。リソースの作成と設定を支援し、リソースの起動、中止、移行などの管理タスクの実行を容易にします。詳細については、第5章 *クラスタリソースの設定と管理(GUI)*(61 ページ)を参照してください。

HA Web Konsole

Linux以外のコンピュータからも、Linuxクラスタを管理できるWebベースのユーザインターフェイス。このインターフェイスは、システムにグラフィックユーザインターフェイスがない場合も理想的なソリューションです。詳細については、第7章 *Webインターフェイスによるクラスタリソースの管理*(119 ページ)を参照してください。

crm

強力な統合コマンドラインインターフェイス。リソースの設定とすべての監視または管理タスクの実行に使用できます。詳細については、第6章ク

ラスタリソースの設定と管理(コマンドライン)(95 ページ)を参照してください。

1.2 利点

High Availability Extensionでは最大16台のLinuxサーバを可用性の高いクラスタ(HAクラスタ)に構成し、クラスタ内の任意のサーバにリソースをダイナミックに切り替えたり、移動することができます。サーバ障害発生時のリソースの自動マイグレーションの設定ができます。また、ハードウェアのトラブルシューティングやワークロードのバランスをとるために、リソースを手動で移動することもできます。

High Availability Extensionは一般的なコンポーネントを使用して、高度な可用性を実現します。アプリケーションと操作をクラスタに統合することによって、運用コストを削減できます。またHigh Availability Extensionではクラスタ全体を一元管理し、変化するワークロード要件に応じてリソースを調整することもできます(手動でのクラスタの「負荷分散」)。3ノード以上でクラスタを構成すると、複数のノードが「ホットスペア」を共用できて無駄がありません。

その他にも重要な利点として、予測できないサービス停止を削減したり、ソフトウェアおよびハードウェアの保守やアップグレードのための計画的なサービス停止を削減できる点が挙げられます。

次に、クラスタによるメリットについて説明します。

- 可用性の向上
- パフォーマンスの改善
- 運用コストの低減
- スケーラビリティ
- 障害回復
- データの保護
- サーバの集約

- ストレージの集約

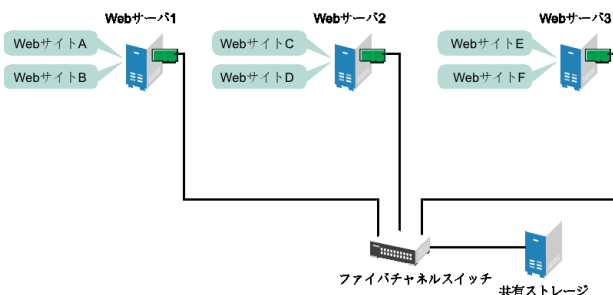
共有ディスクサブシステムにRAIDを導入することによって、共有ディスクの耐障害性を強化できます。

次のシナリオは、High Availability Extensionの利点を紹介するものです。

クラスタシナリオ例

サーバ3台でクラスタが構成され、それぞれのサーバにWebサーバをインストールしたと仮定します。クラスタ内の各サーバが、2つのWebサイトをホストしています。各Webサイトのすべてのデータ、グラフィックス、Webページコンテンツは、クラスタ内の各サーバに接続された、共有ディスクサブシステムに保存されています。次の図は、このクラスタの構成を示しています。

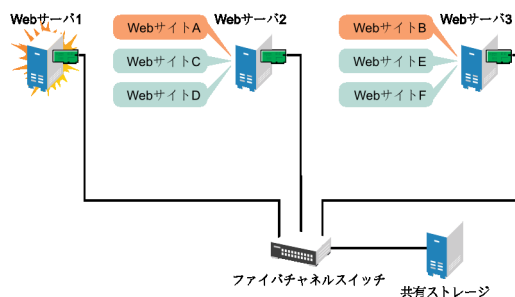
図 1.1 3サーバクラスタ



通常のクラスタ操作では、クラスタ内の各サーバが他のサーバと常に交信し、すべての登録済みリソースを定期的にポーリングして、障害を検出します。

Webサーバ1でハードウェアまたはソフトウェアの障害が発生したため、このサーバを利用してインターネットアクセス、電子メール、および情報収集を行っているユーザの接続が切断されたとします。次の図は、Webサーバ1で障害が発生した場合のリソースの移動を表したものです。

図 1.2 1台のサーバに障害が発生した後の3サーバクラスタ



WebサイトAがWebサーバ2に、WebサイトBがWebサーバ3に移動します。IPアドレスと証明書もWebサーバ2とWebサーバ3に移動します。

クラスタを設定するときに、それぞれのWebサーバがホストしているWebサイトについて、障害発生時の移動先を指定します。先に説明した例では、WebサイトAの移動先としてWebサーバ2が、WebサイトBの移動先としてWebサーバ3が指定されています。このようにして、Webサーバ1によって処理されていたワークロードが、残りのクラスタメンバーに均等に分散され、可用性を維持できます。

Webサーバ1に障害が発生すると、High Availability Extensionソフトウェアは次のような処理を行います。

- 障害を検出し、Webサーバ1が本当に機能しなくなっていることをSTONITHを使用して検証。STONITHは、「Shoot The Other Node In The Head」(他のノードの頭を撃て)の頭字語であり、誤動作しているノードをダウンさせて、それらがクラスタ内に問題を発生させることを防ぎます。
- Webサーバ1にマウントされていた共有データディレクトリを、Webサーバ2およびWebサーバ3に再マウント。
- Webサーバ1で動作していたアプリケーションを、Webサーバ2およびWebサーバ3で再起動。
- IPアドレスをWebサーバ2およびWebサーバ3に移動。

この例では、フェールオーバープロセスが迅速に実行され、ユーザはWebサイトの情報へのアクセスを数秒程度で回復できます。多くの場合、再度ログインする必要はありません。

ここで、Webサーバ1で発生した問題が解決し、通常に操作できる状態に戻たと仮定します。WebサイトAおよびWebサイトBは、Webサーバ1に自動的にフェールバック(復帰)することも、そのままの状態を維持することもできます。これは、リソースの設定方法によって決まります。Webサーバ1へのマイグレーションは多少のダウンタイムを伴うため、High Availability Extensionではサービス中断がほとんど、またはまったく発生しないタイミングまでマイグレーションを延期することもできます。いずれの場合でも利点と欠点があります。

High Availability Extensionはリソースマイグレーション機能も提供しています。アプリケーション、Webサイトなどをシステム管理の必要性に応じて、クラスタ内の他のサーバに移動することができます。

たとえば、WebサイトAまたはWebサイトBをWebサーバ1からクラスタ内の他のサーバに手動で移動することができます。これは、Webサーバ1のアップグレードや定期メンテナンスを実施する場合、また、Webサイトのパフォーマンスやアクセスを向上させる場合に有効な機能です。

1.3 クラスタ構成: ストレージ

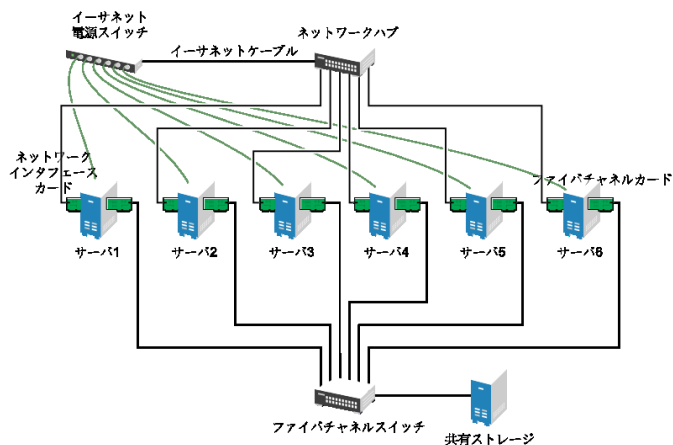
High Availability Extensionでのクラスタ構成には、共有ディスクサブシステムが含まれる場合と含まれない場合があります。共有ディスクサブシステムの接続には、高速ファイバチャネルカード、ケーブル、およびスイッチを使用でき、また構成にはiSCSIを使用することができます。サーバの障害時には、クラスタ内の別の指定されたサーバが、障害の発生したサーバにマウントされていた共有ディスクディレクトリを自動的にマウントします。この機能によって、ネットワークユーザは、共有ディスクサブシステム上のディレクトリに対するアクセスを中断することなく実行できます。

重要項目: cLVMを伴う共有ディスクサブシステム

共有ディスクサブシステムをcLVMと使用する場合、クラスタ内の、アクセスが必要なすべてのサーバにそのサブシステムを接続する必要があります。

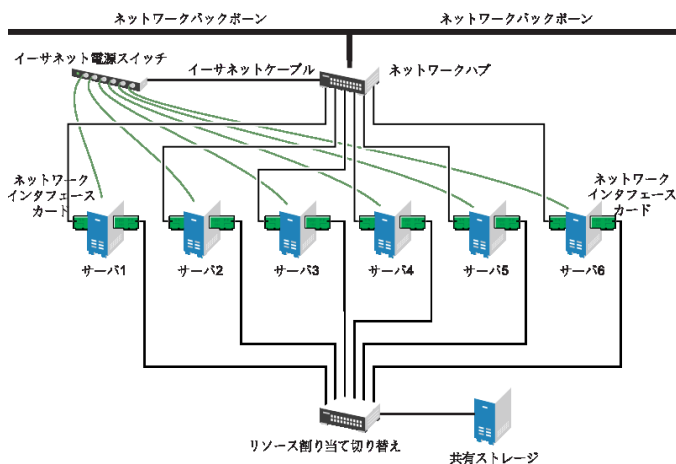
一般的なリソースの例としては、データ、アプリケーション、およびサービスなどがあります。次の図は、一般的なファイバチャネルクラスタの構成を表したものです。

図 1.3 一般的なファイバチャネルクラスタの構成



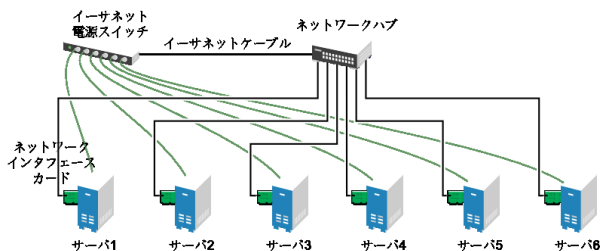
ファイバチャネルは最も高いパフォーマンスを提供しますが、iSCSIを利用するようにクラスタを設定することもできます。iSCSIは低コストなストレージエリアネットワーク(SAN)を作成するための方法として、ファイバチャネルの代わりに使用できます。次の図は、一般的なiSCSIクラスタの構成を表したものです。

図 1.4 一般的なiSCSIクラスタの構成



ほとんどのクラスタには共有ディスクサブシステムが含まれていますが、共有ディスクサブシステムなしのクラスタを作成することもできます。次の図は、共有ディスクサブシステムなしのクラスタを表したものです。

図 1.5 共有ストレージなしの一般的なクラスタ構成



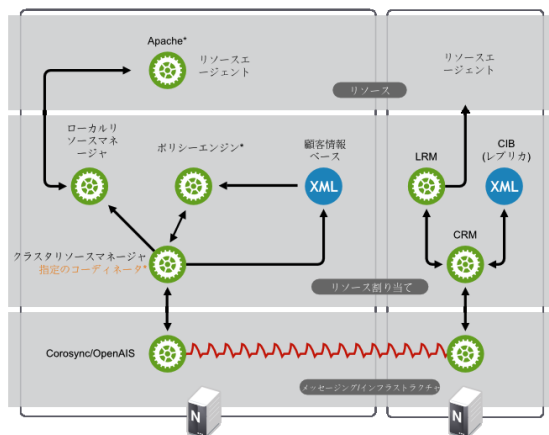
1.4 アーキテクチャ

このセクションではHigh Availability Extensionアーキテクチャについて簡単に説明します。アーキテクチャコンポーネントと、その相互運用方法について説明します。

1.4.1 アーキテクチャ層

High Availability Extensionは階層化されたアーキテクチャになっています。図 1.6 「アーキテクチャ」 (13 ページ)に異なる層と関連するコンポーネントを示します。

図 1.6 アーキテクチャ



メッセージングおよびインフラストラクチャ層

プライマリまたは最初の層は、メッセージングおよびインフラストラクチャの層で、OpenAIS層とも呼ばれます。この層には、「I'm alive」信号やその他の情報を含むメッセージを送信するコンポーネントが含まれます。High Availability Extensionのプログラムはメッセージングおよびインフラストラクチャ層に常駐しています。

リソース割り当て層

次の層はリソース割り当て層です。この層は最も複雑で、次のコンポーネントから構成されています。

CRM (クラスターリソースマネージャ)

リソース割り当て層のすべてのアクションは、クラスターリソースマネージャを通過します。リソース割り当て層の他のコンポーネント(または上位層のコンポーネント)による通信の必要性が発生した場合は、ローカルCRM経由で行います。

CRMは各ノードにCIB (クラスタ情報ベース) (14 ページ)を持っており、ここにはすべてのクラスタオプション、ノード、リソース、関係の定義や現在の状態が含まれています。クラスタ内の1つのCRMがDC(指定コーディネータ)として選択され、マスタCIBがそこに保存されます。クラスタ内のその他すべてのCIBはマスタCIBのレプリカです。CIBに対する通常の読み書き操作は、マスタCIBによってシリアルに処理されます。DCは、ノードのフェンシングやリソースの移動など、クラスタ全体におよぶ変更が必要かどうかを判断できる、クラスタ内で唯一のエンティティです。

CIB (クラスタ情報ベース)

クラスタ情報ベースは、メモリ内でクラスタ全体の設定や現在の状態をXML形式で表すものです。すべてのクラスタオプション、ノード、リソース、制約、相互関係の定義が含まれます。CIBはすべてのクラスタノードへの更新の同期化も行います。DCが管理するマスタCIBがクラスタ内に1つあります。他のすべてのノードにはCIBのレプリカが含まれます。

PE (ポリシーエンジン)

指定コーディネータがクラスタ全体におよぶ変更を行う(新しいCIBに対応する)ことが必要になるたびに、ポリシーエンジンは現在の状態と設定に基づき、クラスタの次の状態を計算します。PEは(リソース)アクションのリストと、次のクラスタ状態に移るために必要な依存性を含む遷移グラフも作成します。PEはDCのフェールオーバー速度を上げるため、各ノードで実行されます。

LRM(ローカルリソースマネージャ)

LRMはCRMに代わってローカルリソースエージェントを呼び出します(「リソース層」 (15 ページ)を参照)。そのため、操作の開始、停止、監視を行い、結果をCRMに報告します。リソースエージェントに対してサポートされているスクリプト標準規格(OCF、LSB、Heartbeat Version 1)間の違いも非表示にします。LRMはそのローカルノード上のすべてのリソース関連情報の信頼できるソースです。

リソース層

最も上位の層はリソース層です。リソース層には1つ以上のリソースエージェント(RA)が含まれます。リソースエージェントは、一定の種類のサービス(リソース)を開始、停止、監視するために作成されたプログラム(通常はシェルスクリプト)です。リソースエージェントの呼び出しはLRMだけが行います。サードパーティはファイルシステム内の定義された場所に独自のエージェントを配置して、自社ソフトウェア用に、すぐに使えるクラスタ統合機能を提供することができます。

1.4.2 プロセスフロー

SUSE Linux Enterprise High Availability ExtensionはPacemakerをCRMとして使用します。CRMは各クラスタノード上にインスタンスを持つデーモン(crmd)として実装されます。Pacemakerは、マスタとして動作するcrmdインスタンスを1つ選択することにより、クラスタのすべての意思決定を一元化します。選択したcrmdプロセス(またはその下のノード)で障害が発生したら、新しいcrmdプロセスが確立されます。

クラスタの構成とクラスタ内のすべてのリソースの現在の状態を反映したCIBが、各ノードに保存されます。CIBのコンテンツはクラスタ全体で自動的に同期化されます。

クラスタ内で実行するアクションの多くは、クラスタ全体におよぼ変更を伴います。これらのアクションにはクラスタリソースの追加や削除、リソース制約の変更などがあります。このようなアクションを実行する場合は、クラスタ内でどのような変化が発生するのかを理解することが重要です。

たとえば、クラスタIPアドレスリソースを追加するとします。そのためには、コマンドラインツールかGUIを使用してCIBを変更できます。DC上でアクションを実行する必要はなく、クラスタ内の任意のノードでいずれかのツールを使用すれば、DCに反映されます。そして、DCがすべてのクラスタノードにCIBの変更を複製します。

CIBの情報に基づき、PEがクラスタの理想的な状態と実行方法を計算し、指示リストをDCに送ります。DCはメッセージング/インフラストラクチャ層を介してコマンドを送信し、他のノードのcrmdピアがこれらのコマンドを受信します。各crmdはLRM(lrmdとして実装)を使用してリソースを変更します。

lrmdはクラスタに対応しておらず、リソースエージェント(スクリプト)と直接通信します。

すべてのピアノードは操作結果をDCに返送します。DCが、すべての必要な操作がクラスタ内で成功したことを確認すると、クラスタはアイドル状態に戻り、次のイベントを待機します。予定通り実行されなかった操作があれば、CIBに記録された新しい情報を元に、PEを再度呼び出します。

場合によっては、共有データの保護や完全なリソース復旧のためにノードの電源を切らなければならないことがあります。このPacemakerにはフェンシングサブシステムとしてstonithdが内蔵されています。STONITHは「Shoot The Other Node In The Head」(他のノードの即時強制終了)の略語で、通常はリモート電源スイッチを使用して実装されます。Pacemakerでは、STONITHデバイスは、その障害を簡単に監視できるように、リソースとしてモデル化(そしてCIB内で設定)されます。ただし、STONITHトポロジの把握はstonithdが担当し、そのクライアントはノードのフェンシングを要求するだけであり、残りの作業はstonithdが実行します。

はじめに

以降では、システム要件とHigh Availability Extensionをインストールする前の準備について説明します。クラスタのインストールとセットアップのための基本手順の概要を説明します。

2.1 ハードウェア要件

次のリストは、SUSE® Linux Enterprise High Availability Extensionに基づくクラスタのハードウェア要件を示します。これらの要件は、最低のハードウェア構成を表しています。クラスタの使用方法によっては、ハードウェアを追加しなければならないこともあります。

- 2.2項「ソフトウェアの必要条件」(18 ページ)に指定されたソフトウェアを搭載した1～16台のLinuxサーバ。サーバのハードウェア構成(メモリ、ディスクスペースなど)は、同一である必要はありません
- 少なくとも2つのTCP/IP通信メディア。クラスタノードは通信にマルチキャストを使用するので、ネットワーク装置はマルチキャストをサポートする必要があります。通信メディアは100Mbit/s以上のデータレートをサポートする必要があります。可能ならば、Ethernetチャンネルをボンドします。
- オプション:クラスタ内の、アクセスが必要なすべてのサーバに接続された、共有ディスクサブシステム。
- STONITHメカニズム。STONITHは「Shoot the other node in the head」の略です。STONITHデバイスとは、クライアントが停止または誤動作してい

るとみなしたノードをリセットするために使用する電源スイッチです。ハートビートを実行していないノードのリセットは、ハングして停止したようにしか見えないノードによるデータ破損を防ぐ、唯一の信頼できる方法です。

詳細については、第9章 フェンシングとSTONITH (131 ページ)を参照してください。

2.2 ソフトウェアの必要条件

次のソフトウェア要件を満たしていることを確認してください。

- クラスタの一部となるすべてのノードに、使用できるすべてのオンライン更新がインストールされた、SUSE® Linux Enterprise Server 11 SP1。
- クラスタの一部となるすべてのノードに、使用できるすべてのオンライン更新がインストールされた、SUSE Linux Enterprise High Availability Extension 11 SP1。

2.3 共有ディスクのシステム要件

クラスタで、データの可用性を高めたい場合は、共有ディスクシステム (SAN:Storage Area Network)の利用をお勧めします。共有ディスクシステムを使用する場合は、次の要件を満たしていることを確認してください。

- メーカーの指示のに従い、共有ディスクシステムが適切に設定され、正しく動作していることを確認します。
- 共有ディスクシステム中のディスクを、ミラーリングまたはRAIDを使用して耐障害性が高められるように設定してください。ハードウェアベースのRAIDをお勧めします。ホストベースのソフトウェアRAIDはどの構成でもサポートされていません。
- 共有ディスクシステムのアクセスにiSCSIを使用している場合、iSCSIイニシエータとターゲットを正しく設定していることを確認します。
- DRBDを使用してデータを2台のマシンに分配するミラーリングRAIDシステムを実装する際は、複製されたデバイスのみにアクセスしてください。

クラスタの残りが提供される冗長性を利用する、同じ(ボンドされた)NICを使用します。

2.4 準備作業

High Availability Extensionをインストールする前に、次の準備手順を実行します。

- クラスタ内の各サーバの/etc/hostsファイルを編集することにより、ホスト名の解決を設定し、静的ホスト情報を使用します。詳細については、<http://www.novell.com/documentation>にある『*SUSE Linux Enterprise Server admin;*』を参照してください。特に、「*Basic Networking > Configuring Hostname and DNS*」の章を参照してください。

クラスタのメンバーが名前で互いを見つけられることが重要です。名前を使用できない場合、内部クラスタ通信は失敗します。

- クラスタノードをクラスタ外部のタイムサーバと同期させ、時刻同期を構成します。詳細については、<http://www.novell.com/documentation>にある『*SUSE Linux Enterprise Server 管理ガイド*』を参照してください。特に、「*Time Synchronization with NTP*」の章を参照してください。

クラスタノードは、タイムサーバを時刻同期ソースとして使用します。

2.5 概要:クラスタのインストールとセットアップ

準備完了後、SUSE® Linux Enterprise High Availability Extensionでクラスタをインストールして設定するには、次の基本手順が必要です。

1. SUSE® Linux Enterprise ServerとSUSE® Linux Enterprise High Availability ExtensionをSUSE Linux Enterprise Server上にアドオンとしてインストール。詳細については、3.1項「High Availability Extensionのインストール」(21 ページ)を参照してください。
2. クラスタの初期セットアップ (22 ページ)

3. クラスタをオンラインにする (30 ページ)

4. グローバルクラスタオプションの設定とクラスタリソースの追加

両方とも、グラフィックユーザインターフェイス(GUI)またはコマンドラインツールを使用して実行できます。詳細については、第5章 クラスタリソースの設定と管理(GUI) (61 ページ)または第6章 クラスタリソースの設定と管理(コマンドライン) (95 ページ)を参照してください。

5. フェンシングとSTONITHによってデータが破損することを防ぐため、STONITHデバイスをリソースとして構成します。詳細については、第9章 フェンシングとSTONITH (131 ページ)を参照してください。

要件によっては、次のファイルシステムとストレージ関係のコンポーネントをクラスタに設定することもできます。

- 共有ディスク(SAN (Storage Area Network)上)でファイルシステムを作成します。必要な場合は、それらのファイルシステムをクラスタリソースとして設定します。
- クラスタ対応型のファイルシステムが必要な場合は、OCFS2を使用します。
- クラスタによる論理ボリュームマネージャ(LVM)を使用した共有ストレージ管理を可能にするには、LVMのクラスタ化拡張機能の集まりであるcLVMを使用します。
- データの整合性を保護するには、フェンシングメカニズムの使用と排他的ストレージアクセスの確保によって、ストレージの保護を実装します。
- 必要な場合は、DRBDによるデータレプリケーションを使用します。

詳細については、パートIII「ストレージおよびデータレプリケーション」(159 ページ)を参照してください。

YaSTによるインストールと基本設定

High Availability クラスタに必要なソフトウェアをインストールするには、2つの方法があります。コマンドラインから `zypper` を使用するか、または YaST のグラフィカルユーザインタフェースを使用します。クラスタに属するすべてのノードにソフトウェアをインストールしたら、次は、ノードが互いに通信できるようにクラスタを初期設定し、クラスタをオンラインにするために必要なサービスを開始します。初期のクラスタ設定は、手動か(設定ファイルの編集とコピー)、または YaST クラスタモジュールを使用して実行できます。

この章では、最初から SUSE Linux Enterprise High Availability Extension 11 SP1 を使用して新規のインストールと設定を行う方法について説明します。旧バージョンの SUSE Linux Enterprise High Availability Extension を実行する既存クラスタを移行する場合や、実行中のクラスタに属するノードでソフトウェアパッケージを更新する場合は、付録B クラスタの最新製品バージョンへのアップグレード(373 ページ)を参照してください。

3.1 High Availability Extension のインストール

High Availability Extension によるクラスタの設定と管理に必要なパッケージは、High Availability インストールパターンに含まれています。このパターンは、SUSE® Linux Enterprise High Availability Extension がアドオンとしてインストールされた後でのみ利用できます。アドオン製品のインストール方法については、<http://www.novell.com/documentation> で入手できる『SUSE

Linux Enterprise 11 SP1 導入ガイド』を参照してください。特に、「アドオン製品のインストール」の章を参照してください。

注記: ソフトウェアパッケージのインストール

High Availability クラスタに必要なソフトウェアパッケージはクラスタノードに自動的にコピーされません。

SUSE® Linux Enterprise Server 11 SP1 と SUSE® Linux Enterprise High Availability Extension 11 SP1 を、クラスタに属するすべてのノードに手動インストールしたくない場合は、AutoYaST を使用して、既存ノードのクローンを作成します。詳細については、3.4 項「AutoYaST による大量展開」(31 ページ)を参照してください。

手順 3.1 High Availability パターンをインストールする

- 1 YaST を root ユーザとして開始し、[ソフトウェア] > [ソフトウェア管理] の順に選択します。

または、コマンドラインで `yast2 sw_single` を使用して、YaST パッケージマネージャを root として起動します。

- 2 [フィルタ] リストで、[パターン] を選択して、パターンリストで [高可用性] をアクティブにします。
- 3 [同意する] をクリックして、パッケージのインストールを開始します。

3.2 クラスタの初期セットアップ

HA パッケージのインストール後は、初期クラスタ設定に進みます。これには、次の基本ステップがあります。

- 1 通信チャネルの定義 (23 ページ)
- 2 認証設定の定義 (25 ページ)
- 3 すべてのノードへの設定の転送 (26 ページ)

次の手順では、YaSTクラスタモジュールを使用して、各ステップを実行します。クラスタの設定ダイアログにアクセスするには、YaSTをrootとして起動し、[High Availability] > [クラスタ] の順に選択します。または、コマンドラインでyast2 clusterを使用して、YaSTクラスタモジュールをrootとして起動します。

初めてクラスタモジュールを起動した場合は、モジュールが、ウィザードのように、基本設定に必要なすべてのステップをガイドします。そうでない場合は、左パネルのカテゴリをクリックして、ステップごとに設定オプションにアクセスします。

3.2.1 通信チャネルの定義

クラスタノード間で正常な通信を行うには、少なくとも1つの通信チャネルを定義します。ただし、2つ以上の冗長パスを使用して通信を確立することを推奨します(このためには、ネットワークデバイスボンディングを使用するか、Corosyncで2つ目の通信チャネルを追加します)。通信チャネルごとに、次のパラメータを定義する必要があります。

バインドネットワークアドレス(bindnetaddr)

バインド先のネットワークアドレス。クラスタ間の設定ファイルの共有を軽減するため、OpenAISはネットワークインタフェースネットマスクを使用して、ネットワークのルーティングに使用されるアドレスビットのみをマスクします。クラスタマルチキャストに使用するサブネットに値を設定します。

マルチキャストアドレス(mcastaddr)

IPv4アドレスまたはIPv6アドレス。

マルチキャストポート(mcastport)

mcastaddr用に指定されたUDPポート。

クラスタ内のすべてのノードは、同じマルチキャストアドレスおよび同じポート番号の使用によって、互いに認識します。別のクラスタは、別のマルチキャストアドレスを使用します。

Corosyncとの冗長な通信を設定するには、/etc/corosync/corosync.confで、複数のinterfaceセクションを、セクションごとに異なるringnumberを付けて定義する必要があります。RRP (Redundant Ring Protocol)を使用して、これ

らのインターフェイスの使い方をクラスタに指示します。**RRP**は、3つのモード(`rrp_mode`)を持つことができます。`active`に設定されている場合は、**Corosync**がすべてのインターフェイスをアクティブに使用します。`passive`に設定されると、**Corosync**は、最初のリングが失敗した場合のみ、2番目のインターフェイスを使用します。`rrp_mode`が`none`に設定されると、**RRP**は無効化されます。**RRP**では、2つの物理的に別個のネットワークが通信に使用されます。1つのネットワークが失敗しても、クラスタノードは、もう一方のネットワークを介して通信できます。

複数のリングが設定されている場合は、各ノードが複数のIPアドレスを持つことができます。`rrp_mode`が有効になるとただちに、デフォルトでは、ノード間の通信に、(TCPの代わりに)**SCTP** (Stream Control Transmission Protocol)が使用されます。

手順 3.2 通信チャネルを定義する

- 1 **YaST** クラスタモジュール内で、**[通信チャネル]** カテゴリに切り替えます。
- 2 すべてのクラスタノードに使用する、**[Bind Network Address(バインドネットワークアドレス)]**、**[Multicast Address(マルチキャストアドレス)]**、**[Multicast Port(マルチキャストポート)]** を定義します。

The screenshot shows the 'Cluster - 通信チャネル' (Cluster - Communication Channels) configuration window in YaST. The window is divided into two main sections: 'チャネル' (Channel) and 'ノードID' (Node ID). In the 'チャネル' section, the 'Bind Network Address' is set to '192.168.8.0', the 'Multicast Address' is '192.168.0.254', and the 'Multicast Port' is '5405'. There is an unchecked checkbox for '冗長チャネル' (Redundant Channel). In the 'ノードID' section, the 'Auto Generate Node ID' checkbox is unchecked, and the 'Node ID' is set to '2'. The 'rrp mode' is set to 'none'. At the bottom, there are buttons for 'ヘルプ' (Help), 'キャンセル(G)' (Cancel), and '完了(F)' (Finish).

3 2つ目のチャンネルを定義する場合は、次の手順を実行します。

3a [冗長チャンネル] を有効にします。

3b [バインドネットワークアドレス]、[マルチキャストアドレス]、および [マルチキャストポート] を冗長チャンネル用に定義します。

3c 使用する [rrp_mode] を選択します。RRPを無効にするには、[なし] を選択します。モードの詳細については、[ヘルプ] をクリックしてください。

RRPを使用する場合、`/etc/corosync/corosync.conf`で、プライマリリング(設定した最初のチャンネル)には`ringnumber 0`が設定され、2つ目のリング(冗長チャンネル)には`ringnumber 1`が設定されます。

4 [ノードIDの自動生成] を有効にして、クラスタタイプごとに一意のIDを自動生成します。

5 既存クラスタの通信チャンネルを変更するだけの場合は、[完了] をクリックして、設定を`/etc/corosync/corosync.conf`に書き込み、YaSTクラスタモジュールを終了します。YaSTはファイアウォール設定も自動的に調整し、マルチキャストに使用されるUDPポートを開きます。

6 さらにクラスタ設定を続ける場合は、手順3.3「安全な認証を有効にする」(25 ページ)に進みます。

3.2.2 認証設定の定義

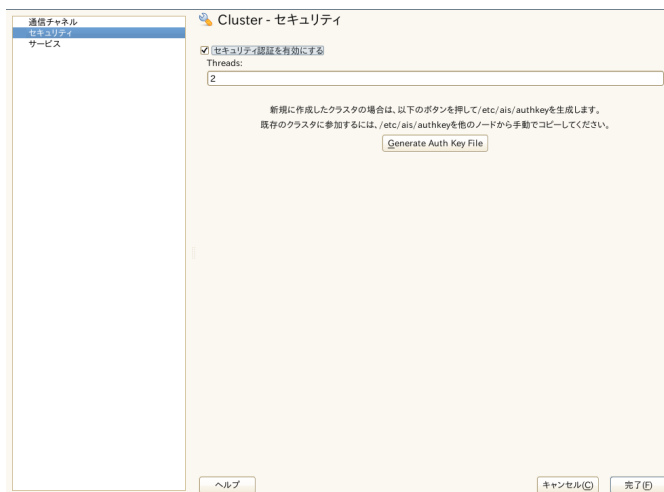
次のステップとして、クラスタの認証設定を定義します。共有の秘密項目が必要なHMAC/SHA1認証を使用して、メッセージを保護し、認証することができます。指定した認証キー(パスワード)が、クラスタ中のすべてのノードで使用されます。

手順 3.3 安全な認証を有効にする

1 YaSTクラスタモジュール内で、[Security] カテゴリに切り替えます。

2 [安全認証の有効化] をオンにします。

- 3 新しく作成したクラスタの場合は、[認証キーファイルの生成] をクリックします。これによって、認証キーが作成され、`/etc/corosync/authkey`に書き込まれます。



- 4 認証設定を変更するだけの場合は、[完了] をクリックすると、`/etc/corosync/corosync.conf`に設定が書き込まれ、YaSTクラスタモジュールが終了します。
- 5 さらにクラスタ設定を続ける場合は、3.2.3項「すべてのノードへの設定の転送」(26 ページ)に進みます。

3.2.3 すべてのノードへの設定の転送

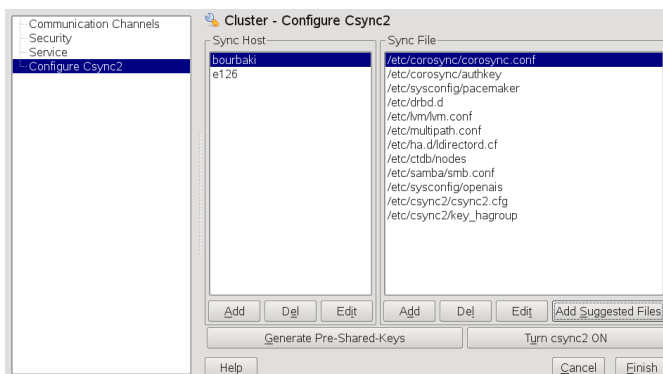
結果として生成された設定ファイルをすべてのノードに手動でコピーする代わりに、`csync2`ツールを使用して、クラスタ内のすべてのノードにレプリケートします。`Csync2`は、同期グループ別にソートされた任意の数のホストを操作できます。各同期グループは、メンバホストの独自のリストとその包含/除外パターン(同期グループ内でどのファイルを同期するか定義するパターン)を持っています。グループ、各グループに属するホスト名、および各グループの包含/除外ルールは、`Csync2`設定ファイル`/etc/csync2/csync2.cfg`で指定されます。

Csync2は、認証には、同期グループ内でIPアドレスと事前共有キーを使用します。管理者は、同期グループごとに1つのキーファイルを生成し、そのファイルをすべてのグループメンバにコピーする必要があります。

Csync2の詳細については、<http://oss.linbit.com/csync2/paper.pdf>を参照してください。

手順 3.4 YaSTでCsync2を設定する

- 1 YaSTクラスタモジュール内で、[Csync2] カテゴリに切り替えます。
- 2 同期グループを指定するには、[同期ホスト] グループで[追加]をクリックし、クラスタ内のすべてのノードのローカルホスト名を入力します。ノードごとに、hostnameコマンドから返された文字列を正確に使用する必要があります。
- 3 [事前共有キーの生成] をクリックして、同期グループのキーファイルを生成します。キーファイルは、/etc/csync2/key_hagroupに書き込まれます。このファイルは、作成後に、クラスタのすべてのメンバーに手動でコピーする必要があります。
- 4 すべてのノード間で、通常、同期される必要のあるファイルを[同期ファイル] リストに入れるには、[推奨ファイルの追加] をクリックします。



- 5 同期するファイルのリストからファイルを[編集]、[追加]、または[削除] する場合は、該当する各ボタンを使用します。ファイルごとに絶対パス名を入力する必要があります。
- 6 [Csync2をオンにする] をクリックして、Csync2をアクティブにします。これによって、ブート時にCsync2が自動的に開始します。
- 7 すべてのオプションが望みどおりに設定されたら、[完了] をクリックして、YaSTクラスタモジュールを終了します。YaSTがCsync2の設定内容を/etc/csync2/csync2.cfgに書き込みます。

Csync2の設定後は、次に示すように、コマンドラインから同期プロセスを開始します。

手順 3.5 Csync2で設定ファイルを同期する

Csync2でファイルを正常に同期するには、次の前提条件を満たしておく必要があります。

- 同じCsync2設定をすべてのノードで使用できる必要があります。Csync2で同期するファイルのリストに/etc/csync2/csync2.cfgを含めるか、手順3.4「YaSTでCsync2を設定する」(27 ページ)で説明されているようにファイルを設定した後で、すべてのノードに手動でファイルをコピーします。
- ステップ 3 (27 ページ)で1つのノードに作成した/etc/csync2/key_hagroupファイルを、クラスタ内のすべてのノードにコピーしてください。このファイルは、Csync2による認証で必要になります。ただし、すべてのノードで同じファイルでなければならないので、他のノードではファイルを再生成しないでください。
- xinetdがすべてのノードで実行されていることを確認してください。Csync2は、このデーモンに依存するからです。次のコマンドを使用して、rootとしてxinetdを起動します。

```
rcxinetd start
```

注記: ブート時でのサービスの開始

Csync2とxinetdをブート時に自動的に開始したい場合は、すべてのノードで、次のコマンドを実行します。


```
chkconfig csync2 on
chkconfig xinetd on
```

- 1 次のコマンドをノードの1つで実行することで、ファイルの同期を開始します。

```
csync2 -xv
```

これによって、すべてのファイルが一度に同期されます。すべてのファイルが正常に同期できると、**Csync2**がエラーなしで終了します。

同期対象の1つ以上のファイルが(現在のノードだけでなく)他のノードでも変更されている場合は、**Csync2**から競合が報告されます。次の出力とよく似た出力が表示されます。

```
While syncing file /etc/corosync/corosync.conf:
ERROR from peer hex-14: File is also marked dirty here!
Finished with 1 errors.
```

- 2 現在のノードのファイルバージョンが「最良」だと確信する場合は、そのファイルを強制して再同期を行い、競合を解決できます。

```
csync2 -f /etc/corosync/corosync.conf
csync2 -x
```

Csync2オプションの詳細については、`csync2 -help`を実行してください。

注記: 同期のトリガ

Csync2は、ノード間のファイルを絶えず同期しているわけではありません。同期が必要なファイルを更新するたびに、それらのファイルを手動で再同期する必要があります。

クラスタ内のすべてのノードにキーファイルを同期させたら、3.3項「クラスタをオンラインにする」(30 ページ)で説明されているように、基本的なサービスを開始して、クラスタをオンラインにします。

3.2.4 サービスの開始

YaSTクラスタモジュールでは、オプションとして、ブート時にノードで一定のサービスを開始するかどうか定義できます。このモジュールでは、サービ

を手動で開始および停止することもできます(そのためにコマンドラインを使用したくない場合)。クラスタノードをオンラインにし、クラスタリソースマネージャを起動するには、OpenAISをサービスとして開始する必要があります。

手順 3.6 サービスを開始または停止する

- 1 YaSTクラスタモジュール内で、[サービス] カテゴリに切り替えます。
- 2 このクラスタノードがブートするたびにOpenAISを開始するには、[ブート] グループで該当するオプションを選択します。
- 3 クラスタリソースの設定、管理、および監視に、Pacemaker GUIを使用する場合は、[*mgmt*dも開始する] をオンにします。このデーモンは、GUIのために必要です。
- 4 OpenAISをただちに開始または停止するには、それぞれのボタンをクリックします。
- 5 [完了] をクリックして、YaSTクラスタモジュールを終了します。

[ブート] グループで、[オフ] を選択した場合は、このノードがブートするたびに、手動で、OpenAISを開始する必要があります。OpenAISを手動で開始するには、`rcopenais start`コマンドを使用します。

3.3 クラスタをオンラインにする

初期クラスタ設定の完了後は、スタックをオンラインにするために必要なサービスを開始できます。

手順 3.7 OpenAIS/Corosyncを開始し、ステータスをチェックする

- 1 クラスタの各ノードで次のコマンドを実行して、OpenAIS/Corosyncを開始します。

```
rcopenais start
```

- 2 ノードの1つで、次のコマンドを使用してクラスタの状態を確認します。

```
crm_mon
```

すべてのノードがオンラインの場合、出力は次のようになります。

```
=====
Last updated: Tue Mar  2 18:35:34 2010
Stack: openais
Current DC: e229 - partition with quorum
Version: 1.1.1-530add2a3721a0ecccb24660a97dbfdaa3e68f51
2 Nodes configured, 2 expected votes
0 Resources configured.
=====

Online: [ e231 e229 ]
```

この出力は、クラスタリソースマネージャが起動し、リソースを管理できる状態にあることを示しています。

基本設定を完了し、ノードがオンラインになったら、クラスタリソースの設定を開始できます。crmコマンドラインツールか、またはグラフィックユーザインターフェイスを使用します。詳細は、第5章 クラスタリソースの設定と管理(GUI) (61 ページ)または第6章 クラスタリソースの設定と管理(コマンドライン) (95 ページ)を参照してください。

3.4 AutoYaSTによる大量展開

AutoYaSTは、ユーザの介入なしで、1つ以上のSUSE Linux Enterpriseシステムを自動的にインストールするためのシステムです。SUSE Linux Enterpriseでは、インストールと設定のデータを含むAutoYaSTプロファイルを作成できます。プロファイルによって、インストールする対象と、インストールしたシステムが最終的に完全に使用準備が整ったシステムになるように設定する方法がAutoYaSTに指示されます。このプロファイルは、さまざまな方法で大量展開に使用できます。

さまざまなシナリオでのAutoYaSTの使用方法的詳細については、<http://www.novell.com/documentation>にある『SUSE Linux Enterprise 11 SP1 導入ガイド』を参照してください。特に、「自動インストール」の章を参照してください。

手順 3.8 AutoYaSTでクラスタノードのクローンを作成する

既存ノードのクローンであるクラスタノードを展開する場合は、次の手順を使用できます。クローンしたノードには、同じパッケージがインストールされ、クローンノードは同じシステム設定を持つことになります。

同じでないハードウェア上にクラスタノードを展開する必要がある場合は、<http://www.novell.com/documentation>にある『SUSE Linux Enterprise 11 SP1 導入ガイド』の「ルールベースの自動インストール」セクションを参照してください。

重要項目: 同一のハードウェアを使用している環境

このシナリオでは、同一のハードウェア設定を持つコンピュータ群にSUSE Linux Enterprise High Availability Extension 11 SP1を展開すると想定します。

- 1** クローンを作成するノードが、3.1項「High Availability Extensionのインストール」(21 ページ)と3.2項「クラスタの初期セットアップ」(22 ページ)の説明に従って、正しくインストールされ、設定されていることを確認します。
- 2** 単純な大量インストールの場合は『SUSE Linux Enterprise 11 SP1 導入ガイド』に概略されている説明に従います。これには、次の基本ステップがあります。
 - 2a** AutoYaSTプロファイルを作成します。AutoYaST GUIを使用して、既存のシステム設定からプロファイルを作成し、変更します。
AutoYaSTでは、*[High Availability]* モジュールを選択し、*[クローン]* ボタンをクリックします。必要な場合は、他のモジュールの設定を調整し、その結果のコントロールファイルをXMLとして保存します。
 - 2b** AutoYaSTプロファイルのソースと、他のノードのインストールルーチンに渡すパラメータを決定します。
 - 2c** SUSE Linux Enterprise ServerとSUSE Linux Enterprise High Availability Extensionのインストールデータのソースを決定します。
 - 2d** 自動インストールのブートシナリオを決定し、設定します。
 - 2e** パラメータを手動で追加するか、またはinfoファイルを作成することにより、インストールルーチンにコマンド行を渡します。
 - 2f** 自動インストールプロセスを開始および監視します。

クローンのインストールに成功したら、次の手順を実行して、クローンノードをクラスタに加えます。

手順 3.9 クローンノードをオンラインにする

- 1 3.2.3項「すべてのノードへの設定の転送」(26 ページ)の説明に従って、Csync2を使用して、設定済みのノードからクローンノードへ重要な設定ファイルを転送します。
- 2 3.3項「クラスタをオンラインにする」(30 ページ)の説明に従って、クローンノードでOpenAISサービスを開始して、ノードをオンラインにします。

これで、`/etc/corosync/corosync.config`ファイルがCsync2を介してクローンノードに適用されたので、クローンノードがクラスタに加わります。CIBは、クラスタノード間で自動的に同期されます。

パート II. 設定および管理

設定および管理の基本事項

HAクラスタの主な目的はユーザサービスの管理です。ユーザサービスの典型的な例は、Apache Webサーバまたはデータベースです。サービスとは、ユーザの観点からすると、指示に基づいて特別な何かを行うことを意味していますが、クラスタにとっては開始や停止できるリソースにすぎません。サービスの性質はクラスタには無関係なのです。

この章では、リソースを設定しクラスタを管理する場合に知っておく必要のある基本概念を紹介します。後続の章では、High Availability Extensionが提供する各管理ツールを使用して、主要な設定と管理のタスクを実行する方法を説明します。

4.1 グローバルクラスタオプション

グローバルクラスタオプションは、一定の状況下でのクラスタの動作を制御します。それらは、セットとしてグループ化され、Pacemaker GUIやcrmシェルなどのクラスタ管理ツールで表示したり、変更できます。事前に定義されている値は、ほとんどの場合、そのまま保持できます。ただし、クラスタの主要機能を正しく機能させるには、クラスタの基本的なセットアップ後に、次のパラメータを調整する必要があります。

- オプションno-quorum-policy (38 ページ)
- オプションstonith-enabled (39 ページ)

これらのパラメータをGUIで調整する方法については、手順5.1「グローバルクラスタオプションを変更する」(65 ページ)を参照してください。コマンドラインを使用したい場合は、6.2項「グローバルクラスタオプションの設定」(102 ページ)を参照してください。

4.1.1 オプションno-quorum-policy

このグローバルオプションは、クラスタにクォーラムがない(ノードの過半数がパーティションに含まれない)場合どうするかを定義します。

許容値は、次のとおりです。

`ignore`

クォーラム状態がクラスタの動作に全く影響せず、リソース管理が続行されます。

この設定は、次のシナリオで有効です。

- 2ノードクラスタ: 1つのノードに障害が発生すると、常に過半数が失われるため、通常は、構わずクラスタを続行させます。リソースの整合性は、フェンシングの使用で確保されます。フェンシングは、スプリットブレインシナリオも防止します。
- リソース駆動型クラスタ: 冗長な通信チャネルを持つローカルクラスタの場合、スプリットブレインシナリオには一定の確率しかありません。したがって、ノードとの通信の喪失は、ほとんどの場合、そのノードがクラッシュしていること、残りのノードは回復して、リソースのサービスを再開する必要があることを示します。

`no-quorum-policy`が`ignore`に設定されている場合、4ノードクラスタは、サービスが失われる前の3ノードの同時障害を克服できますが、他の設定値では、2ノードの同時障害後はクォーラムを失います。

`freeze`

クォーラムが失われた場合は、クラスタがフリーズします。リソース管理は続行されます。実行中のリソースは停止されません(ただし、イベントの監視に対応して再起動される可能性があります)。ただし、影響を受けたパーティション内では、以後のリソースが開始されません。

一定のリソースが他のノードとの通信に依存しているクラスタの場合(たとえば、OCFS2マウントなど)は、この設定が推奨されます。この場合、デフォルト設定no-quorum-policy=stopは、次のようなシナリオになるので有効ではありません。つまり、ピアノードが到達不能な間はそれらのリソースを停止できなくなります。その代わり、停止の試行は最終的にタイムアウトし、stop failureになり、エスカレートされた復元とフェンシングを引き起こします。

stop (デフォルト値)

クォラムが失われると、影響を受けるクラスタパーティション内のすべてのリソースが整然と停止します。

suicide

影響を受けるクラスタパーティション内のすべてのノードをフェンシングします。

4.1.2 オプションstonith-enabled

このグローバルオプションは、フェンシングを適用して、STONITHデバイスによる、障害ノードや停止できないリソースを持つノードのダウンを許可するかどうか定義します。通常のクラスタ操作には、STONITHデバイスの使用が必要なので、このグローバルオプションは、デフォルトでtrueに設定されています。デフォルト値では、クラスタは、STONITHリソースが定義されていない場合にはリソースの開始を拒否します。

なんらかの理由でフェンシングを無効にする必要がある場合は、stonith-enabledをfalseに設定します。

すべてのグローバルクラスタオプションとそのデフォルト値の概要については、『*Pacemaker 1.0—Configuration Explained*』(<http://clusterlabs.org/wiki/Documentation>)を参照してください。特に、「*Available Cluster Options*」のセクションを参照してください。

4.2 クラスタリソース

クラスタの管理者は、クラスタ内のサーバ上の各リソースや、サーバ上で実行する各アプリケーションに対してクラスタリソースを作成する必要があります。

ます。クラスタリソースには、Webサイト、電子メールサーバ、データベース、ファイルシステム、仮想マシン、およびユーザが常時使用できるその他のサーバベースのアプリケーションまたはサービスなどが含まれます。

4.2.1 リソース管理

リソースは、クラスタで使用する前にセットアップする必要があります。たとえば、Apacheサーバをクラスタリソースとして使用する場合は、まず、Apacheサーバをセットアップし、Apacheの環境設定を完了してから、クラスタで個々のリソースを起動します。

リソースに特定の環境要件がある場合は、それらの要件がすべてのクラスタノードに存在し、同一であることを確認してください。このタイプの設定は、High Availability Extensionでは管理されません。これは、管理者自身が行う必要があります。

注記: クラスタによって管理されるサービスには介入しないでください。

High Availability Extensionでリソースを管理しているとき、同じリソースを他の方法(クラスタ外で、たとえば、手動、ブート、リブートなど)で開始したり、停止してはなりません。High Availability Extensionソフトウェアが、すべてのサービスの開始または停止アクションを実行します。

ただし、サービスが適切に構成されているか確認したい場合は手動で開始しますが、High Availabilityが起動する前に停止してください。

クラスタ内でリソースを設定したら、クラスタ管理ツールを使用して、すべてのリソースを手動で起動、停止、クリーンアップ、削除、または移行します。詳細については、第5章 クラスタリソースの設定と管理(GUI) (61 ページ) または第6章 クラスタリソースの設定と管理(コマンドライン) (95 ページ) を参照してください。

4.2.2 サポートされるリソースエージェント クラス

追加するクラスタリソースごとに、リソースエージェントが準拠する基準を定義する必要があります。リソースエージェントは、提供するサービスを抽

象化し、正確な状態をクラスタに提供するので、クラスタは管理するリソースにコミットしなくて済みます。クラスタは、リソースエージェントに依存して、**start**、**stop**、または**monitor**のコマンドの発行に適宜対応します。

通常、リソースエージェントはシェルスクリプトの形式で配布されます。**High Availability Extension**は次のリソースエージェントクラスをサポートします。

従来のHeartbeat 1リソースエージェント

Heartbeatバージョン1には独自のスタイルのリソースエージェントが付属していました。多くのユーザが独自のエージェントをその表記方法に従って開発したため、このリソースエージェントはまだサポートされています。ただし、可能な場合は構成を**High Availability OCF RA**にマイグレートすることを推奨します。

Linux Standards Base (LSB)スクリプト

LSBリソースエージェントは一般にオペレーティングシステム/ディストリビューションによって提供され、`/etc/init.d`にあります。リソースエージェントをクラスタで使用するには、それらのエージェントが**LSB ini**スクリプトの仕様に準拠している必要があります。たとえば、リソースエージェントには、いくつかのアクションが実装されている必要があります。それらのアクションとして、少なくとも**start**、**stop**、**restart**、**reload**、**force-reload**、**status**があります。詳細については、<http://ldn.linuxfoundation.org/lsb/lsb4-resource-page%23Specification>を参照してください。

これらのサービスの構成は標準化されていません。**High Availability**でLSBスクリプトを使用する場合は、該当のスクリプトの設定方法を理解する必要があります。これに関する情報は、多くの場合、`/usr/share/doc/packages/PACKAGENAME`内の該当パッケージのマニュアルに記載されています。

Open Cluster Framework (OCF)リソースエージェント

OCF RAエージェントは、**High Availability**での使用に最適であり、特に、マスタリソースまたは特殊な監視機能が必要とする場合に適しています。それらのエージェントは、通常、`/usr/lib/ocf/resource.d/provider`にあります。この機能は**LSB**スクリプトの機能と同様です。ただし、環境設定では、常に、パラメータの受け入れと処理を容易にする環境変数が使用されます。OCF仕様は<http://www.opencf.org/cgi-bin/viewcvs.cgi/specs/ra/resource-agent-api.txt?rev=>

`HEAD&content-type=text/vnd.viewcvs-markup`で参照できます(リソースエージェントに関連するため)。OCF仕様には、アクション終了コードの厳密な定義があります。8.3項「OCF戻りコードと障害回復」(127ページ)を参照してください。クラスは、それらの仕様に正確に準拠します。使用できるすべてのOCF RAの詳細なリストは、第19章 *HA OCF Agents* (275 ページ)を参照してください。

すべてのOCFリソースエージェントは少なくともstart、stop、status、monitor、meta-dataのアクションを持つ必要があります。meta-dataアクションは、エージェントの構成方法についての情報を取得します。たとえば、プロバイダheartbeatでIPAddrエージェントの詳細を知りたい場合は、次のコマンドを使用します。

```
OCF_ROOT=/usr/lib/ocf /usr/lib/ocf/resource.d/heartbeat/IPAddr meta-data
```

出力は、XML形式の情報であり、いくつかのセクションを含みます(一般説明、利用可能なパラメータ、エージェント用の利用可能なアクション)。

STONITHリソースエージェント

このクラスは、フェンシング関係のリソース専用に使われます。詳細については、第9章 フェンシングと*STONITH* (131 ページ)を参照してください。

High Availability Extensionで指定されたエージェントはOCF仕様に従って作成されています。

4.2.3 リソースのタイプ

次のリソースタイプを作成できます。

プリミティブ

プリミティブリソースはリソースの中で最も基本的なタイプです。

GUIでプリミティブリソースを作成する方法については、手順5.2「プリミティブリソースを追加する」(66 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.1項「クラスタリソースの作成」(103ページ)を参照してください。

グループ

グループには、一緒の場所で見つけ、連続して開始し、逆の順序で停止する必要のあるリソースセットが含まれます。詳細については、「グループ」(43 ページ)を参照してください。

クローン

クローンは、複数のホスト上でアクティブにできるリソースです。対応するリソースエージェントがサポートしていれば、どのようなリソースもクローン化できます。詳細については、「クローン」(45 ページ)を参照してください。

マスタ

マスタは、クローンリソースの特殊なタイプで、複数のモードを持つことができます。詳細については、「マスタ」(46 ページ)を参照してください。

4.2.4 高度なリソースタイプ

プリミティブは、最も単純なタイプのリソースなので、設定が容易ですが、クラスタ設定には、より高度なリソースタイプ(グループ、クローン、マスタなど)が必要になることがあります。

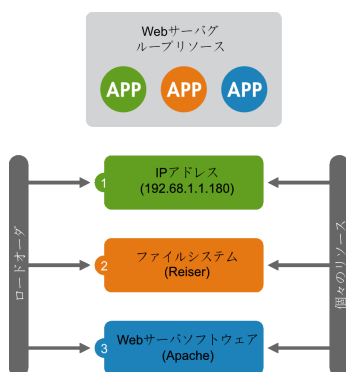
グループ

クラスタリソースの中には、他のコンポーネントやリソースに依存しており、各コンポーネントや各リソースを特定の順序で開始したり、同じサーバ上で一緒に実行しなければならないものもあります。この設定を簡素化するには、グループを使用できます。

例 4.1 Webサーバのリソースグループ

リソースグループの1例として、IPアドレスとファイルシステムを必要とするWebサーバがあります。この場合、各コンポーネントは、個々のクラスタリソースであり、それらが組み合わされてクラスタリソースグループを構成します。リソースグループは1つ以上のサーバで実行し、ソフトウェアやハードウェアの障害発生時には、個々のクラスタリソースと同様に、クラスタ内の別なサーバにフェールオーバーします。

図 4.1 グループリソース



グループには次のプロパティがあります。

を開始/停止する

リソースは認識される順序で開始し、逆の順番で停止します。

依存関係

グループ内のリソースがどこかで開始できない場合は、グループ内のその後の全リソースは実行することができません。

目次

グループにはプリミティブクラスタリソースしか含むことができません。グループには1つ以上のリソースを含む必要があります。空の場合は設定は無効になります。グループリソースの子を参照するには、グループのIDではなく子のIDを使用します。

制約

制約でグループの子を参照することはできますが、通常はグループ名を使用することをお勧めします。

固着性

固着性はグループ内で統合可能なプロパティです。グループ内のアクティブな各メンバーは、グループの合計値に対して固着性を追加します。したがって、デフォルトのresource-stickinessが100で、グループに7つのメンバーがあり、そのうち5つがアクティブな場合は、グループが全体として、スコア500で、現在の場所を優先します。

リソース監視

グループのリソース監視を有効にするには、グループ内で監視の必要な各リソースに対して監視を設定する必要があります。

GUIを使用してグループを作成する方法については、手順5.12「リソースグループを追加する」(83 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.9項「クラスタリソースグループの構成」(114 ページ)を参照してください。

クローン

クラスタ内の複数のノードで特定のリソースを同時に実行することができます。このためには、リソースをクローンとして設定する必要があります。クローンとして設定するリソースの1例として、STONITHや、OCFS2などのクラスタファイルシステムが挙げられます。提供されたどのリソースも、クローンとして設定できます。これは、リソースのリソースエージェントによってサポートされます。クローンリソースは、ホスティングされているノードによって異なる設定をすることもできます。

リソースクローンには次の3つのタイプがあります。

匿名クローン

最も簡単なクローンタイプです。実行場所にかかわらず、同じ動作をします。このため、マシンごとにアクティブな匿名クローンのインスタンスは1つだけ存在できます。

グローバルに固有なクローン

このリソースは独自のエントリです。1つのノードで実行しているクローンのインスタンスは、別なノードの別なインスタンスとは異なり、同じノードの2つのインスタンスが同一になることもありません。

ステートフルなクローン

このリソースのアクティブインスタンスは、アクティブとパッシブという2つの状態に分けられます。プライマリとセカンダリ、またはマスタとスレーブと呼ばれることもあります。ステートフルなクローンが、匿名またはグローバルに固有の場合もあります。「マスタ」(46 ページ)も参照してください。

クローンは、グループまたは通常リソースを1つだけ含む必要があります。

リソースの監視または制約を設定する場合、マスタには、単純なリソースとは異なる要件があります。詳細については、『*Pacemaker 1.0—Configuration Explained*』(<http://clusterlabs.org/wiki/Documentation>から入手可能)を参照してください。特に、「*Clones - Resources That Should be Active on Multiple Hosts*」のセクションを参照してください。

GUIを使用してクローンを作成する方法については、手順5.14「クローンを追加または変更する」(88 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.10項「クローンリソースの構成」(114 ページ)を参照してください。

マスタ

マスタは、インスタンスを2つのモードの1つ(masterまたはslave)で動作させることのできるクローンの特殊なタイプです。マスタには単一のグループ、または単一の正規リソースだけを含む必要があります。

リソースの監視または制約を設定する場合、マスタには、単純なリソースとは異なる要件があります。詳細については、『*Pacemaker 1.0—Configuration Explained*』(<http://clusterlabs.org/wiki/Documentation>から入手可能)を参照してください。特に、「*Multi-state - Resources That Have Multiple Modes*」のセクションを参照してください。

4.2.5 リソースオプション(メタ属性)

追加した各リソースについて、オプションを定義できます。クラスタはオプションを使用して、リソースの動作方法を決定します。CRMに特定のリソースの処理方法を通知します。リソースオプションは、`crm_resource --meta` コマンドまたはGUIを使用して設定できます(手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)参照)。

表 4.1 プリミティブリソースのオプション

オプション	説明
優先度	一部のリソースをアクティブにできない場合、クラスタは優先度の低いリソースを停止して、

オプション	説明
	優先度の高いリソースをアクティブに維持します。
target-role	クラスタが維持しようとするこのリソースの状態。使用できる値: stopped、started
is-managed	クラスタがリソースを開始して停止できるかどうか。使用できる値:true、false
resource-stickiness	リソースが現在の状態をどの程度維持したいか。デフォルトはdefault-resource-stickinessの値。
migration-threshold	ノードがこのリソースをホストできなくなるまで、このリソースについてノード上で発生する失敗の回数。デフォルト:none
multiple-active	複数のノードでアクティブなリソースを検出した場合のクラスタの動作。使用できる値:block (リソースを管理されていないとマークする)、stop_only、stop_start
failure-timeout	失敗が発生していないように動作する(リソースを失敗したノードに戻す)前に、待機する秒数デフォルト:never
allow-migrate	migrate_toまたはmigrate_fromのアクションをサポートするリソースにリソース移行を許可。

4.2.6 インスタンス属性

すべてのリソースクラスのスクリプトでは、動作方法および管理するサービスのインスタンスを指定するパラメータを提供します。リソースエージェントがパラメータをサポートする場合、それらのパラメータをcrm_resource

コマンドまたはGUIを使用して追加できます(手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)参照)。crmコマンドラインユーティリティで、インスタンス属性はparamsと呼ばれます。OCFスクリプトでサポートされているインスタンス属性のリストは、次のコマンドをrootとして実行すると参照できます。

```
crm ra info [class:[provider:]]resource_agent
```

または、次のように、コマンドを短くすることもできます。

```
crm ra info resource_agent
```

出力には、サポートされているすべての属性、それらの目的、およびデフォルト値が一覧されます。

たとえば、次のコマンドを使用します。

```
crm ra info Ipaddr
```

次の出力が返されます。

```
Manages virtual IPv4 addresses (portable version) (ocf:heartbeat:IPaddr)
```

```
This script manages IP alias IP addresses  
It can add an IP alias, or remove one.
```

```
Parameters (* denotes required, [] the default):
```

```
ip* (string): IPv4 address
```

```
The IPv4 address to be configured in dotted quad notation, for example  
"192.168.1.1".
```

```
nic (string, [eth0]): Network interface
```

```
The base network interface on which the IP address will be brought  
online.
```

```
If left empty, the script will try and determine this from the  
routing table.
```

```
Do NOT specify an alias interface in the form eth0:1 or anything here;  
rather, specify the base interface only.
```

```
cidr_netmask (string): Netmask
```

```
The netmask for the interface in CIDR format. (ie, 24), or in  
dotted quad notation 255.255.255.0).
```

```
If unspecified, the script will also try to determine this from the  
routing table.
```

```
broadcast (string): Broadcast address
```

Broadcast address associated with the IP. If left empty, the script will determine this from the netmask.

iflabel (string): Interface label
You can specify an additional label for your IP address here.

lvs_support (boolean, [false]): Enable support for LVS DR
Enable support for LVS Direct Routing configurations. In case a IP address is stopped, only move it to the loopback device to allow the local node to continue to service requests, but no longer advertise it on the network.

local_stop_script (string):
Script called when the IP is released

local_start_script (string):
Script called when the IP is added

ARP_INTERVAL_MS (integer, [500]): milliseconds between gratuitous ARPs
milliseconds between ARPs

ARP_REPEAT (integer, [10]): repeat count
How many gratuitous ARPs to send out when bringing up a new address

ARP_BACKGROUND (boolean, [yes]): run in background
run in background (no longer any reason to do this)

ARP_NETMASK (string, [ffffffffffff]): netmask for ARP
netmask for ARP - in nonstandard hexadecimal format.

Operations' defaults (advisory minimum):

start	timeout=90
stop	timeout=100
monitor_0	interval=5s timeout=20s

注記: グループ、クローン、またはマスタのインスタンス属性

グループ、クローン、およびマスタには、インスタンス属性がないので注意してください。ただし、インスタンス属性のセットは、グループ、クローン、またはマスタの子によって継承されます。

4.2.7 リソース操作

デフォルトで、クラスタはリソースが良好な状態であることを保証しません。クラスタにこれを行わせるには、リソースの定義に監視操作を追加する必要があります。監視操作は、すべてのクラスまたはリソースエージェントに追

加できます。詳細については、4.3項「リソース監視」(51 ページ)を参照してください。

表 4.2 リソース操作

説明	説明
id	アクションに指定する名前。一意にする必要があります。(IDは表示されません)
name	実行するアクション。共通の値: monitor、start、stop
間隔	操作を実行する頻度。単位: 秒
タイムアウト	アクションが失敗したと宣言する前に待機する長さ。
requires	このアクションが発生する前に満たす必要のある条件。使用できる値: nothing、quorum、fencingデフォルトは、フェンシングが有効でリソースのクラスがstonithかどうかによります。 STONITH リソースの場合、デフォルトはnothingです。
on-fail	このアクションが失敗した場合に実行するアクション。使用できる値: <ul style="list-style-type: none">• ignore: リソースが失敗しなかったのように動作します。• block: リソースにこれ以上の操作を実行しません。• stop: リソースを停止して、他の場所でも開始しません。• restart: リソースを停止して再起動します(別のノード上で)。

説明	説明
	<ul style="list-style-type: none"> • <code>fence</code>: リソースが失敗したノードを停止します (STONITH)。 • <code>stanby</code>: リソースが失敗したノードからすべてのリソースを移動させます。
対応	<code>false</code> の場合、操作は存在していない場合と同様に処理されます。使用できる値: <code>true</code> 、 <code>false</code>
役割	リソースにこの役割がある場合のみ操作を実行します。
<code>record-pending</code>	グローバルに設定したり、個々のリソースに対して設定できます。リソース上の「 <code>in-flight</code> 」操作の状態をCIBに反映させます。
<code>description</code>	操作について説明します。

4.3 リソース監視

リソースが実行中であるかどうか確認するには、そのリソースにリソースの監視を設定しておく必要があります。

リソースモニタが障害を検出すると、次の処理が行われます。

- `/etc/corosync/corosync.conf`の`logging`セクションで指定された設定に従って、ログファイルメッセージが生成されます。デフォルトでは、ログは`syslog` (通常、`/var/log/messages`)に書き込まれます。
- 障害がクラスタ管理ツール(Pacemaker GUI、HA Web Konsole `crm_mon`)と、CIBステータスセクションに反映されます。
- クラスタが明瞭な復旧アクションを開始します。これらのアクションには、リソースを停止して障害状態を修復する、ローカルまたは別のノード

でリソースを再起動するなどが含まれる場合があります。設定やクラスタの状態によっては、リソースがまったく再起動されないこともあります。

リソースの監視を設定しなかった場合、開始成功後のリソース障害は通知されず、クラスタは常にリソース状態を良好として表示してしまいます。

GUIでリソースに監視操作を追加する方法については、手順5.11「監視操作を追加または変更する」(81 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.8項「リソース監視の設定」(113 ページ)を参照してください。

4.4 リソースの制約

すべてのリソースを構成することは、ジョブのほんの一部です。クラスタが必要なすべてのリソースを認識しても、正しく処理できるとは限りません。リソースの制約を指定して、リソースを実行可能なクラスタノード、リソースのロード順序、特定のリソースが依存している他のリソースを指定することができます。

4.4.1 制約のタイプ

使用可能な制約には3種類あります。

情報の取得先

場所の制約はリソースを実行できるノード、できないノード、または実行に適したノードを定義するものです。

リソースコロケーション

コロケーションの制約は、ノード上で一緒に実行可能な、または一緒に実行することが禁止されているリソースをクラスタに伝えます。

Resource Order (リソース順序)

アクションの順序を定義する、順序の制約。

制約の設定の詳細や、オーダーおよびコロケーションの基本的な概念について詳細なバックグラウンド情報は、<http://clusterlabs.org/wiki/Documentation>に提供されている次のドキュメントを参照してください。

- 『*Pacemaker 1.0—Configuration Explained*』の「*Resource Constraints*」の章
- 『コロケーションの概要』
- 『オーダーの概要』

GUIでさまざまな種類の制約を追加する方法については、5.3.3項「リソース制約の設定」(71 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.4項「リソース制約の設定」(108 ページ)を参照してください。

4.4.2 スコアと無限大

制約を定義する際は、スコアも扱う必要があります。あらゆる種類のスコアはクラスタの動作方法と密接に関連しています。スコアの手操作によって、リソースのマイグレーションから、速度が低下したクラスタで停止するリソースの決定まで、あらゆる作業を実行できます。スコアはリソースごとに計算され、リソースに対して負のスコアが付けられているノードは、そのリソースを実行できません。リソースのスコアを計算した後、クラスタはスコアが最も高いノードを選択します。

INFINITYは現在1,000,000と定義されています。この値の増減は、次の3つの基本ルールに従います。

- 任意の値 + INFINITY = INFINITY
- 任意の値 - INFINITY = -INFINITY
- INFINITY - INFINITY = -INFINITY

リソースび制約を定義する際は、各制約のスコアを指定します。スコアはこのリソース制約に割り当てる値を示します。スコアの低い制約は、それよりもスコアが高い制約より先に適用されます。特定のリソースに対して異なるスコアで追加の場所の制約を作成することで、リソースのフェールオーバー先のノードの順序を指定できます。

4.4.3 フェールオーバー

リソースは、障害が発生すると、自動的に再起動されます。現在のノードで再起動できない場合、または現在のノードでN回失敗した場合は、別なノード

へのフェールオーバーを試みます。リソースが失敗するたびに、その失敗回数が増加します。新しいノードへのマイグレートを行う基準 (migration-threshold) となるリソースの失敗数を定義できます。クラスター内に3つ以上ノードがある場合、特定のリソースのフェールオーバー先のノードはHigh Availabilityソフトウェアが選択します。

ただし、リソースに1つ以上の場所の制約とmigration-thresholdを設定することで、そのリソースのフェールオーバー先にするノードを指定できます。GUIでこれを行う方法の詳細については、5.3.4項「リソースフェールオーバーノードの指定」(75 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.5項「リソースフェールオーバーノードの指定」(110 ページ)を参照してください。

例 4.2 移行しきい値 - プロセスフロー

たとえば、リソース「r1」の場所の制約を設定し、このリソースを「node1」で優先的に実行するように指定したと仮定します。そのノードで実行できなかった場合は、「migration-threshold」を確認して失敗回数と比較します。失敗回数 \geq マイグレーションしきい値の場合は、リソースは次の優先実行先として指定されているノードにマイグレートされます。

デフォルトでは、いったんしきい値に達すると、そのノードでは、リソースの失敗回数がリセットされるまで、失敗したリソースを実行できなくなります。これは、手動でクラスタ管理者が行うか、リソースにfailure-timeoutオプションを設定することで実行できます。

たとえば、migration-threshold=2とfailure-timeout=60sを設定すると、リソースは、2回の失敗の後に新しいノードに移行し、1分後に復帰できる可能性があります(固着性と制約のスコアによる)。

移行しきい値の概念には2つの例外があり、これらの例外は、リソースの開始失敗か、停止失敗のどちらかで発生します。

- ・ 起動の失敗では、失敗回数がINFINITYに設定されるので、常に、即時に移行が行われます。
- ・ 停止時の失敗ではフェンシングが発生します([stonith-enabled] がデフォルトである「true」に設定されている場合)。

STONITHリソースが定義されていない場合は(または
[stonith-enabled] が「false」に設定されている場合)、リソース
のマイグレーションはまったく行われません。

移行しきい値の使用と失敗回数のリセットの詳細については、5.3.4項「リソースフェールオーバーノードの指定」(75 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.5項「リソースフェールオーバーノードの指定」(110 ページ)を参照してください。

4.4.4 フェールバックノード

ノードがオンライン状態に戻り、クラスタ内にある場合は、リソースが元のノードにフェールバックすることがあります。フェールオーバー前にリソースを実行していたノードにリソースをフェールバックさせたくない場合や、リソースのフェールバック先として別のノードを指定する場合は、リソースのresource stickiness値を変更する必要があります。リソースの固着性は、リソースの作成時でも、その後も指定できます。

リソース固着性値の指定時には、次の予想される結果について考慮してください。

0の値:

デフォルトです。リソースはシステム内で最適な場所に配置されます。現在よりも「状態のよい」、または負荷の少ないノードが使用可能になると、移動することを意味しています。このオプションは自動フェールバックとほとんど同じですが、以前アクティブだったノード以外でもリソースをフェールバックできるという点が異なります。

0より大きい値:

リソースは現在の場所に留まることを望んでいます、状態がよいノードが使用可能になると移動される可能性があります。値が大きくなるほど、リソースが現在の場所に留まることを強く望んでいることを示します。

0より小さい値:

リソースは現在の場所から別な場所に移動することを望んでいます。絶対値が大きくなるほど、リソースが移動を強く望んでいることを示します。

INFINITYの値:

ノードがリソースの実行権利がなくなったために強制終了される場合(ノードのシャットダウン、ノードのスタンバイ、migration-thresholdに到達、または設定変更)以外は、リソースは常に現在の場所に留まります。このオプションは自動フェールバックを完全に無効にする場合とほとんど同じです。

-INFINITYの値:

リソースは現在の場所から常に移動されます。

4.4.5 負荷インパクトに基づくリソースの配置

すべてのリソースが同等ではありません。Xenゲストなどの一部のリソースでは、そのホストであるノードがリソースの容量要件を満たす必要があります。リソースの組み合わせられたニーズが提供された容量より大きくなるようにリソースが配置されると、リソースのパフォーマンスが低下します(あるいは失敗することさえあります)。

これを考慮に入れて、High Availability Extensionでは、次のパラメータを指定できます。

1. 一定のノードが提供する容量
2. 一定のリソースが要求する容量
3. リソースの配置に関する全体的なストラテジ

これらの設定は現在、静的であり、設定は管理者が行う必要があります。動的に検出されたり、調整されることはありません。

GUIで、これらの設定値を設定する方法については、5.3.6項「負荷インパクトに基づくリソース配置の設定」(77 ページ)を参照してください。コマンドラインを使用したい場合は、6.3.7項「負荷インパクトに基づくリソース配置の設定」(111 ページ)を参照してください。

ノードは、リソースの要件を満たすだけの空き容量があれば、そのリソースに対して資格があるとみなされます。High Availability Extensionにとって、要求または提供される容量の性質は重要ではありません。High Availability

Extensionは、リソースをノードに移動する前に、リソースのすべての容量要件が満たされているかどうか確認するだけです。

リソースの要件とノードが提供する容量を設定するには、使用属性を使用します。使用属性に任意の名前を付け、設定に必要なだけ名前/値のペアを定義します。ただし、属性値は、整数にする必要があります。

配置ストラテジは、`placement-strategy`プロパティ(グローバルクラスターオプションにある)で指定できます。次の値を使用できます。

`default` (デフォルト値)

使用値をまったく考慮しません。リソースは、場所のスコアに従って割り当てられます。スコアが同じであれば、リソースはノード間で均等に分散されます。

`utilization`

リソースの要件を満たすだけの空き容量がノードにあるかどうか決定する際に、使用値を考慮します。ただし、負荷分散は、まだ、ノードに割り当てられたリソースの数に基づいて行われます。

`minimal`

リソースの要件を満たすだけの空き容量がノードにあるかどうか決定する際に、使用値を考慮します。できるだけ少ない数のノードにリソースを集中しようとしています(残りのノードの電力節約のため)。

`balanced`

リソースの要件を満たすだけの空き容量がノードにあるかどうか決定する際に、使用値を考慮します。リソースを均等に分散して、リソースのパフォーマンスを最適化しようとしています。

注記: リソース優先度の設定

使用できる配置ストラテジは、最善策であり、まだ、複雑なヒューリスティックソルバで、常に最適な割り当て結果を得るには至っていません。したがって、最重要なリソースが最初にスケジュールされるように、リソースの優先度を設定してください。

例 4.3 負荷分散型配置の設定例

次の例は、同等のノードから成る3ノードクラスタと4つの仮想マシンを示しています。

```
node node1 utilization memory="4000"
node node2 utilization memory="4000"
node node3 utilization memory="4000"
primitive xenA ocf:heartbeat:Xen utilization memory="3500" \
    meta priority="10"
primitive xenB ocf:heartbeat:Xen utilization memory="2000" \
    meta priority="1"
primitive xenC ocf:heartbeat:Xen utilization memory="2000" \
    meta priority="1"
primitive xenD ocf:heartbeat:Xen utilization memory="1000" \
    meta priority="5"
property placement-strategy="minimal"
```

3ノードはすべてアクティブであり、まず、リソースxenAがノードに配置され、次に、xenDが配置されます。xenBとxenCは、一緒に割り当てられるか、またはどちらか1つがxenDとともに割り当てられます。

1つのノードに障害が発生した場合、残りのノード上で利用できるメモリ合計が少なすぎて、これらのリソースすべてはホストできません。xenAは確実に割り当てられ、xenDも同様です。ただし、残りのリソースxenBとxenCは、そのどちらかしか割り当てられません。xenBとxenCの優先度は同等なので、結果はまだ決められません。これを解決するためにも、どちらかに高い優先度を設定する必要があります。

4.5 詳細情報

<http://clusterlabs.org/>

High Availability Extensionに含まれているクラスタリソースマネージャであるPacemakerのホームページ。

<http://linux-ha.org>

The High Availability Linuxプロジェクトのホームページ。

<http://clusterlabs.org/wiki/Documentation>

『CRMコマンドラインインタフェース』: crmコマンドラインツールの紹介。

<http://clusterlabs.org/wiki/Documentation>

『*Pacemaker 1.0 - Configuration Explained*』 : Pacemakerの設定に使用されている概念の説明。包括的で非常に詳細な参照用情報です。

クラスタリソースの設定と管理 (GUI)

クラスタリソースを設定および管理するには、グラフィックユーザインタフェース(Pacemaker GUI)またはcrmコマンドラインユーティリティを使用します。コマンドラインを使用した方法については、第6章 クラスタリソースの設定と管理(コマンドライン) (95 ページ)を参照してください。

この章では、Pacemaker GUIを紹介し、基本的なリソースと高度なリソース(グループとクローン)の作成、制約の設定、フェールオーバーノードとフェールバックノードの指定、リソースモニタリングの設定、リソースの起動、クリーンアップ、または削除、および手動によるリソースの移行など、クラスタの設定と管理に必要な基本的なタスクについて説明します。

GUIのサポートは、2つのパッケージで提供されます。pacemaker-mgmtパッケージには、GUIのバックエンド(mgmt-dデーモン)が含まれています。このパッケージは、GUIで接続するすべてのクラスタノードにインストールする必要があります。GUIを実行するコンピュータには、pacemaker-mgmt-clientパッケージをインストールします。

5.1 Pacemaker GUI - 概要

Pacemaker GUIを開始するには、コマンドラインに「crm_gui」と入力します。設定と管理のオプションにアクセスするには、クラスタにログインする必要があります。

5.1.1 クラスタへの接続

注記: ユーザの認証

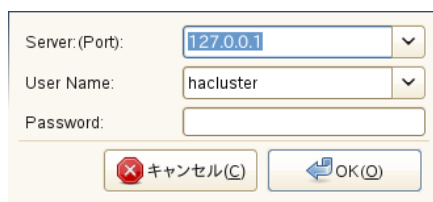
Pacemaker GUIからクラスタにログインするには、ユーザがhaclientグループのメンバである必要があります。インストール時にhaclusterという名前のLinuxユーザが作成されますが、このユーザがhaclientグループのメンバです。

Pacemaker GUIを使用する前に、haclusterユーザのパスワードを設定するか、haclientグループのメンバとして新しいユーザを作成してください。

Pacemaker GUIを使用して接続先の各ノードについてこの作業を行います。

クラスタに接続するには、[接続] > [ログイン] の順に選択します。デフォルトでは、[サーバ] フィールドにローカルホストのIPアドレスとhaclusterが[ユーザ名] として表示されています。ユーザのパスワードを入力して続行します。

 **5.1** クラスタへの接続



Server (Port): 127.0.0.1

User Name: hacluster

Password:

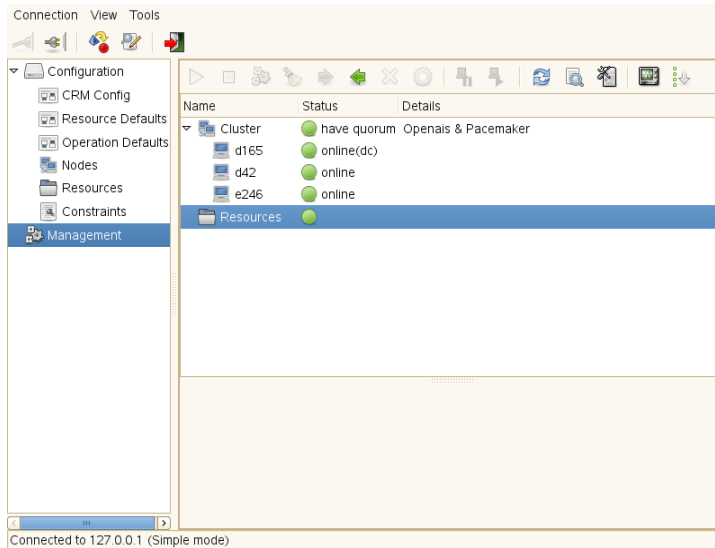
キャンセル(C) OK(O)

Pacemaker GUIをリモートに実行している場合は、クラスタノードのIPアドレスを[サーバ] として入力します。[ユーザ名] に、haclientグループに属する他の任意のユーザを使用して、クラスタに接続することもできます。

5.1.2 メインウィンドウ

接続後、メインウィンドウが開きます。

☒ 5.2 Pacemaker GUI - メインウィンドウ



CRM、リソース、ノード、制約などのクラスタコンポーネントを表示または変更するには、左のペインにある [設定] カテゴリのサブエントリを選択し、使用可能になったオプションを右のペインで使用します。さらに、Pacemaker GUIでは、サブ項目 [リソースのデフォルト]、[操作のデフォルト]、[ノード]、[リソース]、および [制約] に関して、CIBのXMLフラグメントを容易に表示、編集、インポート、およびエクスポートできます。[設定] のサブ項目のどれかを選択し、ウィンドウの上右隅で、[表示] > [XML モード] の順に選択します。

すでにリソースを設定済みの場合は、左ペインの [管理] カテゴリをクリックして、クラスタとそのリソースのステータスを表示します。この表示画面では、ノードをstandbyに設定したり、ノードの管理ステータス(現在、ノードがクラスタで管理されているかどうか)を変更することもできます。リソースの主要機能(リソースの起動、停止、クイックアップ、または移行)にアクセスするには、右のペインでリソースを選択し、ツールバーにあるアイコンを

使用します。または、リソースを右クリックして、コンテキストメニューから該当するメニュー項目を選択します。

Pacemaker GUIでは、さまざまな表示モードに切り替えて、ソフトウェアの動作を操作したり、一定の側面を表示/非表示にすることもできます。

簡易モード

ウィザードのようなモードで、リソースを追加できます。リソースの作成と変更では、サブオブジェクトに関して頻繁に使用されるタブが表示され、タブからそのタイプのオブジェクトを直接追加できます。

左ペインで [*CRM Config*] エントリを選択すると、すべての使用可能なグローバルクラスタオプションを表示し、変更できます。右ペインには、現在設定されている値が表示されます。オプションに特定の値が設定されていない場合は、デフォルト値が表示されます。

エキスパートモード

ウィザード方式またはダイアログウィンドウでリソースを追加できます。リソースの作成および変更時には、特定タイプのサブオブジェクトがすでにCIBに存在する場合は、該当するタブだけが表示されます。新しいサブオブジェクトの追加時には、オブジェクトタイプを選択するように促され、サポートされているすべてのタイプのサブオブジェクトを追加できます。

左ペインで [*CRM Config*] を選択すると、実際に設定されているグローバルクラスタオプションの値だけが表示されます。自動的にデフォルトを使用するクラスタオプションは、(値が設定されていないので)すべて非表示です。このモードでは、グローバルクラスタオプションは、個々の設定ダイアログからのみ変更できます。

ハックモード

エキスパートモードと同じ機能があります。設定をより動的にする特定のルールを含む属性セットを追加できます。たとえば、リソースに、そのホストノードによって異なるインスタンス属性を持たせることができます。さらに、メタ属性セットに時間ベースのルールを追加して、いつ属性を有効にするか決定することができます。

ウィンドウのステータスバーにも、現在アクティブなモードが表示されます。

以降のセクションでは、クラスタオプションとリソースの設定時に実行する必要がある主要タスクについて説明し、Pacemaker GUIでリソースを管理する

方法を示します。特に説明されない限り、ステップバイステップの説明は、簡易モードで実行される手順を反映しています。

5.2 グローバルクラスタオプションの設定

グローバルクラスタオプションは、一定の状況下でのクラスタの動作を制御します。それらは、セットにグループ化され、Pacemaker GUIやcrmシェルなどのクラスタ管理ツールで表示し、変更することができます。事前に定義されている値は、ほとんどの場合、そのまま保持できます。ただし、クラスタの主要機能を正しく機能させるには、クラスタの基本的なセットアップ後に、次のパラメータを調整する必要があります。

- オプションno-quorum-policy (38 ページ)
- オプションstonith-enabled (39 ページ)

手順 5.1 グローバルクラスタオプションを変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 [表示] > [簡易モード] の順に選択します。
- 3 左ペインで、[CRM Config] を選択して、グローバルクラスタオプションとそれらの現在の値を表示します。
- 4 クラスタ要件に応じて、[クォーラムなしポリシー] を適切な値に設定します。
- 5 何らかの理由でフェンシングを無効にする必要がある場合は、Stonith Enabledの選択を解除します。
- 6 [適用] をクリックして、変更を確認します。

左ペインの[CRM Config] を選択して[デフォルト] をクリックすると、すべてのオプションを、いつでもデフォルト値に戻すことができます。

5.3 クラスタリソースの設定

クラスタの管理者は、クラスタ内のサーバ上の各リソースや、サーバ上で実行する各アプリケーションに対してクラスタリソースを作成する必要があります。クラスタリソースには、Webサイト、電子メールサーバ、データベース、ファイルシステム、仮想マシン、およびユーザが常時使用できるその他のサーバベースのアプリケーションまたはサービスなどが含まれます。

作成できるリソースタイプの概要については、4.2.3項「リソースのタイプ」(42 ページ)を参照してください。

5.3.1 単純なクラスタリソースの作成

最も基本的なタイプのリソースを作成するには、次の手順に従います。

手順 5.2 プリミティブリソースを追加する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左側のペインで、[リソース] を選択し、[[追加] > [プリミティブ]] の順にクリックします。
- 3 次のダイアログで、リソースに次のようなパラメータを設定します。
 - 3a リソースに固有の[ID] を入力します。
 - 3b [クラス] リストから、そのリソースに使用するリソースエージェントクラスを選択します。[heartbeat]、[lsb]、[ocf]、または[stonith] を選択できます。詳細については、4.2.2項「サポートされるリソースエージェントクラス」(40 ページ)を参照してください。
 - 3c [ocf] をクラスとして選択した場合、OCFリソースエージェントの[プロバイダ] も指定します。OCFの指定によって、複数のベンダが同じリソースエージェントを提供できるようになります。

- 3d** [タイプ] リストから、使用するリソースエージェントを選択します(たとえば [IPaddr] または [Filesystem])。このリソースエージェントの簡単な説明を次に表示します。

[タイプ] リストに表示される選択肢は、選択した [クラス] に (OCFリソースの場合は、[プロバイダ] にも)よって異なります。

- 3e** [オプション] の下で、[Initial state of resource(リソースの当初の状態)] を設定します。

- 3f** リソースのヘルスが維持されているかどうかをクラスタに監視させる場合は、[Add monitor operation(モニタ操作の追加)] を有効にします。

Add Primitive - Basic Settings

Required

ID: my_heartbeat

Class: ocf

Provider: pacemaker

Type: Dummy

Description

Dummy resource agent.

This is a Dummy Resource Agent. It does absolutely nothing except keep track of whether its running or not.

Options

Initial state of resource: Stopped

☒ Add monitor operation

キャンセル(C) 進む(F)

- 4** [進む] をクリックします。次のウィンドウには、そのリソースに定義したパラメータの概要が表示されます。そのリソースに必要なすべての [Instance Attributes (インスタンス属性)] が一覧が表示されます。適切な値に設定するには、編集する必要があります。展開や設定によっては、属性の追加が必要な場合もあります。詳細については手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)を参照してください。

- 5 すべてのパラメータが希望どおりに設定されたら、[適用] をクリックして、そのリソースの設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに新しく追加されたリソースが表示されます。

リソースの作成中と作成後は、次のパラメータをリソースに追加したり、変更できます。

- Instance attributes - リソースが制御するサービスのインスタンスを決定します。詳細については、4.2.6項「インスタンス属性」(47 ページ)を参照してください。
- Meta attributes - 特定のリソースの処理方法をCRMに指示します。詳細については、4.2.5項「リソースオプション(メタ属性)」(46 ページ)を参照してください。
- Operations - リソースモニタリングに必要です。詳細については、4.2.7項「リソース操作」(49 ページ)を参照してください。

手順 5.3 メタ属性およびインスタンス属性を追加または変更する

- 1 Pacemaker GUIのメインウィンドウで、左側のペインの[リソース] をクリックして、そのクラスタ用に設定されているリソースを表示します。
- 2 右側のペインで、変更するリソースを選択し、[編集] をクリックします(またはリソースをダブルクリックします)。次のウィンドウには、そのリソースに定義された基本的なリソースパラメータと[メタ属性]、[インスタンス属性]、または[操作]が表示されます。

Show: List Mode

Required

ID: my_primitive

Class: ocf

Provider: heartbeat

Type: IPAddr

Optional

Description

Manages virtual IPv4 addresses.

This script manages IP alias IP addresses
It can add an IP alias, or remove one.

Meta Attributes Instance Attributes Operations

Name	Value
ip	192.168.8.212

ID: nvpair-e2f6987-795f459c-b445-7a3d7ba1924f

Name: ip

Value: 192.168.8.212

+ 追加(A) 編集(E) - 削除(R)

キャンセル(C) Reset OK(O)

- 3 新しいメタ属性またはインスタンス属性を追加するには、該当するタブを選択して [追加] をクリックします。
- 4 追加する属性の [名前] を選択します。短い [説明] が表示されます。
- 5 必要に応じて、属性の [値] を指定します。指定しなければ、その属性のデフォルト値が使用されます。
- 6 [OK] をクリックして変更を確認します。新しく追加または変更された属性がタブに表示されます。
- 7 すべてのパラメータが希望どおりに設定されたら、[OK] をクリックして、そのリソースの設定を完了します。環境設定ダイアログが閉じ、メインウィンドウに変更済みのリソースが表示されます。

ティップ: XMLソースコード- リソース用

Pacemaker GUIでは、定義したパラメータから生成されるXMLフラグメントを表示できます。個々のリソースについては、リソース設定ダイアログの右上隅で、[表示] > [XMLモード] の順に選択します。

設定したすべてのリソースのXML表現にアクセスするには、左ペインで [リソース] をクリックし、次に、メインウィンドウの右上隅で、[表示] > [XMLモード] の順に選択します。

XMLコードを表示しているエディタで、XML要素を [インポート] または [エクスポート]、あるいは手動でXMLコードを編集することができます。

5.3.2 STONITHリソースの作成

フェンシングを構成するには、複数のSTONITHリソースを構成する必要があります。

手順 5.4 STONITHリソースを追加する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左側のペインで、[リソース] を選択し、[[追加] > [プリミティブ]] の順にクリックします。
- 3 次のダイアログで、リソースに次のようなパラメータを設定します。
 - 3a リソースに固有の [ID] を入力します。
 - 3b [クラス] リストで、リソースエージェントクラスとして [stonith] を選択します。
 - 3c [タイプ] リストから、使用しているSTONITHデバイスのSTONITHプラグインを選択します。このプラグインの簡単な説明が下に表示されます。
 - 3d [オプション] の下で、[Initial state of resource (リソースの当初の状態)] を設定します。

3e クラスタにフェンシングデバイスの監視を行わせたい場合は、**[Add monitor operation(監視操作の追加)]** を起動します。詳細については、9.4項「フェンシングデバイスの監視」(139 ページ)を参照してください。

4 **[進む]** をクリックします。次のウィンドウには、そのリソースに定義したパラメータの概要が表示されます。選択した**STONITH**プラグインに必要なすべての**[インスタンス属性]**が一覧に表示されます。適切な値に設定するには、編集する必要があります。展開と設定によっては、監視操作のための属性を追加しなければならない場合もあります。詳細は手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)および5.3.7項「リソース監視の設定」(81 ページ)を参照してください。

5 すべてのパラメータが希望どおりに設定されたら、**[適用]** をクリックして、そのリソースの設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに新しく追加されたリソースが表示されます。

フェンシングを設定するには、制約の追加とクローンの使用のいずれか、またはその両方を行います。詳細については、第9章 フェンシングと**STONITH** (131 ページ)を参照してください。

5.3.3 リソース制約の設定

すべてのリソースを構成することは、ジョブのほんの一部です。クラスタが必要なすべてのリソースを認識しても、正しく処理できるとは限りません。リソースの制約を指定して、リソースを実行可能なクラスタノード、リソースのロード順序、特定のリソースが依存している他のリソースを指定することができます。

使用可能な制約タイプの概要については、4.4.1項「制約のタイプ」(52 ページ)を参照してください。制約を定義する際には、スコアも指定する必要があります。スコアとそのクラスタにおける意味の詳細については、4.4.2項「スコアと無限大」(53 ページ)を参照してください。

さまざまな制約タイプの作成方法については、次の手順を参照してください。

手順 5.5 場所の制約を追加または変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 Pacemaker GUIのメインウィンドウの左側のペインで、[制約] をクリックしてそのクラスタに設定済みの制約を表示します。
- 3 左側のペインで [制約] を選択し、[追加] をクリックします。
- 4 [Resource Location (リソース位置)] を選択し、[OK] をクリックします。
- 5 制約に固有の [ID] を入力します。既存の制約を変更する場合、ID はすでに定義されているため、環境設定ダイアログに表示されます。
- 6 制約を設定する [リソース] を選択します。リストには、そのクラスタに設定されているすべてのリソースのIDが表示されます。
- 7 制約の [Score(スコア)] を設定します。正の値は、下で指定した [ノード] でリソースを実行できることを示します。負の値は、このノードでリソースを実行できないことを示します。+/- INFINITYの値は、「can」をmustに変更します。
- 8 制約を設定する [ノード] を選択します。



- 9 [ノード] および [Score (スコア)] フィールドを空のままにしておくと、[追加] > [ルール] の順にクリックしてルールを追加することもできます。有効期間を追加するには、[追加] > [有効期間] の順にクリックします。

- 10 すべてのパラメータが希望どおり設定されたら、[OK] をクリックして、制約の設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに新しく追加または変更された制約が表示されます。

手順 5.6 コロケーションの制約を追加または変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 Pacemaker GUIのメインウィンドウの左側のペインで、[制約] をクリックしてそのクラスタに設定済みの制約を表示します。
- 3 左側のペインで [制約] を選択し、[追加] をクリックします。
- 4 [Resource Collocation (リソースコロケーション)] を選択し、[OK] をクリックします。
- 5 制約に固有の [ID] を入力します。既存の制約を変更する場合、ID はすでに定義されているため、環境設定ダイアログに表示されます。
- 6 コロケーションソースとなる [リソース] を選択します。リストには、そのクラスタに設定されているすべてのリソースのIDが表示されます。

制約が満たされないと、クラスタはリソースがまったく実行しないようにすることがあります。

- 7 [リソース] と [With Resource (対象リソース)] フィールドを両方とも空のままにしておくと、[追加] > [リソースセット] の順にクリックしてリソースを追加することもできます。有効期間を追加するには、[追加] > [有効期間] の順にクリックします。
- 8 [With Resource(対象リソース)] には、コロケーション先を定義します。クラスタはこのリソースの配置先を最初に決定し、次に [リソース] フィールドのリソースを配置する場所を決定します。
- 9 [Score(スコア)] を定義して、両方のリソース間の位置関係を決定します。正の値は、リソースを同じノードで実行しなければならないことを示します。負の値は、リソースを同じノードで実行してはならないことを示します。+/- INFINITYの値はshouldをmustに変更し

ます。スコアと他の要因との組み合わせによって、ノードの配置先が決定します。

- 10 必要に応じて、[*Resource Role*(リソース役割)] などの追加のパラメータも指定します。

選択したパラメータとオプションに応じて、短い[説明]が表示され、設定しているコロケーションの制約の効果を確認できます。

- 11 すべてのパラメータが希望どおり設定されたら、[OK] をクリックして、制約の設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに新しく追加または変更された制約が表示されます。

手順 5.7 順序の制約を追加または変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 Pacemaker GUIのメインウィンドウの左側のペインで、[制約] をクリックしてそのクラスタに設定済みの制約を表示します。
- 3 左側のペインで[制約]を選択し、[追加] をクリックします。
- 4 [*Resource Order*(リソース順序)] を選択し、[OK] をクリックします。
- 5 制約に固有の[*ID*]を入力します。既存の制約を変更する場合、IDはすでに定義されているため、環境設定ダイアログに表示されます。
- 6 [最初] では、[次] で指定するリソースの開始前に開始するリソースを定義します。
- 7 [*Then*(次に)] では、[*First*(最初)] のリソースより後に開始するリソースを定義します。

選択したパラメータとオプションに応じて、短い[説明]が表示され、設定している順序の制約の効果を確認できます。

- 8 必要な場合は、さらにパラメータを定義します。たとえば、次のように定義します。

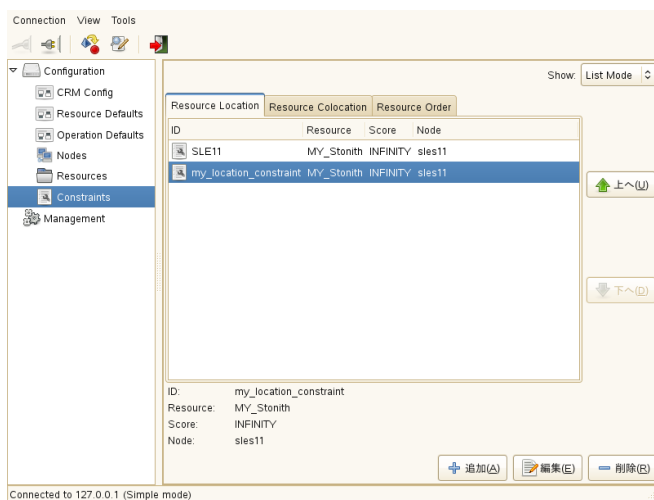
8a [スコア] でスコアを指定します。制約は、ゼロより大きい場合は必須になりますが、そうでない場合はアドバイスにすぎません。デフォルト値は、INFINITYです。

8b [シンメトリック] の値を指定します。trueを指定した場合は、リソースが逆の順序で停止されます。デフォルト値は、trueです。

9 すべてのパラメータが希望どおり設定されたら、[OK] をクリックして、制約の設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに新しく追加または変更された制約が表示されます。

Pacemaker GUIの [制約] ビューで設定したすべての制約にアクセスして変更することができます。

図 5.3 Pacemaker GUI - 制約



5.3.4 リソースフェールオーバーノードの指定

リソースに障害が発生すると、自動的に再起動されます。現在のノードで再起動できない場合、または現在のノードでN回失敗した場合は、別なノードへのフェールオーバーを試みます。新しいノードへのマイグレートを行う基準

(migration-threshold)となるリソースの失敗数を定義できます。クラスタ内に3つ以上ノードがある場合、特定のリソースのフェールオーバー先のノードはHigh Availabilityソフトウェアが選択します。

ただし、次の手順を実行すると、リソースのフェールオーバー先のノードを指定できます。

- 1 手順5.5「場所の制約を追加または変更する」(72 ページ)に記載の手順に従って、そのリソースの場所の制約を設定します。
- 2 手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)に記載の手順に従って、migration-thresholdメタ属性をそのリソースに追加し、マイグレーションしきい値の[値]を入力します。INFINITY未満の正の値を指定する必要があります。
- 3 リソースの失敗回数を自動的に失効させる場合は、手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)に記載の手順に従ってfailure-timeoutメタ属性をそのリソースに追加し、失敗タイムアウトの[値]を入力します。
- 4 リソースの優先的な実行先として、追加のフェールオーバーノードを指定する場合は、追加の場所の制約を作成します。

移行しきい値と失敗回数に関するクラスタ内のプロセスフローの例については、例4.2「移行しきい値 - プロセスフロー」(54 ページ)を参照してください。

リソースの失敗回数は、自動的に期限切れにする代わりに、いつでも、手動でクリーンアップすることもできます。詳細については、5.4.2項「リソースのクリーンアップ」(90 ページ)を参照してください。

5.3.5 リソースフェールバックノードの指定 (リソースの固着性)

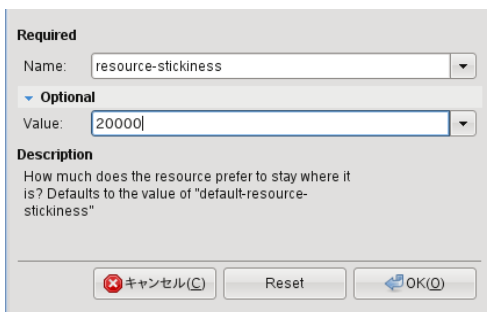
ノードがオンライン状態に戻り、クラスタ内にある場合は、リソースが元のノードにフェールバックすることがあります。フェールオーバー前にリソースを実行していたノードにリソースをフェールバックさせたくない場合や、リソースのフェールバック先として別のノードを指定する場合は、リソース

の固着性の値を変更する必要があります。リソースの固着性は、リソースの作成時でも、その後も指定できます。

さまざまなリソース固着性値の意味については、4.4.4項「フェールバックノード」(55 ページ)を参照してください。

手順 5.8 リソースの固着性を指定する

- 1 手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)に従って、resource-stickinessメタ属性をリソースに追加します。



- 2 resource-stickinessの [値] として、-INFINITYからINFINITYの範囲の値を指定します。

5.3.6 負荷インパクトに基づくリソース配置の設定

すべてのリソースが同等ではありません。Xenゲストなどの一部のリソースでは、そのホストであるノードがリソースの容量要件を満たす必要があります。リソースの組み合わせられたニーズが提供された容量より大きくなるようにリソースが配置されると、リソースのパフォーマンスが低下します(あるいは失敗することさえあります)。

これを考慮に入れて、High Availability Extensionでは、次のパラメータを指定できます。

1. 一定のノードが提供する容量

2. 一定のリソースが要求する容量
3. リソースの配置に関する全体的なストラテジ

パラメータと設定の詳細な背景情報については、4.4.5項「負荷インパクトに基づくリソースの配置」(56 ページ)を参照してください。

リソースの要件とノードが提供する容量の設定については、手順5.9「使用属性を追加または変更する」(78 ページ)の説明に従って使用属性を使用します。使用属性に任意の名前を付け、設定に必要なだけ名前/値のペアを定義します。

手順 5.9 使用属性を追加または変更する

次の例では、クラスタのノードとリソースの基本設定がすでに完了しているところへ、さらに、一定のノードが提供する容量と一定のリソースが必要とする容量の設定を行う場合を想定しています。使用属性の追加手順は、基本的に同じであり、異なるのはステップ 2 (78 ページ)とステップ 3 (78 ページ)だけです。

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 ノードが提供する容量を指定するには、次の手順に従います。
 - 2a 左ペインで、[ノード] をクリックします。
 - 2b 右ペインで、容量を設定するノードを選択し、[編集] をクリックします。
- 3 リソースが要求する容量を指定するには、次の手順に従います。
 - 3a 左ペインで、[リソース] をクリックします。
 - 3b 右ペインで、容量を設定するリソースを選択して、[編集] をクリックします。
- 4 [使用] タブを選択し、[追加] をクリックして使用属性を追加します。

- 5 新しい属性の〔名前〕を入力します。使用属性には、好きな名前を付けることができます。
- 6 属性の〔値〕を入力し、〔OK〕をクリックします。属性値は整数にする必要があります。
- 7 使用属性をさらに追加する場合は、ステップ 5 (79 ページ) からステップ 6 (79 ページ) まで繰り返します。

〔使用〕タブに、そのノードまたはリソースに定義した使用属性の要約が表示されます。
- 8 すべてのパラメータが望みどおりに設定されたら、〔OK〕をクリックして、設定ダイアログを閉じます。

図5.4「ノード容量の設定例」(79 ページ)は、8つのCPUユニットと16GBのメモリを、そのノードで実行中のリソースに提供するノードの設定を示しています。

図 5.4 ノード容量の設定例

Figure 5.4 shows a configuration dialog with two tabs: "Required" and "Optional".

Required Tab:

- Show: List Mode
- ID: bourbaki
- Uname: bourbaki
- Type: normal

Optional Tab:

The "Optional" tab has two sub-tabs: "Instance Attributes" and "Utilization". The "Instance Attributes" sub-tab is active, showing a table with the following data:

Name	Value
cpu	8
memory	16384

Below the table are buttons for "Up" and "Down".

Below the table are fields for:

- ID: nodes-bourbaki-cpu
- Name: cpu
- Value: 8

At the bottom are buttons for "Add", "Edit", "Remove", "Cancel", "Reset", and "OK".

たとえば、ノードの4096メモリ単位と4つのCPUユニットを必要とするリソースの設定例は、次のようになります。

図 5.5 リソース容量の設定例

Required

ID: xen1
Class: ocf
Provider: heartbeat
Type: Xen

Optional

Description
Manages Xen unprivileged domains (DomUs).
Resource Agent for the Xen Hypervisor.
Manages Xen virtual machine instances by managing cluster.

Utilization

Name	Value
cpu	4
memory	4096

Buttons: Add, Edit, Remove, Cancel, Reset, OK

ノードが提供する容量とリソースが要求する容量を設定した後、グローバルクラスタオプションで配置ストラテジを設定する必要があります。これを設定しないと、容量の設定は有効になりません。負荷のスケジュールに使用できるストラテジがいくつかあります。たとえば、負荷をできるだけ少ない数のノードに集中したり、使用可能なすべてのノードに均等に分散できます。詳細については、4.4.5項「負荷インパクトに基づくリソースの配置」(56ページ)を参照してください。

手順 5.10 配置ストラテジを設定する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 [表示] > [簡易モード] の順に選択します。
- 3 左ペインで、[CRMConfig] を選択して、グローバルクラスタオプションとそれらの現在の値を表示します。
- 4 要件に応じて、[配置ストラテジ] を適切な値に設定します。
- 5 何らかの理由でフェンシングを無効にする必要がある場合は、Stonith Enabledの選択を解除します。
- 6 [適用] をクリックして、変更を確認します。

5.3.7 リソース監視の設定

High Availability Extensionはノード障害を検出できますが、ノード上の個々のリソースで障害が発生した場合にも検出することができます。リソースが実行中であるかどうか確認するには、そのリソースにリソースの監視を設定しておく必要があります。リソース監視は、タイムアウト、開始遅延のいずれか、または両方と、間隔を指定することで設定できます。間隔の指定によって、CRMにリソースステータスの確認頻度を指示します。startまたはstop操作に対するTimeoutなど、特定のパラメータも設定できます。

手順 5.11 監視操作を追加または変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 Pacemaker GUIのメインウィンドウで、左側のペインの[リソース]をクリックして、そのクラスタ用に設定されているリソースを表示します。
- 3 右側のペインで、変更するリソースを選択して[編集]をクリックします。次のウィンドウには、そのリソースに定義された基本的なリソースパラメータとメタ属性、インスタンス属性、および操作が表示されます。
- 4 新しい監視操作を追加するには、該当するタブを選択して[追加]をクリックします。

既存の操作を変更するには、該当するエントリを選択して[編集]をクリックします。

- 5 [名前] で、monitor、start、[stop] など、実行するアクションを選択します。

次に示すパラメータは、ここでの選択によって決まります。

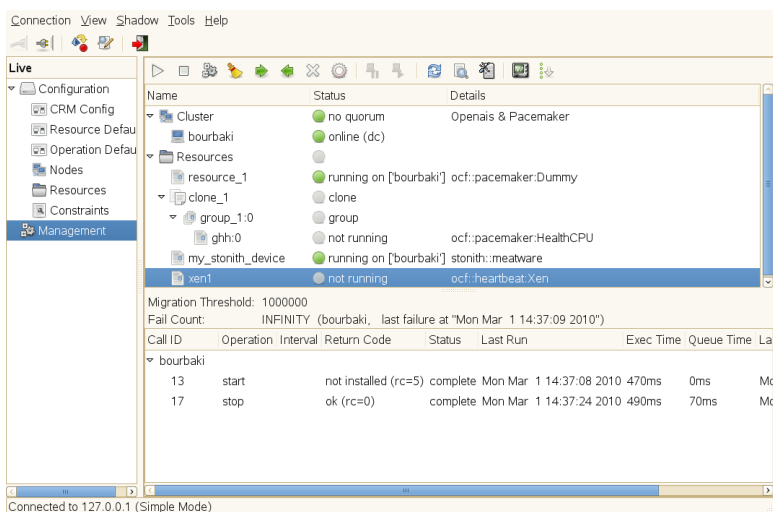
Show List Mode
 ID: my_primitive-op-monitor-5s
 Name: monitor
 Interval: 5s
 Timeout: 20s
 Optional
 追加(A) 編集(E) 削除(D)
 キャンセル(C) Reset OK(O)

- 6 [タイムアウト] フィールドに、値を秒単位で入力します。指定したタイムアウトを過ぎると、操作はfailedと見なされます。PEは何を行うか、あるいは監視操作の[OnFail(障害発生時の動作)] フィールドで指定した内容を実行するかどうかを判断します。
- 7 必要な場合は、[オプション] セクションを展開して、[失敗の場合](このアクションが失敗した場合の処理)などのパラメータを追加します。や[必要](このアクションを発声する前に満たす必要がある条件)を設定します。
- 8 すべてのパラメータが希望どおりに設定されたら、[OK] をクリックして、そのリソースの設定を完了します。構成ダイアログが閉じて、メインウィンドウに変更されたリソースが表示されます。

リソースモニタが障害を検出した場合の処理については、4.3項「リソース監視」(51 ページ)を参照してください。

Pacemaker GUIでリソースの障害を表示するには、左ペインの[管理] をクリックしてから、右ペインで、詳細を表示したいリソースを選択します。障害が発生したリソースの場合、[失敗回数] とリソースの最後の障害が右ペインの中ほど([移行しきい値] エントリの下)に表示されます。

図 5.6 リソースの失敗回数の表示



5.3.8 クラスタリソースグループの構成

クラスタリソースの中には、他のコンポーネントやリソースに依存しており、各コンポーネントや各リソースを特定の順序で開始したり、同じサーバ上で一緒に実行しなければならないものもあります。この構成を簡単にするため、グループのコンセプトをサポートしています。

リソースグループの例と、グループとそのプロパティの詳細について、「グループ」(43 ページ)を参照してください。

注記: 空のグループ

グループには1つ以上のリソースを含む必要があります。空の場合は設定は無効になります。

手順 5.12 リソースグループを追加する

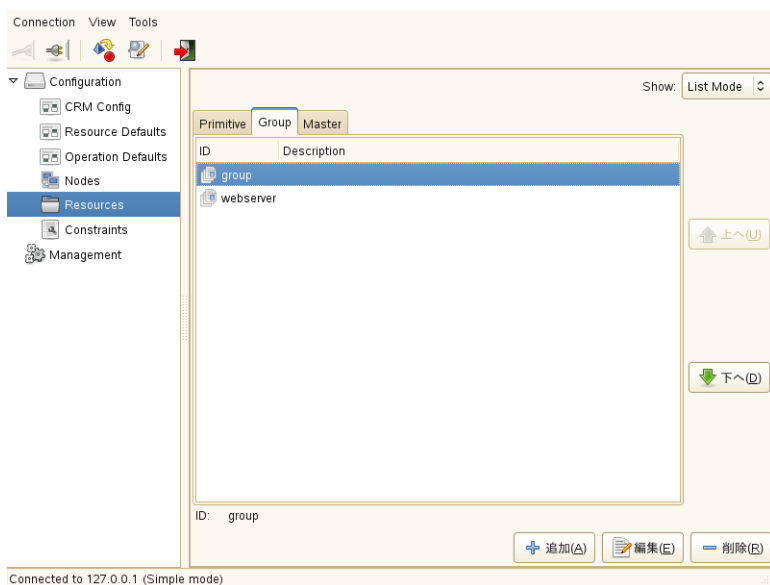
- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。

- 2 左側のペインで [リソース] を選択し、[追加] > [グループ] の順にクリックします。
- 3 グループに固有の [ID] を入力します。
- 4 [オプション] の下で、[Initial state of resource (リソースの当初の状態)] を設定し、[転送] をクリックします。
- 5 次のステップでは、グループのサブリソースとしてプリミティブを追加できます。プリミティブは手順5.2「プリミティブリソースを追加する」(66 ページ)と類似の方法で作成します。
- 6 すべてのパラメータが希望どおりに設定されたら、[適用] をクリックして、プリミティブの設定を完了します。
- 7 次のウィンドウでは、再度 [プリミティブ] を選択し、[OK] をクリックすることで、グループのサブリソースの追加を継続できます。

グループに追加するプリミティブがなくなったら、代わりに [キャンセル] をクリックします。次のウィンドウには、そのグループに定義したパラメータの概要が表示されます。グループの [Meta Attributes(メタ属性)] および [プリミティブ] の一覧が表示されます。[プリミティブ] タブのリソースの場所は、クラスタ内でのリソース開始順序を示します。

- 8 グループ内のリソースの順序は重要なので、[上へ] ボタンと [下へ] ボタンを使用して、グループ内で [プリミティブ] をソートします。
- 9 すべてのパラメータが希望どおりに設定されたら、[OK] をクリックして、そのグループの設定を完了します。環境設定ダイアログが閉じ、メインウィンドウに新しく作成された、または変更されたグループが表示されます。

図 5.7 Pacemaker GUI - グループ



手順5.12「リソースグループを追加する」(83 ページ)の説明どおりにリソースグループを作成済みであると仮定します。次の手順では、例4.1「Webサーバのリソースグループ」(43 ページ)と一致するようにグループを変更する方法を示します。

手順 5.13 既存のグループにリソースを追加する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左側のペインで [リソース] ビューに切り替え、右側のペインで、変更するグループを選択して [編集] をクリックします。次のウィンドウには、そのリソースに定義された基本的なグループパラメータとメタ属性とプリミティブが表示されます。
- 3 [プリミティブ] タブをクリックして、[追加] をクリックします。
- 4 次のダイアログで、次のパラメータを設定してIPアドレスをグループのサブリソースとして追加します。
 - 4a 一意の [ID] を入力します(たとえば、my_ipaddress)。
 - 4b [クラス] リストで、リソースエージェントクラスとして [ocf] を選択します。
 - 4c OCFリソースエージェントの [プロバイダ] として、[heartbeat] を選択します。
 - 4d [タイプ] リストで、リソースエージェントとして [IPaddr] を選択します。
 - 4e [進む] をクリックします。
 - 4f [Instance Attribute(インスタンス属性)] タブで、[IP] エントリを選択して [編集] をクリックします(または [IP] エントリをダブルクリックします)。
 - 4g [値] として、目的のIPアドレスを入力します。たとえば「192.168.1.1」と入力します。
 - 4h [OK]、[適用] の順にクリックします。グループ設定ダイアログには、新しく追加されたプリミティブが表示されます。
- 5 再度 [追加] をクリックして、次のサブリソース(ファイルシステムとWebサーバ)を追加します。

- 6 「ステップ 4a (86 ページ)」から「ステップ 4h (86 ページ)」のような手順に従って、グループの全サブリソースの設定を終了するまで、各サブリソースの該当するパラメータを設定します。



クラスタ内で開始する順序でサブリソースを設定したので、[プリミティブ] タブ内の順序はすでに正しいものになっています。

- 7 グループのリソースの順序を変更する必要がある場合は、[上へ] ボタンと [下へ] ボタンを使用して、[プリミティブ] タブのリソースをソートします。
- 8 グループからリソースを削除するには、[プリミティブ] タブのリソースを選択し、[削除] をクリックします。
- 9 [OK] をクリックしてそのグループの設定を完了します。環境設定ダイアログが閉じて、メインウィンドウに変更されたグループが表示されます。

5.3.9 クローンリソースの構成

クラスタ内の複数のノードで特定のリソースを同時に実行することができます。このためには、リソースをクローンとして設定する必要があります。クローンとして設定するリソースの1例として、STONITHや、OCFS2などのクラスタファイルシステムが挙げられます。提供されたどのリソースも、クロー

ンとして設定できます。これは、リソースのリソースエージェントによってサポートされます。クローンリソースは、ホスティングされているノードによって異なる設定をすることもできます。

使用できるリソースクローンのタイプの概要については、「クローン」(45 ページ)を参照してください。

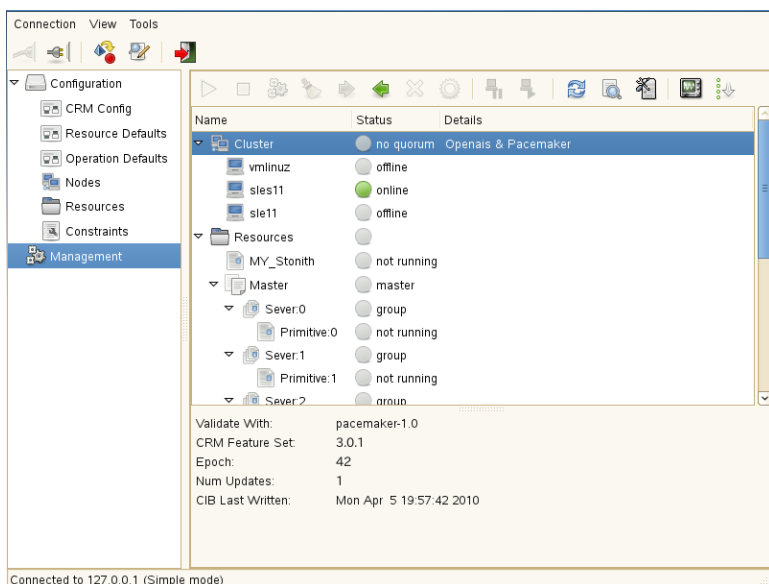
手順 5.14 クローンを追加または変更する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左側のペインで「リソース」を選択し、「追加」>「クローン」の順にクリックします。
- 3 クローンに固有の「ID」を入力します。
- 4 「オプション」の下で、「Initial state of resource (リソースの当初の状態)」を設定します。
- 5 クローンに設定するオプションを起動し、「転送」をクリックします。
- 6 次のステップでは、「プリミティブ」または「グループ」をクローンのサブリソースとして追加することができます。手順5.2「プリミティブリソースを追加する」(66 ページ)または手順5.12「リソースグループを追加する」(83 ページ)で説明している方法のように作成します。
- 7 クローン設定ダイアログ内のすべてのパラメータが希望どおりに設定されたら、「適用」をクリックして、クローンの設定を完了します。

5.4 クラスタリソースの管理

Pacemaker GUIでは、クラスタリソースの設定が可能なだけでなく、既存リソースを管理することもできます。管理ビューに切り替え、使用可能なオプションにアクセスするには、左ペインで「管理」をクリックします。

図 5.8 Pacemaker GUI - 管理



5.4.1 リソースの開始

クラスタリソースは、起動する前に、正しく設定されているようにします。たとえば、Apacheサーバをクラスタリソースとして使用する場合は、まず、Apacheサーバをセットアップし、Apacheの環境設定を完了してから、クラスタで個々のリソースを起動します。

注記: クラスタによって管理されるサービスには介入しないでください。

High Availability Extensionでリソースを管理しているとき、同じリソースを他の方法(クラスタ外で、たとえば、手動、ブート、リブートなど)で開始したり、停止してはなりません。High Availability Extensionソフトウェアが、すべてのサービスの開始または停止アクションを実行します。

ただし、サービスが適切に構成されているか確認したい場合は手動で開始しますが、High Availabilityが起動する前に停止してください。

現在クラスタで管理されているリソースへの介入については、まず、5.4.5 項「リソースの管理モードの変更」(94 ページ)に説明されているように、リソースをunmanaged modeに設定します。

Pacemaker GUIによるリソースの作成時に、リソースの初期状態をtarget-roleメタ属性で設定できます。その値をstoppedに設定した場合、リソースは、その作成後に自動的に起動しません。

手順 5.15 新しいリソースを起動

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左ペインで、[管理] をクリックします。
- 3 右ペインで、リソースを右クリックして、コンテキストメニューから[開始] を選択します(または、ツールバーの[リソースの開始] アイコンを使用します)。

5.4.2 リソースのクリーンアップ

リソースは、失敗した場合は自動的に再起動しますが、失敗のたびにリソースの失敗回数が増加します。リソースの失敗回数をPacemaker GUIで表示するには、左ペインで[管理] をクリックしてから、右ペインでリソースを選択します。リソースが失敗している場合は、その[失敗回数] が右ペインの中ほど([移行しきい値] エントリの下)に表示されます。

migration-thresholdがそのリソースに設定されている場合は、失敗の数が移行しきい値に達するとただちに、そのリソースはノードで実行できなくなります。

リソースの失敗回数は、(リソースにfailure-timeoutオプションを設定することにより)自動的にリセットするか、または次に示すように手動でリセットできます。

手順 5.16 リソースをクリーンアップする

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。

- 2 左ペインで、[管理] をクリックします。
- 3 右ペインで、該当するリソースを右クリックし、コンテキストメニューから [リソースのクリーンアップ] を選択します(またはツールバーの [リソースのクリーンアップ] アイコンを使用します)。

これによって指定したノード上の指定したリソースに対して、コマンド `crm_resource -C` および `crm_failcount -D` が実行されます。

詳細については `crm_resource(8)` (248 ページ) と `crm_failcount(8)` (239 ページ) も参照してください。

5.4.3 クラスタリソースの削除

リソースをクラスタから削除する必要がある場合は、次の手順に従って、設定エラーが発生しないようにします。

注記: 参照されているリソースの削除

任意の制約によってIDが参照されているクラスタリソースは削除できません。リソースを削除できない場合は、リソースIDが参照されている場所を確認し、最初に制約からリソースを削除します。

手順 5.17 クラスタリソースを削除する

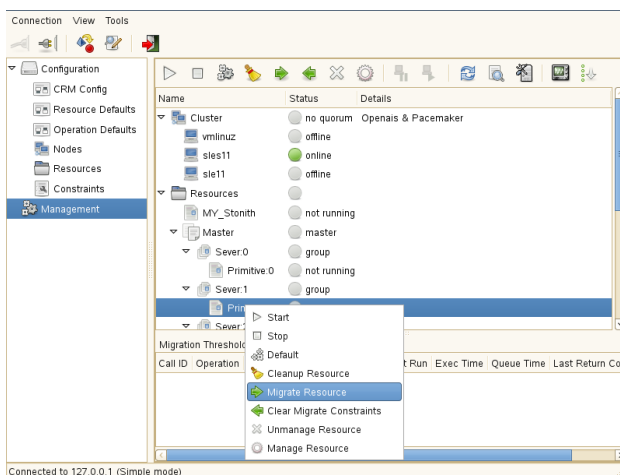
- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左ペインで、[管理] をクリックします。
- 3 右ペインで、該当するリソースを選択します。
- 4 手順5.16「リソースをクリーンアップする」(90 ページ)の説明に従って、すべてのノードでリソースをクリーンアップします。
- 5 [停止] でリソースを停止します。
- 6 リソースに関するすべての制約を削除します。これを行わないと、リソースは削除できません。

5.4.4 クラスタリソースの移行

「5.3.4項「リソースフェールオーバーノードの指定」(75 ページ)」で説明したように、ソフトウェアまたはハードウェアの障害時には、クラスタは定義可能な特定のパラメータ(たとえばマイグレーションしきい値やリソースの固着性など)に従って、リソースを自動的にフェールオーバー(マイグレート)させます。それ以外に、クラスタリソース内の別なノードにリソースを手動でマイグレートさせることもできます。

手順 5.18 リソースを手動で移行する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左ペインで、[管理] をクリックします。
- 3 右ペインで、該当するリソースを右クリックし、[リソースの移行] を選択します。



- 4 新規ウィンドウで、[ToNode(マイグレート先ノード)] で、リソースの移動先ノードを選択します。これによって移動先ノードに対して INFINITYスコアの場所の制約が作成されます。

- 5 リソースを一時的にマイグレートするには、`[Duration(期間)]` をアクティブにしてリソースが新規ノードにマイグレートされる時間を入力します。指定した時間を経過したら、リソースは元の場所に戻ることができます。あるいは、現在の場所に残ることもできます(リソースへの固着性による)。
- 6 リソースをマイグレートできない場合は(リソースの固着性と制約スコアの合計が現在のノードでINFINITYを超えている場合)、`[強制]` オプションを有効にします。これによって現在の場所に対するルールと `-INFINITY` のスコアを作成して、リソースを強制的に移動させます。

注記

これにより、`[移行制約のクリア]` で制約を解除するか、期限切れになるまで、このノードでリソースが実行されなくなります。

- 7 `[OK]` をクリックして、マイグレーションを確認します。

リソースを再び元に戻すには、次の手順に従います。

手順 5.19 移行制約をクリアする

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左ペインで、`[管理]` をクリックします。
- 3 右ペインで、該当するリソースを右クリックし、`[移行制約のクリア]` を選択します。

これによって、`crm_resource -U` コマンドが使用されます。リソースは元の場所に戻ることができます。あるいは現在の場所に残ることもできます(リソースの固着性によって)。

詳細については、`crm_resource(8)`(248 ページ)か、または<http://clusterlabs.org/wiki/Documentation>から入手できる『Pacemaker 1.0—Configuration Explained』を参照してください。特に、「Resource Migration」のセクションを参照してください。

5.4.5 リソースの管理モードの変更

リソースがクラスタで管理されているときは、他の方法で(つまり、クラスタ外で)リソースを操作しないでください。個々のリソースの保守する場合は、各リソースをunmanaged modeに設定すると、クラスタ外でそのリソースを変更できます。

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左ペインで、[管理] をクリックします。
- 3 右ペインで、該当するリソースを右クリックし、コンテキストメニューから、[リソースの管理解除] を選択します。
- 4 そのリソースの保守タスクを完了したら、右ペインで、再び該当するリソースを右クリックして、[リソースの管理] を選択します。

リソースは、この時点から、再び、High Availability Extensionによって管理されます。

クラスタリソースの設定と管理 (コマンドライン)

クラスタリソースを設定および管理するには、グラフィックユーザインタフェース(Pacemaker GUI)またはcrmコマンドラインユーティリティを使用します。GUIを使用する方法については、第5章 クラスタリソースの設定と管理(GUI) (61 ページ)を参照してください。

この章では、crmコマンドラインツールを紹介し、このツールの概要、テンプレートの使用用法、そして、主にクラスタリソースの設定と管理(基本的なリソースと高度なリソース(グループとクローン)の作成、制約の設定、フェールオーバーノードとフェールバックノードの指定、リソース監視の設定、リソースの開始、クリーンアップ、または削除、およびり手動によるリソースの移行について説明します。

6.1 crmコマンドラインツール - 概要

インストール後は、通常、crmコマンドだけが必要となります。このコマンドには、リソース、CIB、ノード、リソースエージェントなどを管理するサブコマンドがあります。crm helpを実行すると、使用できるすべてのコマンドの概要が表示されます。このコマンドには、例を組み込んだ詳細なヘルプシステムが用意されています。

crmコマンドは、次のように使用できます。

- **直接** すべてのサブコマンドをcrmに追加し、<Enter>を押すと、ただちにその出力が表示されます。たとえば、crm help raを入力すると、raサブコマンド(リソースエージェント)に関する情報を取得できます。

- **シェルスクリプトとして使用** `crm`と`crm`コマンドを含むスクリプトを使用します。これには、2つの方法があります。

```
crm -f script.cli  
crm < script.cli
```

スクリプトには、`crm`から任意のコマンドを含めることができます。例:

```
# A small example  
status  
node list
```

ハッシュ記号(#)で始まる行はコメントなので、無視されます。行が長すぎる場合は、行末にバックスラッシュ(\)を挿入し、次の行に続けます。

- **内部シェルとして対話式に使用** 「`crm`」とタイプして、内部シェルに入ります。プロンプトが`crm(live)#`に変化します。`help`を使用すると、利用可能なサブコマンドの概要を取得できます。内部シェルにはさまざまなサブコマンドレベルがあり、1つのサブコマンドをタイプして<Enter>を押すだけで、そのサブコマンドのレベルに「入る」ことができます。

たとえば、「`resource`」とタイプすると、リソース管理レベルに入ります。プロンプトは`crm(live)resource#`に変わります。内部シェルを終了したい場合は、コマンド`quit`、`bye`、または`exit`を使用します。1レベル戻る場合は、`up`、`end`、または`cd`を使用します。

「`crm`」と該当するサブコマンドをオプションなしでタイプすると、そのレベルに直接入ることができます。

内部シェルは、サブコマンドとリソースのタブによる完了もサポートします。コマンドの冒頭をタイプして<Tab>を押すと、`crm`がそのオブジェクトを完了します。

注記: 管理サブコマンドと設定サブコマンド間の相違

`crm`ツールには管理機能(サブコマンド`resources`および`node`)があり、設定に使用できます(`cib`、`configure`)。

以降のサブセクションでは、`crm`ツールの重要な側面について、その概要を示します。

6.1.1 OCFリソースエージェントに関する情報の表示

リソースエージェントはクラスタ設定で常に操作する必要があるため、`crm` ツールには、リソースエージェントの情報を取得し、リソースエージェントを管理するための`ra`コマンドが含まれています(詳細は4.2.2項「サポートされるリソースエージェントクラス」(40 ページ)参照)。

```
# crm ra
crm(live)ra#
```

コマンド`classes`は、すべてのクラスとプロバイダのリストを返します。

```
crm(live)ra# classes
heartbeat
lsb
ocf / heartbeat linbit lvm2 ocfs2 pacemaker
stonith
```

クラス(およびプロバイダ)に使用できるすべてのリソースエージェントの概要を取得するには、`list`コマンドを使用します。

```
crm(live)ra# list ocf
AoEtarget          AudibleAlarm        CTDB                 ClusterMon
Delay              Dummy               EvmsSCC              Evmsd
Filesystem         HealthCPU           HealthSMART          ICP
IPAddr            IPAddr2             IPsrcaddr            IPv6addr
LVM               LinuxSCSI            MailTo               ManageRAID
ManageVE          Pure-FTPD            Raid1                Route
SAPDatabase       SAPInstance         SendArp              ServeRAID
...
```

リソースの概要は、`info`で表示できます。

```
crm(live)ra# info ocf:drbd:linbit
This resource agent manages a DRBD resource
as a master/slave resource. DRBD is a shared-nothing replicated storage
device. (ocf:linbit:drbd)
```

Master/Slave OCF Resource Agent for DRBD

Parameters (* denotes required, [] the default):

```
drbd_resource* (string): drbd resource name
    The name of the drbd resource from the drbd.conf file.
```

```
drbdconf (string, [/etc/drbd.conf]): Path to drbd.conf
    Full path to the drbd.conf file.
```

Operations' defaults (advisory minimum):

```
start          timeout=240
promote        timeout=90
demote         timeout=90
notify         timeout=90
stop           timeout=100
monitor_Slave_0 interval=20 timeout=20 start-delay=1m
monitor_Master_0 interval=10 timeout=20 start-delay=1m
```

ビューアは、「Q」を押すと終了できます。構成の例は、付録A 単純なテストリソースのセットアップ例(369 ページ)を参照してください。

ティップ: crmの直接使用

前の例では、crmコマンドの内部シェルを使用しました。ただし、必ずしも、それを使用する必要はありません。該当するサブコマンドをcrmに追加すれば、同じ結果が得られます。たとえば、すべてのOCFリソースエージェントを一覧するには、シェルに「crmra list ocf」を入力すれば済みます。

6.1.2 テンプレートの使用

テンプレートとは、既成のクラスタ設定です。最小限の操作で、特定ユーザーのニーズに合わせて調整できます。テンプレートで設定を作成する際には、警告メッセージでヒントが与えられます。これは、後から編集することができます、さらにカスタマイズできます。

次の手順は、簡単ですが機能的なApache設定を作成する方法を示しています。

- 1 rootとしてログインします。
- 2 crmツールを起動します。

```
# crm configure
```
- 3 テンプレートから新しい設定を作成します。

3a templateサブコマンドに切り替えます。

```
crm(live)configure# template
```

3b 使用可能なテンプレートを一覧します。

```
crm(live)configure template# list templates  
gfs2-base    filesystem virtual-ip apache  clvm      ocf:2    gfs2
```

3c 必要なテンプレートを決めます。Apache設定が必要なので、apacheテンプレートを選択します。

```
crm(live)configure template# new intranet apache  
INFO: pulling in template apache  
INFO: pulling in template virtual-ip
```

4 パラメータを定義します。

4a 作成したばかりの設定を一覧します。

```
crm(live)configure template# list  
intranet
```

4b 入力が必要とする最小限の変更項目を表示します。

```
crm(live)configure template# show  
ERROR: 23: required parameter ip not set  
ERROR: 61: required parameter id not set  
ERROR: 65: required parameter configfile not set
```

4c 好みのテキストエディタを起動し、ステップ4b(99ページ)でエラーとして表示されたすべての行に入力します。

```
crm(live)configure template# edit
```

5 設定を表示し、設定が有効かどうか確認します(太字のテキストは、ステップ4c(99ページ)で入力した設定によって異なります)。

```
crm(live)configure template# show  
primitive virtual-ip ocf:heartbeat:IPaddr \  
    params ip="192.168.1.101"  
primitive apache ocf:heartbeat:apache \  
    params configfile="/etc/apache2/httpd.conf"  
monitor apache 120s:60s  
group intranet \  
    apache virtual-ip
```

6 設定を適用します。

```
crm(live)configure template# apply
crm(live)configure# cd ..
crm(live)configure# show
```

7 変更内容をCIBに送信します。

```
crm(live)configure# commit
```

詳細がわかっている場合は、コマンドをさらに簡素化できます。次のコマンドをシェルで使用して、上記の手順を要約できます。

```
crm configure template \
  new intranet apache params \
  configfile="/etc/apache2/httpd.conf" ip="192.168.1.101"
```

内部crmシェルに入っている場合は、次のコマンドを使用します。

```
crm(live)configure template# new intranet apache params \
  configfile="/etc/apache2/httpd.conf" ip="192.168.1.101"
```

ただし、このコマンドは、テンプレートから設定を作成するだけです。設定をCIBに適用したり、コミットすることはありません。

6.1.3 シャドー構成のテスト

シャドー構成は、異なる構成シナリオのテストに使用されます。複数のシャドウ設定を作成した場合は、1つ1つテストして変更を加えた影響を確認できます。

通常の処理は次のようになります。

- 1 シェルを開いてrootになります。
- 2 次のコマンドで、**crm**シェルを開始します。

```
crm configure
```

- 3 新しいシャドウ設定を作成します。

```
crm(live)configure# cib new myNewConfig
INFO: myNewConfig shadow CIB created
```


- 4 現在のライブ設定をシャドウ設定にコピーする場合は、次のコマンドを使用します。コピーしない場合は、このステップをスキップします。

```
crm(myNewConfig)# cib reset myNewConfig
```

このコマンドを使用すると、既存のリソースを後から編集する場合に、簡単に編集できます。

- 5 通常どおり変更を行います。シャドウ設定の作成後は、すべての変更がシャドウ設定に適用されます。すべての変更を保存するには、次のコマンドを使用します。

```
crm(myNewConfig)#
```

- 6 ライブクラスタ設定が再び必要な場合は、次のコマンドでライブ設定に戻ります。

```
crm(myNewConfig)configure# cib use live  
crm(live)#
```

6.1.4 構成の変更のデバッグ

設定の変更をクラスタにロードする前に、変更内容をptestでレビューすることを推奨します。ptestを指定すると、変更のコミットによって生じるアクションのダイアグラムを表示できます。ダイアグラムを表示するには、graphvizパッケージが必要です。次の例は監視操作を追加するスクリプトです。

```
# crm configure  
crm(live)configure# show fence-node2  
primitive fence-node2 stonith:apcsmart \  
    params hostlist="node2"  
crm(live)configure# monitor fence-node2 120m:60s  
crm(live)configure# show changed  
primitive fence-node2 stonith:apcsmart \  
    params hostlist="node2" \  
    op monitor interval="120m" timeout="60s"  
crm(live)configure# ptest  
crm(live)configure# commit
```

6.2 グローバルクラスタオプションの設定

グローバルクラスタオプションは、一定の状況下でのクラスタの動作を制御します。crmツールで表示し、変更できます。事前に定義されている値は、ほとんどの場合、そのまま保持できます。ただし、クラスタの主要機能を正しく機能させるには、クラスタの基本的なセットアップ後に、次のパラメータを調整する必要があります。

- オプションno-quorum-policy (38 ページ)
- オプションstonith-enabled (39 ページ)

手順 6.1 crmでグローバルクラスタオプションを変更する

- 1 シェルを開いてrootになります。
- 2 「crm configure」と入力して、内部シェルを開きます。
- 3 次のコマンドを使用して、2ノードクラスタだけのオプションを設定します。

```
crm(live)configure# property no-quorum-policy=ignore
crm(live)configure# property stonith-enabled=false
```

- 4 変更内容を表示します。

```
crm(live)configure# show
property $id="cib-bootstrap-options" \
  dc-version="1.1.1-530add2a3721a0ecccb24660a97dbfdaa3e68f51" \
  cluster-infrastructure="openais" \
  expected-quorum-votes="2" \
  no-quorum-policy="ignore" \
  stonith-enabled="false"
```

- 5 変更内容をコミットして終了します。

```
crm(live)configure# commit
crm(live)configure# exit
```

6.3 クラスタリソースの設定

クラスタの管理者は、クラスタ内のサーバ上の各リソースや、サーバ上で実行する各アプリケーションに対してクラスタリソースを作成する必要があります。クラスタリソースには、Webサイト、電子メールサーバ、データベース、ファイルシステム、仮想マシン、およびユーザが常時使用できるその他のサーバベースのアプリケーションまたはサービスなどが含まれます。

作成できるリソースタイプの概要については、4.2.3項「リソースのタイプ」(42 ページ)を参照してください。

6.3.1 クラスタリソースの作成

クラスタで使用できるRA(リソースエージェント)には3種類あります(背景情報については4.2.2項「サポートされるリソースエージェントクラス」(40 ページ)参照)。クラスタリソースを作成するには、crmツールを使用します。新しいリソースをクラスタに追加するには、次の手順に従います。

- 1 シェルを開いてrootになります。
- 2 「crm configure」と入力して、内部シェルを開きます。
- 3 プリミティブIPアドレスを設定します。

```
crm(live)configure# primitive myIP ocf:heartbeat:IPaddr \  
    params ip=127.0.0.99 op monitor interval=60s
```

前のコマンドは「プリミティブ」に名前myIPを設定します。クラス(ここではocf)、プロバイダ(heartbeat)、およびタイプ(IPaddr)を選択する必要があります。さらに、このプリミティブでは、IPアドレスなどのパラメータが必要です。自分の設定に合わせてアドレスを変更してください。

- 4 行った変更を表示して確認します。

```
crm(live)configure# show
```

- 5 変更をコミットして反映させます。

```
crm(live)configure# commit
```

6.3.2 NFSサーバの構成例

NFSサーバをセットアップするには、次の手順を完了する必要があります。

- 1 DRBDを設定します。
- 2 ファイルシステムリソースをセットアップします。
- 3 NFSサーバをセットアップし、IPアドレスを設定します。

これらを実行する方法については、次のサブセクションで学習します。

DRBDの作成

DRBD High Availabilityの設定を開始する前に、DRBDデバイスを手動でセットアップします。これは、基本的にDRBDを設定して同期させることです。正しい手順については、第13章 *Distributed Replicated Block Device (DRBD)* (171 ページ)を参照してください。ここでは、両方のクラスタノードに `/dev/drbd_r0` デバイスでアクセスできる `r0` リソースを設定したと想定します。

DRBDリソースは、OCFマスタ/スレーブリソースです。これは、DRBDリソースエージェントのメタデータの説明にあります。ただし、`promote`と`demote`のアクションが、メタデータの`actions`セクションに存在することが重要です。これらは、マスタ/スレーブリソースに必須のエントリで、通常、他のリソースにはありません。

High Availabilityの場合、マスタ/スレーブリソースはさまざまなノードに複数のマスタを持つことができます。マスタとスレーブを同じノードに持つこともできます。このため、このリソースをただ1つのマスタとスレーブを持ち、それぞれが別のノードで実行するように構成します。masterリソースの`meta`属性でこれを実行します。マスタ/スレーブリソースは、High Availabilityでのクローンリソースの特殊なタイプです。マスタとスレーブはそれぞれクローンとしてカウントされます。

次の手順に従って、DRBDリソースを設定します。

- 1 シェルを開いてrootになります。
- 2 「`crm configure`」と入力して、内部シェルを開きます。

- 3** 2ノードクラスタの場合は、msリソースごとに、次のプロパティを設定します。

```
crm(live)configure# primitive my-stonith stonith:external/ipmi ...
crm(live)configure# ms ms_drbd_r0 res_drbd_r0 meta \
    globally-unique=false ...
crm(live)configure# property no-quorum-policy=ignore
crm(live)configure# property stonith-enabled=true
```

- 4** プリミティブDRBDリソースを作成します。

```
crm(live)configure# primitive drbd_r0 ocf:linbit:drbd params \
    drbd drbd_resource=r0 op monitor interval="30s"
```

- 5** マスタ/スレーブリソースを作成します。

```
crm(live)configure# ms ms_drbd_r0 res_drbd_r0 meta master-max=1 \
    master-node-max=1 clone-max=2 clone-node-max=1 notify=true
```

- 6** コロケーションと順序の制約を指定します。

```
crm(live)configure# colocation fs_on_drbd_r0 inf: res_fs_r0
ms_drbd_r0:Master
crm(live)configure# order fs_after_drbd_r0 inf: ms_drbd_r0:promote
res_fs_r0:start
```

- 7** showコマンドで、変更内容を表示します。

- 8** commitコマンドで、変更内容をコミットします。

ファイルシステムリソースのセットアップ

filesystemリソースは、DRBDとともにOCFプリミティブリソースとして設定されます。開始および停止の要求時に、デバイスをディレクトリにマウントおよびアンマウントするタスクを実行します。この場合、デバイスは/dev/drbd0であり、マウントポイントとして使用するディレクトリは/srv/failoverです。使用されるファイルシステムはxfsです。

次のコマンドをcrmシェルで使用して、ファイルシステムリソースを設定します。

```
crm(live)# configure
crm(live)configure# primitive filesystem_resource \
```

```
ocf:linbit:drbd \  
params device=/dev/drbd_r0 directory=/srv/failover fstype=xfs
```

NFSサーバとIPアドレス

NFSサーバを常に同じIPアドレスでできるようにするため、コンピュータが通常操作に使用するアドレスに加えて追加のIPアドレスを使用します。このIPアドレスは、システムのIPアドレスに加えてアクティブなNFSサーバに割り当てられます。

NFSサーバとNFSサーバのIPアドレスは常に同じマシン上でアクティブにする必要があります。この場合、開始順序はそれ程重要ではありません。同時に開始してかまいません。これはグループリソースの代表的な要件です。

High Availability RA構成を開始する前に、NFSサーバをYaSTで構成します。システムでNFSサーバを起動させないでください。環境設定ファイルをセットアップするだけにします。この作業を手動で行うには、マニュアルページのexports(5)(man 5 exports)を参照してください。環境設定ファイルは/etc/exportsです。NFSサーバはLSBリソースとして構成されます。

IPアドレスをHigh Availability RA構成で完全に構成します。システムではこれ以外の変更は不要です。IPアドレスRAはOCF RAです。

```
crm(live)# configure  
crm(live)configure# primitive nfs_resource ocf:nfsserver \  
    params nfs_ip=10.10.0.1 nfs_shared_infodir=/shared  
crm(live)configure# primitive ip_resource ocf:heartbeat:IPaddr \  
    params ip=10.10.0.1  
crm(live)configure# group nfs_group nfs_resource ip_resource  
crm(live)configure# show  
primitive ip_res ocf:heartbeat:IPaddr \  
    params ip="192.168.1.10"  
primitive nfs_res ocf:heartbeat:nfsserver \  
    params nfs_ip="192.168.1.10" nfs_shared_infodir="/shared"  
group nfs_group nfs_res ip_res  
crm(live)configure# commit  
crm(live)configure# end  
crm(live)# quit
```

6.3.3 STONITHリソースの作成

crmからは、STONITHデバイスは単なる1つのリソースと認識されます。STONITHリソースを作成するには、次の手順に従います。

- 1 シェルを開いてrootになります。
- 2 「crm」 と入力して、内部シェルを開きます。
- 3 次のコマンドで、すべてのSTONITHタイプのリストを取得します。

```
crm(live)# ra list stonith
apcmaster                apcsmart                baytech
cyclades                 drac3                   external/drac5
external/hmchttp         external/ibmrsa         external/ibmrsa-telnet
external/ipmi            external/kdumpcheck    external/rackpdu
external/riloe           external/sbd            external/ssh
external/vmware          external/xen0           external/xen0-ha
ibmhmc                   ipmilan                 meatware
null                     nw_rpc100s              rcd_serial
rpsl0                    ssh                     suicide
```

- 4 上記のリストからSTONITHタイプを選択し、利用できるオプションのリストを表示します。次のコマンドを実行します。

```
crm(live)# ra info stonith:external/ipmi
IPMI STONITH external device (stonith:external/ipmi)
```

ipmitool based power management. Apparently, the power off method of ipmitool is intercepted by ACPI which then makes a regular shutdown. If case of a split brain on a two-node it may happen that no node survives. For two-node clusters use only the reset method.

Parameters (* denotes required, [] the default):

```
hostname (string): Hostname
    The name of the host to be managed by this STONITH device.
...
```

- 5 stonithクラス、ステップ4で選択したタイプ、および必要に応じて該当するパラメータを使用してSTONITHリソースを作成します。たとえば、次のコマンドを使用します。

```
crm(live)# configure
crm(live)configure# primitive my-stonith stonith:external/ipmi \
    params hostname="node1" \
    ipaddr="192.168.1.221" \
    userid="admin" passwd="secret" \
    op monitor interval=60m timeout=120s
```

6.3.4 リソース制約の設定

すべてのリソースを構成することは、ジョブのほんの一部です。クラスタが必要なすべてのリソースを認識しても、正しく処理できるとは限りません。たとえば、`drbd`のスレーブノードにファイルシステムをマウントしないようにしてください(実際、`drbd`では失敗します)。このような情報をクラスタが利用できるように、制約を定義します。

制約の詳細については、4.4項「リソースの制約」(52 ページ)を参照してください。

場所の制約

この種類の制約は、各リソースに複数追加できます。すべての`location`制約は、所定のリソースに関して評価されます。`fs1-loc`というIDを持つリソースを`earth`という名前のノードで実行するプリファレンスを100にする簡単な例を、次に示します。

```
crm(live)configure# location fs1-loc fs1 100: earth
```

もう1つの例は、`pingd`を使用するロケーションです。

```
crm(live)configure# primitive pingd pingd \  
    params name=pingd dampen=5s multiplier=100 host_list="r1 r2"  
crm(live)configure# location node_pref internal_www \  
    rule 50: #uname eq node1 \  
    rule pingd: defined pingd
```

コロケーションの制約

`colocation`コマンドは、同じホストまたは異なるホストでどのようなリソースを実行するか定義します。

スコアは`+inf`または`-inf`のどちらかしか設定できません。前者は、常に同じノードで実行されるリソースを定義し、後者は、同じノードで実行されてはならないリソースを定義します。無限大以外のスコアの使用も可能です。その場合は、コロケーションは、*advisory*と呼ばれ、競合がある場合、他のリソースを停止しないように、クラスタがそれらの制約に従わないようにします。

たとえば、常に同じホストにあり、IDが`filesystem_resource`と`nfs_group`である2つのリソースには、次の制約を使用します。


```
crm(live)configure# colocation nfs_on_filesystem inf: nfs_group
filesystem_resource
```

マスタスレーブ構成では、現在のノートがマスタかどうかと、リソースをローカルに実行しているかどうかを把握することが必要です。

順序の制約

リソースのアクションや操作の順序を指定することが必要な場合があります。たとえば、デバイスがシステムで利用できるようになるまで、ファイルシステムはマウントできません。順序の制約を使用して、開始、停止、マスタへの昇格など、別のリソースが特殊な条件を満たす直前または直後に、サービスを開始または停止できます。順序の制約を設定するには、次のようなコマンドをcrmシェルで使用します。

```
crm(live)configure# order nfs_after_filesystem mandatory: group_nfs
filesystem_resource
```

サンプル構成の制約

この章で使用される例は、制約を追加しないと機能しません。すべてのリソースは、必ず、マスタであるdrbdリソースと同じマシンで実行される必要があります。drbdリソースは、他のリソースが開始する前にマスタにする必要があります。マスタでないときに、drbdデバイスをマウントしようとすると失敗します。次の制約を満たす必要があります。

- ファイルシステムは、常に、DRDBリソースのマスタと同じノード上に存在する必要があります。

```
crm(live)configure# colocation filesystem_on_master inf: \
filesystem_resource drbd_resource:Master
```

- NFSサーバとIPアドレスは、ファイルシステムと同じノードに存在する必要があります。

```
crm(live)configure# colocation nfs_with_fs inf: \
nfs_group filesystem_resource
```

- NFSサーバとIPアドレスは、ファイルシステムがマウントされた後に開始されます。

```
crm(live)configure# order nfs_second mandatory: \
filesystem_resource:start nfs_group
```

- ファイルシステムは、drbdリソースがこのノードのマスタに昇格した後にマウントされる必要があります。

```
crm(live)configure# order drbd_first inf: \  
drbd_resource:promote filesystem_resource
```

6.3.5 リソースフェールオーバーノードの指定

リソースフェールオーバーを判定するには、メタ属性migration-thresholdを使用します。次に例を示します。

```
crm(live)configure# location r1-node1 r1 100: node1
```

通常、r1はノード1で実行されます。そこで失敗すると、migration-thresholdがチェックされ、失敗回数と比較されます。失敗回数がmigration-threshold以上の場合、次の候補のノードにマイグレートします。

開始が失敗すると、start-failure-is-fatalオプションによっては、失敗回数がinfに設定されます。stopの失敗により、フェンシングが発生します。STONITHが定義されていない場合には、リソースはまったく移行しません。

概要については、4.4.3項「フェールオーバー」(53 ページ)を参照してください。

6.3.6 リソースフェールバックノードの指定 (リソースの固着性)

ノードがオンライン状態に戻り、クラスタ内にある場合は、リソースが元のノードにフェールバックすることがあります。フェールオーバー前にリソースを実行していたノードにリソースをフェールバックさせたくない場合や、リソースのフェールバック先として別のノードを指定する場合は、リソースの固着性の値を変更する必要があります。リソースの固着性は、リソースの作成時でも、その後も指定できます。

概要については、4.4.4項「フェールバックノード」(55 ページ)を参照してください。

6.3.7 負荷インパクトに基づくリソース配置の設定

負荷インパクトに基づくリソース配置の設定

すべてのリソースが同等ではありません。Xenゲストなどの一部のリソースでは、そのホストであるノードがリソースの容量要件を満たす必要があります。リソースの組み合わせられたニーズが提供された容量より大きくなるようにリソースが配置されると、リソースのパフォーマンスが低下します(あるいは失敗することさえあります)。

これを考慮に入れて、High Availability Extensionでは、次のパラメータを指定できます。

1. 一定のノードが提供する容量
2. 一定のリソースが要求する容量
3. リソースの配置に関する全体的なストラテジ

パラメータと設定の詳細な背景情報については、4.4.5項「負荷インパクトに基づくリソースの配置」(56 ページ)を参照してください。

リソースの要件とノードが提供する容量の設定については、手順5.9「使用属性を追加または変更する」(78 ページ)の説明に従って使用属性を使用します。使用属性に任意の名前を付け、設定に必要なだけ名前/値のペアを定義します。

次の例では、クラスタのノードとリソースの基本設定がすでに完了しているところへ、さらに、一定のノードが提供する容量と一定のリソースが必要とする容量の設定を行う場合を想定しています。

手順 6.2 *crm*で使用属性を追加または変更する

- 1 次のコマンドで、*crm*シェルを開始します。

```
crm configure
```

- 2 ノードが提供する容量を指定するには、次のコマンドを使用し、プレースホルダ*NODE_1*をノードの名前に置き換えます。

```
crm(live)configure# node NODE_1 utilization memory=16384 cpu=8
```

これらの値によって、`NODE_1`は16GBのメモリと8つのCPUコアをリソースに提供すると想定されます。

- 3** リソースが要求する容量を指定するには、次のコマンドを使用します。

```
crm(live)configure# primitive xen1 ocf:heartbeat:Xen ... \  
    utilization memory=4096 cpu=4
```

これによって、リソースはnodeAからの4096のメモリ単位と4つのcpuユニットを使用します。

- 4** `property`コマンドを使用して、配置ストラテジを設定します。

```
crm(live)configure# property ...
```

配置ストラテジには、4つの値を使用できます。

```
propertyplacement-strategy=default
```

デフォルトによって、使用値はまったく考慮されません。リソースは、場所のスコアに従って割り当てられます。スコアが同じであれば、リソースはノード間で均等に分散されます。

```
propertyplacement-strategy=utilization
```

ノードにリソースの要件を満たすだけの空き容量がある場合は、ノードに資格があるかどうか決定する際に使用値を考慮に入れます。ただし、負荷分散は、まだ、ノードに割り当てられたリソースの数に基づいて行われます。

```
propertyplacement-strategy=minimal
```

ノードにリソースを提供する資格があるかどうか決定する際に使用値を考慮に入れます。できるだけ少ない数のノードにリソースを集中することにより、残りのノードで電力を節約できるようにします。

```
propertyplacement-strategy=balanced
```

ノードにリソースを提供する資格があるかどうか決定する際に使用値を考慮に入れます。リソースを均等に分散することにより、リソースのパフォーマンスの最適化を図ります。

配置ストラテジは、最善策であり、まだ、複雑なヒューリスティックソルバで常に最適な割り当て結果を得るには至っていません。リ

ソースの優先度を正しく設定して、最重要なリソースが最初にスケジュールされるようにしてください。

5 変更をコミットしてから、crmシェルを終了します。

```
crm(live)configure# commit
```

次の例は、同等のノードから成る3ノードクラスタと4つの仮想マシンを示しています。

```
crm(live)configure# node node1 utilization memory="4000"
crm(live)configure# node node2 utilization memory="4000"
crm(live)configure# node node3 utilization memory="4000"
crm(live)configure# primitive xenA ocf:heartbeat:Xen \
    utilization memory="3500" meta priority="10"
crm(live)configure# primitive xenB ocf:heartbeat:Xen \
    utilization memory="2000" meta priority="1"
crm(live)configure# primitive xenC ocf:heartbeat:Xen \
    utilization memory="2000" meta priority="1"
crm(live)configure# primitive xenD ocf:heartbeat:Xen \
    utilization memory="1000" meta priority="5"
crm(live)configure# property placement-strategy="minimal"
```

3ノードはすべてアクティブであり、まず、**xenA**がノードに配置され、次に、**xenD**が配置されます。**xenB**と**xenC**は、一緒に割り当てられるか、またはどちらか1つが**xenD**とともに割り当てられます。

1つのノードに障害が発生した場合、残りのノード上で利用できるメモリ合計が少なすぎて、これらのリソースすべてはホストできません。**xenA**は確実に割り当てられ、**xenD**も同様です。ただし、**xenB**と**xenC**は、そのどちらか1つしか割り当てられません。**xenB**と**xenC**の優先度は同等なので、結果はまだ未定義です。これを解決するためにも、どちらかに高い優先度を設定する必要があります。

6.3.8 リソース監視の設定

リソースを監視するには、2つの方法(**op**キーワードで監視処理を定義するか、**monitor**コマンドを使用するか)があります。次の例では、**Apache**リソースを設定し、**op**キーワードの使用で30分ごとに監視します。

```
crm(live)configure# primitive apache apache \
    params ... \
    op monitor interval=60s timeout=30s
```

同じことを次のようにしても実行できます。

```
crm(live)configure# primitive apache apache \  
    params ...  
crm(live)configure# monitor apache 60s:30s
```

概要については、4.3項「リソース監視」(51 ページ)を参照してください。

6.3.9 クラスタリソースグループの構成

クラスタの最も一般的な要素の1つは、一緒の場所で見つける必要のあるリソースのセットです。連続的に開始し、逆の順序で停止します。この構成を簡単にするため、グループのコンセプトをサポートしています。次の例では、2つのプリミティブ(IPアドレスと電子メールリソース)を作成します。

- 1 crmコマンドをシステム管理者として実行します。プロンプトが `crm(live)` に変化します。

- 2 プリミティブを構成します。

```
crm(live)# configure  
crm(live)configure# primitive Public-IP ocf:IPaddr:heartbeat \  
    params ip=1.2.3.4  
crm(live)configure# primitive Email lsb:exim
```

- 3 該当するIDを使用して、正しい順序で、プリミティブをグループ化します。

```
crm(live)configure# group shortcut Public-IP Email
```

概要については、「グループ」(43 ページ)を参照してください。

6.3.10 クローンリソースの構成

クローンは当初、IPアドレスのN個のインスタンスを開始し、負荷分散のためにクラスタ上に分散させる便利な方法と考えられていました。それらは、DLM との統合、サブシステムおよびOCFS2のフェンシングなど、他にも多くの目的に非常に有効であることがわかってきました。どのようなリソースでも、リソースエージェントがサポートしていれば、クローン化できます。

クローンリソースの詳細については、「クローン」 (45 ページ)を参照してください。

匿名クローンリソースの作成

匿名クローンリソースを作成するには、まずプリミティブリソースを作成して、それをcloneコマンドで指定することです。次の操作を実行します。

- 1 シェルを開いてrootになります。
- 2 「crm configure」と入力して、内部シェルを開きます。
- 3 次のように、プリミティブを構成します。

```
crm(live)configure# primitive Apache lsb:apache
```

- 4 プリミティブをクローンします。

```
crm(live)configure# clone apache-clone Apache
```

ステートフル/マルチステートクローンリソースの作成

ステートフルクローンリソースを作成するには、まずプリミティブリソースを作成してから、マスタ/スレーブリソースを作成します。

- 1 シェルを開いてrootになります。
- 2 「crm configure」と入力して、内部シェルを開きます。
- 3 プリミティブを作成します。必要に応じて間隔を変更します。

```
crm(live)configure# primitive myRSC ocf:myCorp:myAppl \  
    op monitor interval=60 \  
    op monitor interval=61 role=Master
```

- 4 マスタスレーブリソースを作成します。

```
crm(live)configure# clone apache-clone Apache
```

6.4 クラスタリソースの管理

crmツールでは、クラスタリソースの設定が可能だけでなく、既存リソースを管理することもできます。移行のサブセクションで概要を示します。

6.4.1 新しいクラスタリソースの開始

新しいクラスタリソースを開始するには、そのIDが必要です。次の手順に従います。

1 シェルを開いて、rootになります。

2 「crm」と入力して、内部シェルを開きます。

3 リソースレベルに切り替えます。

```
crm(live)# resource
```

4 startでリソースを開始し、<Tab>キーを押してすべての既知のリソースを表示します。

```
crm(live)resource# start start ID
```

6.4.2 リソースのクリーンアップ

リソースは、失敗した場合は自動的に再起動しますが、失敗のたびにリソースの失敗回数が増加します。migration-thresholdがそのリソースに設定されている場合は、失敗の数が移行しきい値に達するとただちに、そのリソースはノードで実行できなくなります。

1 シェルを開いて、rootユーザとしてログインします。

2 すべてのリソースのリストを取得します。

```
crm resource list
...
Resource Group: dlm-clvm:1
    dlm:1 (ocf::pacemaker:controld) Started
    clvm:1 (ocf::lvm2:clvmd) Started
    cmirrord:1 (ocf::lvm2:cmirrord) Started
```


- 3 リソースが実行中の場合は、まず、そのリソースを停止する必要があります。RSCをリソースの名前で置き換えてください。

```
crm resource stop RSC
```

たとえば、dlm-clvmリソースグループからのDLMリソースを停止したい場合は、RSCをdlmで置き換えます。

- 4 リソース自体を削除します。

```
crm configure delete ID
```

6.4.3 クラスタリソースを削除する

クラスタリソースを削除するには、その該当するIDが必要です。次の手順に従います。

- 1 シェルを開いてrootになります。
- 2 次のコマンドを実行して、リソースのリストを取得します。

```
crm(live)# resource status
```

たとえば、出力はこのようなになります(ここで、myIPはリソースの該当するID)。

```
myIP      (ocf::IPaddr:heartbeat) ...
```

- 3 該当するIDを持つリソースを削除します(これは、commitも含意します)。

```
crm(live)# configure delete YOUR_ID
```

- 4 変更をコミットします。

```
crm(live)# configure commit
```

6.4.4 クラスタリソースのマイグレーション

リソースは、ハードウェアまたはソフトウェアに障害が発生した場合、クラスタ内の他のノードに自動的にフェールオーバー(つまり移行)するよう設定さ

れていますが、**Pacemaker GUI**またはコマンドラインを使用して、手動でリソースをクラスタ内の別のノードに移行することもできます。

- 1** シェルを開いて、rootになります。
- 2** 「crm」と入力して、内部シェルを開きます。
- 3** ipaddress1という名前のリソースをnode2という名前にクラスタノードにマイグレートするには、次のように入力します。

```
crm(live)# resource
crm(live)resource# migrate ipaddress1 node2
```

Webインターフェイスによる クラスタリソースの管理

crmコマンドラインツールとPacemaker GUIに加えて、High Availability Extensionには、HA Web Konsoleという管理タスク用のWebベースのユーザインターフェイスが含まれています。このインターフェイスを使用すると、Linux以外のコンピュータからも、Linuxクラスタを監視および管理できます。さらに、このインターフェイスは、システムにグラフィックユーザインターフェイスがなかったり、使用できない場合の理想的なソリューションになります。

このWebインターフェイスは、hawkパッケージに含まれています。これは、HA Web Konsoleで接続したいすべてのクラスタノードにインストールする必要があります。HA Web Konsoleの使用によってクラスタノードにアクセスするコンピュータには、JavaScriptとクッキーを有効にして接続を確立できるようにした(グラフィック)Webブラウザが必要です。

注記: ユーザの認証

HA Web Konsoleからクラスタにログインするには、そのユーザがhaclientグループのメンバである必要があります。インストール時にhaclusterという名前のLinuxユーザが作成されますが、このユーザがhaclientグループのメンバです。

HA Web Konsoleを使用する前に、haclusterユーザのパスワードを設定するか、haclientグループのメンバとして新しいユーザを作成してください。

HA Web Konsoleを使用して接続する各ノードでこれを実行します。

7.1 HA Web Konsoleの起動とログイン

手順 7.1 HA Web Konsoleを起動する

HA Web Konsoleを使用するには、このWebインターフェイスで接続したいノードで、該当のWebサービスが開始されている必要があります。通信については、標準のHTTP(s)プロトコルとポート7630を使用します。

- 1 接続先にするノードで、シェルを開きrootとしてログインします。
- 2 次のように入力して、サービスのステータスをチェックします。

```
rchawk status
```

- 3 サービスが実行されていない場合は、次のコマンドでサービスを開始します。

```
rchawk start
```

ブート時にHA Web Konsoleを自動的に起動したい場合は、次のコマンドを実行します。

```
chkconfig hawk on
```

- 4 任意のコンピュータで、Webブラウザを起動し、JavaScriptとクッキーが有効なことを確認します。
- 5 Webブラウザを、任意のクラスタノードのIPアドレスまたはホスト名、あるいは設定した任意のIPAddr (2) リソースのアドレスにポイントします。

```
https://IPaddress:7630/main/status
```

注記: 証明書の警告

ブラウザとブラウザオプションによっては、初めてURLにアクセスしたときに、証明書の警告が表示される場合があります。これは、HA Web Konsoleがデフォルトでは信頼できるとみなされていない自己署名の証明書を使用するからです。

続行するには、ブラウザに例外を追加して警告をバイパスします。最初から警告を回避するには、自己署名の証明書を、公式の認証局に

よって署名された証明書で置き換えることもできます。方法の詳細については、自己署名証明書の置き換え (123 ページ)を参照してください。

- 6 HA Web Konsoleログイン画面で、haclusterユーザ(または、haclientグループのメンバである他の任意のユーザ)の [ユーザ名] と [パスワード] を入力し、 [ログイン] をクリックします。

[クラスタステータス] 画面が開き、クラスタノードとリソースのステータスを表示します。これは、crm_monの出力とよく似ています。

7.2 HA Web Konsoleの使用

ログイン後、HA Web Konsoleによって、最も重要なグローバルクラスタパラメータと、クラスタノードおよびリソースのステータスが表示されます。ステータスの表示には、次のカラーコードが使用されます。

- 緑: OK。たとえば、リソースが実行中か、ノードがオンラインです。
- 赤: 不良。クリーンでないたとえば、リソースに障害が発生したか、ノードがクリーンにシャットダウンされませんでした。
- 黄: 過渡期。たとえば、現在、ノードのシャットダウン中です。
- 灰色: 実行中ではありません。ただし、クラスタは、それが実行中であると予測します。たとえば、管理者が停止したか、standbyモードにしたノードです。オフラインのノードも、灰色で表示されます(クリーンにシャットダウンされた場合)。

図 7.1 HA Web Konsole - クラスタステータス

Cluster Status User: hacluster [Log Out](#)

Failed op: node hex-14 resource ctdb:1: call-id=46 operation=monitor rc-code=7

2 nodes configured

- hex-13: online
- hex-14: online

7 resources configured

- Clone Set: c-ocfs2-3
- Clone Set: ctdb-clone
 - ctdb:0: Started: hex-13
 - ctdb:1: Stopped
- Clone Set: dlm-clone
- Clone Set: o2cb-clone
- fencing-sbd: Started: hex-13
- Group: ga
- Clone Set: cg

Stack: openais
Version: 1.1.0-46679a8fec7
Current DC: hex-13
Stickiness: 1
STONITH: Disabled
Cluster is: Symmetric
No Quorum: stop

Copyright © 2008-2010 Novell, Inc. Host: hex-13

〔ノード〕グループと〔リソース〕グループの矢印記号をクリックすると、ツリービューが展開したり、閉じたりします。

リソースに障害が発生した場合は、詳細を示す障害メッセージが画面上部に赤色で表示されます。

ノードまたはリソースの右にあるレンチ型のアイコンをクリックすると、コンテキストメニューにアクセスできます。このメニューでは、リソースを開始、停止、またはクリーンアップしたり、ノードをonlineまたはstandbyモードにしたり、ノードをフェンシングしたりできます。

現在、HA Web Konsoleでは、基本的なオペレータタスクしかできませんが、将来は、リソースやノードの設定など、さらに機能が追加されます。

7.3 トラブルシューティング

HA Web Konsoleログファイル

HA Web Konsoleログファイルは、`/srv/www/hawk/log`にあります。なんらかの理由でHA Web Konsoleに全くアクセスできない場合は、それらのファイルをチェックすると有益です。

HA Web Konsoleでリソースの開始や停止に問題がある場合は、Pacemakerによって記録されたログファイル(デフォルトでは/var/log/messagesにある)をチェックしてください。

認証ファイル

haclientグループに追加した新規ユーザでHA Web Konsoleにログインできない(または、HA Web Konsoleがこのユーザからのログインを受け入れるまでに遅延がある)場合は、rcnscdデーモンを、rcnscd stopで停止して、再試行してください。

自己署名証明書の置き換え

HA Web Konsoleの最初の起動で自己署名証明書に関する警告が発行されるのを避けるには、自動生成された証明書を、独自の証明書または公式認証局(CA)によって署名された証明書で置き換えてください。

証明書は/etc/lighttpd/certs/hawk-combined.pemに保存され、キーと証明書の両方を含んでいます。新しいキーと証明書を作成した後、または受け取った後に、次のコマンドを実行してそれらを組み合わせます。

```
cat keyfile certificationfile > /etc/lighttpd/certs/hawk-combined.pem
```

パーミッションを変更して、rootだけがファイルをアクセスできるようにします。

```
chown root.root /etc/lighttpd/certs/hawk-combined.pem  
chmod 600 /etc/lighttpd/certs/hawk-combined.pem
```


リソースエージェントの追加または変更

クラスタによる管理が必要なすべてのタスクは、リソースとして使用できなければなりません。主要なグループとして、リソースエージェントとSTONITHエージェントの2つがあります。両方のカテゴリで、エージェントの追加や所有が可能で、クラスタ機能を各自のニーズに合わせて拡張することができます。

8.1 STONITHエージェント

クラスタがノードの1つの誤動作を検出し、そのノードの削除が必要となることがあります。これをフェンシングと呼び、一般にSTONITHリソースで実行されます。すべてのSTONITHリソースは各ノードの`/usr/lib/stonith/plugins`にあります。

警告: SSHおよびSTONITHはサポートされていません。

SSHが他のシステムの問題にどのように反応するかを知る方法はありません。このため、SSHとSTONITHエージェントは本番環境ではサポートされていません。

現在使用可能なすべてのSTONITHデバイス(ソフトウェア側から)のリストを入手するには、コマンド`stonith -L`を使用します。

今のところ、STONITHエージェントの作成に関するドキュメントはありません。新しいSTONITHエージェントを作成する場合は、`heartbeat-common`パッケージのソースに提供されている例を参照してください。

8.2 OCFリソースエージェントの作成

`/usr/lib/ocf/resource.d/`で提供されているすべてのOCFリソースエージェントの詳細については、4.2.2項「サポートされるリソースエージェントクラス」(40 ページ)を参照してください。各リソースエージェントは、それを制御する次の操作をサポートしている必要があります。

`start`

リソースを開始または有効化します。

`stop`

リソースを中止または無効化します。

`status`

リソースのステータスを返します。

`monitor`

`status`と同様ですが、予期しない状態もチェックします。

`validate`

リソースの設定を検証します。

`meta-data`

リソースエージェントの情報をXMLで返します。

OCF RAを作成する一般的な手順は、次のとおりです。

- 1 テンプレートとして、`/usr/lib/ocf/resource.d/pacemaker/Dummy` ファイルをロードします。
- 2 新しいリソースエージェントごとに新しいサブディレクトリを作成して、名前が競合しないようにします。たとえばリソースグループ `kitchen` にリソース `coffee_machine` がある場合、このリソースを `/usr/lib/ocf/resource.d/kitchen/` ディレクトリに追加します。このRAにアクセスするには、コマンド `crm` を実行します。

```
configure
```

```
primitive coffee_1 ocf:coffee_machine:kitchen ...
```

3 異なるシェル関数を実装し、異なる名前でファイルを保存します。

OCFリソースエージェントの作成についての詳細は、http://linux-ha.org/wiki/Resource_Agentsを参照してください。コンセプトの特別な情報については、第1章 製品の概要(3 ページ)を参照してください。

8.3 OCF戻りコードと障害回復

OCF仕様によると、アクションが返す必要がある出口コードの厳密な定義があります。クラスタは常に、予期される結果に対する戻りコードを確認します。結果が予期された値と一致しない場合、アクションは失敗したとみなされ、回復処理が開始されます。障害回復には3種類あります。

表 8.1 障害回復の種類

回復の種類	説明	クラスタが行うアクション
soft	一時的なエラーが発生しました。	リソースを再起動するか、新しい場所に移動させます。
hard	一時的ではないエラーが発生しました。エラーは、現在のノードに固有の場合があります。	リソースを他の場所に移動して、現在のノードで再試行されないようにします。
致命的	すべてのクラスタノードに共通の、一時的ではないエラーが発生しました。これは、不正な設定が指定されたことを示しています。	リソースを停止して、どのクラスタノードでも開始されないようにします。

アクションが失敗したと仮定して、次の表では、エラーコードを受け取った場合の異なるOCF戻りコードとクラスタが開始する回復の種類を概説します。

表 8.2 OCF戻りコード

OCF 戻り コード	OCFエイリアス	説明	回復 の種 類
0	OCF_SUCCESS	成功。コマンドは正常に完了しました。これは、すべての start、stop、promote、demote コマンドの予期された結果です。	soft
1	OCF_ERR_GENERIC	汎用の「問題が発生した」ことを示すエラーコード。	soft
2	OCF_ERR_ARGS	リソースの構成がこのマシンで有効ではありません(たとえば、ノード上に見つからない場所/ツールを参照している場合)。	hard
3	OCF_ERR_UNIMPLEMENTED	要求されたアクションは実行されていません。	hard
4	OCF_ERR_PERM	リソースエージェントに、タスクを完了できるだけの権限がありません。	hard
5	OCF_ERR_INSTALLED	リソースが必要とするツールがこのコンピュータにインストールされていません。	hard
6	OCF_ERR_CONFIGURED	リソースの設定が無効です(たとえば、必要なパラメータがないなど)。	致命的
7	OCF_NOT_RUNNING	リソースが実行されていません。クラスタは、どのアクション	該当なし

OCF 戻り コード	OCFエイリアス	説明	回復 の種 類
		<p>ンについてもこれを返すリソースを停止しようとしません。</p> <p>このOCF戻りコードはリソース回復を必要することも必要としないこともあります。予期されたリソースの状態に依存します。予期されない場合は、soft回復を行います。</p>	
8	OCF_RUNNING_MASTER	リソースはマスタモードで実行しています。	soft
9	OCF_FAILED_MASTER	リソースはマスタモードですが、失敗しました。リソースは降格、停止され、再度開始されます(昇格されます)。	soft
その他	該当なし	カスタムエラーコード。	soft

フェンシングとSTONITH

フェンシングはHA(High Availability)向けコンピュータクラスタにおいて、非常に重要なコンセプトです。クラスタがノードの1つの誤動作を検出し、そのノードの削除が必要となることがあります。これをフェンシングと呼び、一般にSTONITHリソースで実行されます。フェンシングは、HAクラスタを既知の状態にするための方法として定義できます。

クラスタのすべてのリソースには、それぞれ、状態が関連付けられています(たとえば、「リソースr1はnode1で開始されている」など)。HAクラスタでは、このような状態は「リソースr1はnode1以外のすべてのノードで停止している」ことを示します。HAクラスタは各リソースが1つのノードでのみ起動されるようにするためです。各ノードはリソースに生じた変更を報告する必要があります。つまり、クラスタの状態は、リソースの状態とノードの状態の集まりです。

どのような理由であれ、一部のノードまたはリソースの状態を正確に確立できない場合、フェンシングが行われます。クラスタが所定のノードで起きていることを認識しない場合でも、フェンシングによって、そのノードが重要なリソースを実行しないようにできます。

9.1 フェンシングのクラス

フェンシングには、リソースレベルとノードレベルのフェンシングという、2つのクラスがあります。後者について、この章で主に説明します。

リソースレベルのフェンシング

リソースレベルのフェンシングでは、クラスタが、ノードによる1つ以上のリソースのアクセスを不可能にできます。代表的な一例はSANで、フェンシング操作によってSANスイッチのルールを変更し、ノードからのアクセスを拒否します。

リソースレベルのフェンシングは、保護対象のリソースが依存している通常のリソースを使用して実行できます。このようなリソースは、このノードでの起動を拒否するため、それに依存するリソースは同じノード上で実行されません。

ノードレベルのフェンシング

ノードレベルのフェンシングでは、ノードがどのリソースも実行しなくなります。これは、通常、乱暴ですが非常にシンプルな方法(電源スイッチでノードをリセットする)で実行されます。これは、ノードが応答しなくなった場合に必要です。

9.2 ノードレベルのフェンシング

SUSE® Linux Enterprise High Availability Extensionでは、フェンシングの実装はSTONITH(Shoot The Other Node in the Head)です。これにより、ノードレベルのフェンシングが実行されます。High Availability Extensionにはstonithコマンドラインツールが付属し、これはクラスタ上のノードの電源をリモートでオフにする拡張インタフェースです。使用できるオプションの概要については、stonith --helpを実行するか、またはstonithのマニュアルページで詳細を参照してください。

9.2.1 STONITHデバイス

ノードレベルのフェンシングを使用するには、まず、フェンシングデバイスを用意する必要があります。High Availability ExtensionでサポートされているSTONITHデバイスのリストを取得するには、次のコマンドをrootとして任意のノード上で実行します。

```
stonith -L
```

STONITHデバイスは次のカテゴリに分類できます。

電源分配装置(PDU)

電源分配装置は、重要なネットワーク、サーバ、データセンター装置の電力と機能を管理する、重要な要素です。接続した装置のリモートロード監視と、個々のコンセントでリモート電源オン/オフのための電力制御を実行できます。

無停電電源装置(UPS)

電力会社からの停電時に別個のソースから電力を供給することで、安定した電源から接続先の装置に緊急電力が提供されます。

ブレード電源制御デバイス

クラスタを一連のブレード上で実行している場合、ブレードエンクロージャの電源制御デバイスがフェンシングの唯一の候補となります。当然、このデバイスは1台のブレードコンピュータを管理する必要があります。

ライトアウトデバイス

ライトアウトデバイス(IBM RSA、HP iLO、Dell DRAC)は急速に広まっており、今後は既製コンピュータの標準装備になる可能性さえあります。ただし、電源をホスト(クラスタノード)と共有するため、これらはUPSデバイスに内蔵されています。ノードに電力が供給されないままでは、それを制御するデバイスも役に立ちません。その場合は、CRMがノードのフェンシングの試行を無限に続け、他のすべてのリソース操作はフェンシング/STONITH操作の完了を待機することになります。

テストिंगデバイス

テストिंगデバイスは、テスト専用に使われます。通常、ハードウェアにあまり負担をかけないようにしています。クラスタが運用に使用される際には、実際のフェンシングデバイスに交換されます。

STONITHデバイスは、予算と使用するハードウェアの種類に応じて選択します。

9.2.2 STONITHの実装

SUSE® Linux Enterprise High Availability Extension のSTONITH実装には、2つのコンポーネントがあります。

stonithd

stonithdは、ローカルプロセスまたはネットワーク経由でアクセスできるデーモンです。stonithdは、フェンシング操作に対応するコマンド(rest、

power-off、power-on)を受け入れます。フェンシングデバイスの状態チェックも行います。

stonithdデーモンはCRM HAクラスタの各ノードで実行されます。DCノードで実行されるstonithdインスタンスは、CRMからフェンシング要求を受け取ります。目的のフェンシング操作を実行するのは、このインスタンスとその他のstonithdプログラムです。

STONITHプラグイン

サポートされているフェンシングデバイスごとに、そのデバイスを制御できるSTONITHプラグインがあります。STONITHプラグインはフェンシングデバイスへのインタフェースです。すべてのSTONITHプラグインは各ノードの/usr/lib/stonith/pluginsにあります。すべてのSTONITHプラグインはstonithdからは同一のものと認識されますが、フェンシングデバイスの性質を反映しているため、大きな違いがあります。

一部のプラグインは、複数のデバイスをサポートします。代表的な例はipmilan(またはexternal/ipmi)で、IPMIプロトコルを実装し、このプロトコルをサポートする任意のデバイスを制御できます。

9.3 STONITHの構成

フェンシングをセットアップするには、1つまたは複数のSTONITHリソースを設定する必要があります。stonithdデーモンでは設定は不要です。すべての構成はCIBに保存されます。STONITHリソースはクラスstonithのリソースです(4.2.2項「サポートされるリソースエージェントクラス」(40 ページ)を参照)。STONITHリソースはSTONITHプラグインのCIBでの表現です。フェンシング操作の他、STONITHリソースはその他のリソースと同様、開始、停止、監視できます。STONITHリソースの開始と停止とは、この場合STONITHの有効化と無効化を意味します。開始と停止は管理上の操作であるため、フェンシングデバイス自体での操作にはなりません。ただし、監視はデバイス状態に反映されます。

STONITHリソースはその他のリソースと同様にして構成できます。リソースの構成については、5.3.2項「STONITHリソースの作成」(70 ページ)または6.3.3項「STONITHリソースの作成」(106 ページ)を参照してください。

パラメータ(属性)のリストは、それぞれのSTONITHの種類に依存します。特定のデバイスのパラメーター一覧を表示するには、stonithコマンドを実行します。

```
stonith -t stonith-device-type -n
```

たとえば、ibmhmcデバイスタイプのパラメータを表示するには、次のように入力します。

```
stonith -t ibmhmc -n
```

デバイスの簡易ヘルプテキストを表示するには、-hオプションを使用します。

```
stonith -t stonith-device-type -h
```

9.3.1 STONITHリソースの構成例

以降では、crmコマンドラインツールの構文で作成された構成例を紹介します。これを適用するには、サンプルをテキストファイル(sample.txtなど)に格納して、実行します。

```
crm < sample.txt
```

crmコマンドラインツールでのリソースの構成については、第6章 クラスタリソースの設定と管理(コマンドライン) (95 ページ)を参照してください。

警告: テスティングの構成

次の例の一部は、説明およびテストのみを目的としています。テストिंगの構成例を実際のクラスタシナリオで使用しないでください。

例 9.1 テスティングの構成

```
configure
primitive st-null stonith:null \
params hostlist="node1 node2"
clone fencing st-null
commit
```

例 9.2 テスティングの構成

別の構成:

```
configure
primitive st-node1 stonith:null \
params hostlist="node1"
primitive st-node2 stonith:null \
params hostlist="node2"
location l-st-node1 st-node1 -inf: node1
location l-st-node2 st-node2 -inf: node2
commit
```

この構成例は、クラスタソフトウェアに関してはまったく問題ありません。実際の構成との違いは、フェンシング操作が行われないことです。

例 9.3 テスティングの構成

より現実的な例(ただし、テスト目的のみ)として、次のexternal/ssh設定があります。

```
configure
primitive st-ssh stonith:external/ssh \
params hostlist="node1 node2"
clone fencing st-ssh
commit
```

これも、ノードをリセットできます。この構成は、null STONITHデバイスを利用する最初の例と非常によく似ています。この例では、クローンが使用されています。これはCRM/Pacemakerの機能です。クローンは、基本的には、ショートカットです。同一だが名前の異なるn個のリソースを定義する代わりに、1つのクローンしたリソースで足ります。すべてのノードからSTONITHデバイスがアクセス可能である限り、クローンの最も一般的な使用方法是、STONITHリソースで使用する事です。

例 9.4 IBM RSA ライトアウトデバイスの構成

実際のデバイス構成とはそれほど違いはありませんが、一部のデバイスにはより多くの属性が必要となります。IBM RSA ライトアウトデバイスは、次のようにして構成できます。

```
configure
primitive st-ibmrsa-1 stonith:external/ibmrsa-telnet \
params nodename=node1 ipaddr=192.168.0.101 \
userid=USERID passwd=PASSWORD
primitive st-ibmrsa-2 stonith:external/ibmrsa-telnet \
params nodename=node2 ipaddr=192.168.0.102 \
userid=USERID passwd=PASSWORD
location l-st-node1 st-ibmrsa-1 -inf: node1
location l-st-node2 st-ibmrsa-2 -inf: node2
commit
```

この例では、`location`制約が使用されていますが、それは、STONITH操作が常に一定の確率で失敗するからです。したがって、(実行側でもあるノード上の)STONITH操作は信頼できません。ノードがリセットされていない場合、フェンシング操作結果について通知を送信できません。これを実行する方法は、操作が成功すると仮定して事前に通知を送信するほかありません。ただし、操作が失敗した場合は、問題が発生する可能性があります。したがって、`stonithd`はホストの強制終了を拒否します。

例 9.5 UPSフェンシングデバイスの構成

UPSタイプのフェンシングデバイスの設定は、上記の例と似ています。詳細は、(演習として)読者に任されています。UPSデバイスは、フェンシングと同じメカニズムを採用していますが、デバイス自体へのアクセス方法は異なります。古いUPSデバイスにはシリアルポートしかなく、ほとんどの場合、特別のシリアルケーブルを使用して1200ボーで接続していました。新型の多くは、まだシリアルポートがありますが、USBインタフェースまたはイーサネットインタフェースも備えています。使用できる接続の種類は、プラグインが何をサポートしているかによって異なります。

たとえば、apcmasterをapcsmartデバイスと、`stonith -t stonith-device-type -n`コマンドを使用して比較します。

```
stonith -t apcmaster -h
```

次の情報が返されます。

```
STONITH Device: apcmaster - APC MasterSwitch (via telnet)
NOTE: The APC MasterSwitch accepts only one (telnet)
connection/session a time. When one session is active,
subsequent attempts to connect to the MasterSwitch will fail.
For more information see http://www.apc.com/
List of valid parameter names for apcmaster STONITH device:
ipaddr
login
password
```

今度は次のコマンドを使用します。

```
stonith -t apcsmart -h
```

次の結果が得られます。

```
STONITH Device: apcsmart - APC Smart UPS
(via serial port - NOT USB!).
Works with higher-end APC UPSes, like
Back-UPS Pro, Smart-UPS, Matrix-UPS, etc.
(Smart-UPS may have to be >= Smart-UPS 700?).
See http://www.networkupstools.org/protocols/apcsmart.html
for protocol compatibility details.
For more information see http://www.apc.com/
List of valid parameter names for apcsmart STONITH device:
ttydev
hostlist
```

最初のプラグインは、ネットワークポートとtelnetプロトコルを持つAPC UPSをサポートします。2番目のプラグインはAPC SMARTプロトコルをシリアル

回線で使用します。これはその他多数のAPC UPS製品ラインでサポートされているものです。

9.3.2 制約とクローン

9.3.1項「STONITHリソースの構成例」(135 ページ)では、STONITHリソースを制約、クローン、またはその両方を使用して設定する方法をいくつか説明しました。設定にどちらのコンストラクトを使用するかは、いくつかの要因(フェンシングデバイスの性質、デバイスで管理されるホスト数、クラスタノード数、または個人の好み)によって決まります。

まとめると、クローンを構成で安心して使用でき、構成が縮小される場合は、クローンされたSTONITHリソースを使用します。

9.4 フェンシングデバイスの監視

他のリソースとまったく同様に、STONITHクラスのエージェントは、ステータスのチェックに使用される監視操作もサポートします。

注記: STONITHリソースの監視

STONITHリソースの監視を強く推奨します。定期的に、間隔を置いて監視します。

フェンシングデバイスはHAクラスタの不可欠な要素ですが、使用する必要が少ないほど好都合です。電源管理装置は通信側では脆弱であることが知られています。ブロードキャストトラフィックが多すぎると、一部のデバイスは機能しません。1分間に10本程度の接続しか処理できないものもあります。2つのクライアントが同時に接続しようすると、混乱するデバイスもあります。大部分は、同時に複数のセッションを処理できません。

したがって、ほとんどの場合、フェンシングデバイスは数時間ごとにチェックすれば十分です。この数時間のうちにフェンシング操作が必要になり、電源スイッチで障害が発生する確率は通常低いものです。

監視操作の構成方法の詳細は、GUIアプローチについては手順5.3「メタ属性およびインスタンス属性を追加または変更する」(68 ページ)、コマンドライ

ンアプローチについては6.3.8項「リソース監視の設定」(113ページ)を参照してください。

9.5 特殊なフェンシングデバイス

実際のSTONITHデバイスを操作するプラグインとは別に、一部のSTONITHプラグインについては、説明を追加する必要があります。

警告: テスト目的のみ

次に示すSTONITHプラグインの一部は、デモとテストだけを目的としています。次のデバイスは、現実のシナリオでは使用しないでください。使用すると、データが損なわれたり、予測できない結果が生じることがあります。

- `external/ssh`
- `ssh`
- `null`

`external/kdumpcheck`

このプラグインは、ノードでカーネルダンプが進行中かどうかチェックする場合に有用です。進行中の場合は、`true`を返すので、そのノードはフェンシングされている(その時点でノードがリソースを実行できない)かのように見えます。これによって、すでにダウンしているがダンプ中(これは時間がかかります)であるノードのフェンシングを避けることができます。このプラグインは、別の実際のSTONITHデバイスとともに使用する必要があります。詳細については、`/usr/share/doc/packages/cluster-glue/README_kdumpcheck.txt`を参照してください。

`external/sbd`

これは自己フェンシングデバイスです。共有ディスクに挿入されることがある、いわゆる「ポイズンピル」に反応します。共有ストレージ接続が失われると、ノードの動作も停止します。このSTONITHエージェントでフェンシングに基づいてストレージを実装する方法については、第15章 スト

レージ保護(195 ページ)を参照してください。詳細については、http://www.linux-ha.org/wiki/SBD_Fencingも参照してください。

external/ssh

別のソフトウェアベースの「フェンシング」メカニズム。ノードは、rootとして、パスワードなしでお互いにログインする必要があります。このメカニズムは、1つのパラメータhostlistをとり、ターゲットにするノードを指定します。これは、本当に障害のあるノードをリセットすることはできないので、実際のクラスタには使用しないでください。これは、テストとデモの目的にのみ使用します。これを共有ストレージに使用すると、データが破損します。

meatware

meatwareではユーザが操作を支援する必要があります。起動すると、meatwareはードのコンソールに表示されるCRIT重大度メッセージを記録します。その場合、オペレータはノードがダウンしていることを確認して、meatclient(8)コマンドを発行します。これにより、meatwareにノードがダウンしていると思われることをクラスタに通知できることを認識させます。詳細については、/usr/share/doc/packages/cluster-glue/README.meatwareを参照してください。

null

これはさまざまなテストデバイスで使用される仮想デバイスです。常に、ノードを停止したかのように動作し、そのように主張しますが、まったく何もしません。処理内容を理解している場合を除き、使用しないでください。

suicide

これはソフトウェアのみのデバイスで、rebootコマンドを使用して実行しているノードを再起動できます。これにはノードのオペレーティングシステムによる操作が必要で、特定の状況では失敗することがあります。したがって、このデバイスの使用は、極力避けてください。ただし、1ノードクラスタで使用する場合は安全です。

suicideとnullは、「自分のホストを停止させない」というルールへの唯一の例外です。

9.6 詳細情報

`/usr/share/doc/packages/cluster-glue`

インストールしたシステムで、このディレクトリには多数のSTONITHプラグインおよびデバイスのREADMEファイルが格納されています。

<http://www.linux-ha.org/wiki/STONITH>

STONITHに関する情報。High Availability Linux Projectのホームページにあります。

http://www.clusterlabs.org/doc/crm_fencing.html

フェンシングに関する情報。Pacemaker Projectのホームページにあります。

http://www.clusterlabs.org/doc/en-US/Pacemaker/1.0/html/Pacemaker_Explained

Pacemakerの設定に使用されるコンセプトの説明。包括的で非常に詳細な参照用情報です。

http://techthoughts.typepad.com/managing_computers/2007/10/split-brain-quo.html

HAクラスタでのスプリットブレイン、クォーラム、フェンシングのコンセプトを説明する記事。

Linux Virtual Serverによる負荷分散

10

LVS(Linux Virtual Server)は、複数のサーバにネットワーク接続を振り分けてワークロードを共有させる基本フレームワークの提供を目的としています。Linux Virtual Serverは、1つ以上のロードバランサとサービス実行用の数台の実際のサーバから成るサーバクラスタですが、外部のクライアントには1つの高速な大型サーバのように見えます。この単一サーバのように見えるサーバは、*仮想サーバ*と呼ばれます。Linux Virtual Serverは、高度にスケーラブルで可用性の高いネットワークサービス(Web、キャッシュ、メール、FTP、メディア、VoIPなど)の構築に使用できます。

実際のサーバとロードバランサは、高速LANまたは地理的に分散されたWANのいずれでも、相互に接続できます。ロードバランサは、さまざまなサーバに要求をディスパッチできます。ロードバランサによって、クラスタのパラレルサービスが1つのIPアドレス(仮想IPアドレスまたはVIP)上の仮想サービスであるかのようにみえます。要求のディスパッチでは、IP負荷分散技術か、アプリケーションレベル負荷分散技術を使用できます。クラスタ内のノードのトランスペアレントな追加または削除によって、システムのスケーラビリティが達成されます。ノードまたはデーモンの障害の検出とシステムの適宜な再設定によって、高度な可用性が提供されます。

10.1 概念の概要

以降のセクションでは、主要なLVSのコンポーネントと概念の概要を示します。

10.1.1 Director

LVSの主要コンポーネントは、`ip_vs` (またはIPVS)カーネルコードです。このコードは、Linuxカーネル内でトランスポート層の負荷分散(レイヤ4スイッチング)を実装します。IPVSコードを含むLinuxカーネルを実行するノードは、ディレクタと呼ばれます。ディレクタで実行されるIPVSコードは、LVSの必須機能です。

クライアントがディレクタに接続すると、着信要求がすべてのクラスタノードに負荷分散されます。つまり、ディレクタは、変更されたルーティングルール(LVSを機能させる)セットを使用して、パケットを実サーバに転送します。たとえば、ディレクタは、接続の送受信側でないと、受信確認を送信しません。ディレクタは、エンドユーザから実サーバ(要求を処理するアプリケーションを実行するホスト)にパケットを転送する特殊なルータとして動作します。

デフォルトでは、IPVSモジュールはカーネルにインストールされていません。IPVSカーネルモジュールは、`cluster-network-kmp-default`パッケージに含まれています。

10.1.2 ユーザスペースのコントローラとデーモン

`ldirectord`デーモンは、Linux Virtual Serverを管理し、負荷分散型仮想サーバのLVSクラスタ内の実サーバを監視するユーザスペースデーモンです。設定ファイル`/etc/ha.d/ldirectord.cf`は、仮想サービスとそれらに関連付けられた実サーバを指定し、LVSリダイレクタとしてサーバを設定する方法を`ldirectord`に指示します。このデーモンは、その初期化時にクラスタの仮想サービスを生成します。

`ldirectord`デーモンは、既知のURLを定期的に要求し、応答を確認することにより、実サーバのヘルスを監視します。障害が発生した実サーバは、ロードバランサで使用可能なサーバのリストから削除されます。サービスモニタは、ダウンしていたサーバが回復し、再度機能していることを検出すると、そのサーバを使用可能サーバリストに戻します。すべての実サーバがダウンする場合については、Webサービスのリダイレクト先にするフォールバックサーバを指定できます。通常、フォールバックサーバは、ローカルホストで

あり、Webサービスが一時的に使用できないことについて緊急ページを表示します。

10.1.3 パケット転送

ディレクタがクライアントから実サーバにパケットを送信する方法は、3つあります。

NAT (Network Address Translation)

着信要求は、仮想IPで着信し、宛先のIPアドレスとポートを、選択された実サーバのIPアドレスとポートに変更することで、実サーバに転送されます。実サーバはロードバランサに応答を送信し、ロードバランサは、宛先IPアドレスを変更し、応答をクライアントへ転送します。その結果、エンドユーザは予期されたソースから応答を受信します。すべてのトラフィックはロードバランサを通過するので、通常、ロードバランサがクラスタのボトルネックになります。

IPトンネリング(IP-IPカプセル化)

IPトンネリングでは、あるIPアドレスにアドレス指定されたパケットを別のアドレス(別のネットワーク上でも可能)にリダイレクトできます。LVSは、IPトンネルを介して実サーバに要求を送信し(別のIPアドレスにリダイレクト)、実サーバは、独自のルーティングテーブルを使用して、クライアントに直接応答します。クラスタメンバは、さまざまなサブネットに属することができます。

直接ルーティング

エンドユーザからのパケットを、直接、実サーバに転送します。IPパケットは変更されない所以、仮想サーバのIPアドレスのトラフィックを受け付けるように、実サーバを設定する必要があります。実サーバからの応答は、直接、クライアントに送信されます。実サーバとロードバランサは、同じ物理ネットワークセグメントに属する必要があります。

10.1.4 スケジューリングアルゴリズム

クライアントから要求された新しい接続に使用する実サーバの決定は、さまざまなアルゴリズムを使用して実装されます。それらは、モジュールとして使用可能であり、特定のニーズに合わせて調整できます。使用可能なモジュールの概要については、`ipvsadm(8)`のマニュアルページを参照してください。

ディレクタは、クライアントから接続要求を受信すると、スケジュールに基づいて実際のサーバをクライアントに割り当てます。スケジューラは、IPVSカーネルコードの一部として、次の新しい接続を取得する実際のサーバを決定します。

10.2 YaSTによるIP負荷分散の設定

YaST `iplb` モジュールを使用して、カーネルベースのIP負荷分散を設定できます。このモジュールは、`ldirectord` のフロントエンドです。

IP負荷分散ダイアログにアクセスするには、`root` として YaST を開始し、`[高可用性] > [IP負荷分散]` の順に選択します。または、コマンドラインで `「yast2 iplb」` を入力して、`root` として YaST クラスタモジュールを起動します。

YaST モジュールは、その設定を `/etc/ha.d/ldirectord.cf` に書き込みます。YaST モジュール内で使用できるタブは、設定ファイル `/etc/ha.d/ldirectord.cf` の構造、グローバルオプションの定義、および仮想サービス用オプションの定義に対応しています。

設定例とその結果のロードバランサ/実サーバ間のプロセスについては、例10.1「単純な `ldirectord` 設定」(151 ページ)を参照してください。

注記: グローバルパラメータと仮想サーバパラメータ

特定のパラメータを仮想サーバセクションとグローバルセクションの両方で指定した場合は、仮想サーバセクションで定義した値が、グローバルセクションで定義した値に優先します。

手順 10.1 グローバルパラメータを設定する

次の手順では、最も重要なグローバルパラメータの設定方法を示します。個々のパラメータ(および、ここに記載されていないパラメータ)の詳細については、`[ヘルプ]` をクリックするか、`ldirectord` のマニュアルページを参照してください。

- 1 `[確認間隔]` で、`ldirectord` が各実サーバに接続していて、それらがまだオンラインかどうか確認する間隔を定義します。

- 2 [確認タイムアウト] で、最後の確認後に実サーバが応答する期限を設定します。
- 3 [確認回数] では、ldirectordが、何回、実サーバに要求すると、確認が失敗したと見なされるか定義できます。
- 4 [ネゴシエーションタイムアウト] で、ネゴシエーション確認のタイムアウトを秒単位で定義します。
- 5 [フォールバック] で、すべての実サーバがダウンした場合にWebサービスのリダイレクト先にするWebサーバのホスト名とIPアドレスを入力します。
- 6 ロギングに代替パスを使用する場合は、[ログファイル] でログのパスを指定します。デフォルトでは、ldirectordは、そのログを/var/log/ldirectord.logに書き込みます。
- 7 実サーバへの接続状態が変わったら、システムにアラートを送信したい場合は、有効な電子メールアドレスを[電子メールアラート]に入力します。
- 8 [電子メールアラート頻度] で、実サーバにアクセスできない状態が続く場合、何秒後に電子メールアラートを繰り返すか定義します。
- 9 [電子メールアラートステータス] で、電子メールアラートを送信する必要があるサーバの状態を指定します。複数の状態を定義する場合は、カンマで区切ったリストを使用します。
- 10 [自動リロード] で、変更の有無について、ldirectordに設定ファイルを継続的に監視させるかどうか定義します。yesに設定した場合は、変更のたびに、設定ファイルが自動的にリロードされます。
- 11 [休止] スイッチで、障害が発生した実サーバをカーネルのLVSテーブルから削除するかどうか定義します。[はい] に設定すると、障害のあるサーバは削除されません。代わりに、それらの重み付けが0に設定され、新しい接続が受け入れられなくなります。すでに確立している接続は、タイムアウトするまで持続します。

図 10.1 YaST IP 負荷分散 - グローバルパラメータ

IPLB - Global Configuration

Global Configure Virtual Server Configure

Check Interval: 5 Check Timeout: 3 Check Count: Negotiate Timeout:

Fallback:

Log File:

Email Alert: Email Alert Freq: Email Alert Status:

Callback: Execute:

Auto Reload: yes Quiescent: yes Fork: Supervised:

Help Cancel OK

手順 10.2 仮想サービスを設定する

仮想サービスごとに、2、3のパラメータを定義することによって、1つ以上の仮想サービスを設定できます。次の手順で、仮想サービスの最も重要なパラメータを設定する方法を示します。個々のパラメータ(および、ここに記載されていないパラメータ)の詳細については、[ヘルプ]をクリックするか、`ldirectord`のマニュアルページを参照してください。

- 1 YaSTiplbモジュール内で、[仮想サーバ設定] タブに切り替えます。
- 2 [追加] で新しい仮想サーバを追加するか、[編集] で既存の仮想サーバを編集します。新しいダイアログに、使用可能なオプションが表示されます。
- 3 [仮想サーバ] で、共有される仮想IPアドレスとポートを入力します。これらのアドレスとポートで、ロードバランサと実サーバをLVSとしてアクセスできます。IPアドレスとポート名の代わりに、ホスト名とサービスを指定できます。または、ファイアウォールマークを使用することもできます。ファイアウォールマークは、VIP:portサービスの任意の集まりを1つの仮想サービスにまとめる方法です。

- 4 [実サーバ] で実際のサーバを指定するには、サーバのIPアドレス(またはホスト名)、ポート(またはサービス名)、および転送方法を入力する必要があります。転送方法は、gate、ipip、またはmasqのいずれかにする必要があります(10.1.3項「パケット転送」(145 ページ)参照)。

[追加] ボタンをクリックし、実サーバごとに必要な引数を入力します。
- 5 [確認タイプ] で、実サーバがまだアクティブかどうかをテストするために実行する必要がある確認のタイプを選択します。たとえば、要求を送信し、応答に予期どおりの文字列が含まれているかどうか確認するには、[ネゴシエーション] を選択します。
- 6 [確認のタイプ] を[ネゴシエーション] に設定した場合は、監視するサービスのタイプも定義する必要があります。[サービス] ドロップダウンリストから選択してください。
- 7 [要求] で、確認間隔中に各実サーバで要求されるオブジェクトへのURLを入力します。
- 8 実サーバからの応答に一定の文字列(「I'm alive」メッセージ)が含まれているかどうか確認する場合は、一致する必要がある正規表現を定義します。正規表現を[受信] に入力します。実サーバからの応答にこの表現が含まれている場合、実サーバはアクティブとみなされます。
- 9 ステップ6(149ページ)で選択した[サービス] のタイプによっては、さらにパラメータを指定する必要があります(たとえば、[ログイン]、[パスワード]、[データベース]、[秘密] など)。詳細については、YaSTヘルプのテキストか、ldirectordのマニュアルページを参照してください。
- 10 ロードに使用する[スケジューラ] を選択します。使用可能なスケジューラについては、ipvsadm(8)のマニュアルページを参照してください。
- 11 使用する[プロトコル] を選択します。仮想サービスをIPアドレスとポートとして指定する場合は、プロトコルをtcpまたはudpのどちら

かにする必要があります。仮想サービスをファイアウォールマークとして指定する場合は、プロトコルをfwmにする必要があります。

- 必要な場合は、さらにパラメータを定義します。[OK] を選択して、設定を確認します。YaSTが設定を/etc/ha.d/ldirectord.cfに書き込みます。

図 10.2 YaST IP 負荷分散 - 仮想サービス

IPLB - Virtual Servers Configuration

Virtual Server
192.168.0.200:80

Real Servers
192.168.0.110:80 gate
192.168.0.120:80 gate

Check Type: negotiate Service: http Check Command: Check Port:

Request: "test.html" Receive: "still alive" Http Method: Virtual Host:

Login: Password: Database Name: Radius Secret:

Persistent: Netmask: Scheduler: wlc Protocol: tcp

Check Timeout: Negotiate Timeout: Check Count: Email Alert:

Email Alert Freq: Email Alert Status: Fallback 127.0.0.1:80 Quiescent:

Help Cancel OK

例 10.1 単純なldirectord設定

図10.1「YaST IP負荷分散 - グローバルパラメータ」(148 ページ)と図10.2「YaST IP負荷分散 - 仮想サービス」(150 ページ)で示された値を使用すると、次のような設定になり、/etc/ha.d/ldirectord.cfで定義されます。

```
autoreload = yes ❶
checkinterval = 5 ❷
checktimeout = 3 ❸
quiescent = yes ❹
    virtual = 192.168.0.200:80 ❺
    checktype = negotiate ❻
    fallback = 127.0.0.1:80 ❼
    protocol = tcp ❽
    real = 192.168.0.110:80 gate ❾
    real = 192.168.0.120:80 gate ❾
    receive = "still alive" ❿
    request = "test.html" ⓫
    scheduler = wlc ⓬
    service = http ⓭
```

- ❶ ldirectordが変更の有無について設定ファイルを継続的に確認するように定義します。
- ❷ 実サーバがまだオンラインかどうか確認するため、ldirectordが各実サーバに接続する間隔。
- ❸ 最後の確認後、実サーバが応答しなければならない時間的な期限
- ❹ 障害が発生した実サーバをカーネルのLVSテーブルから削除せず、代わりに、それらのサーバの重み付けを0に設定します。
- ❺ LVSの仮想IPアドレス(VIP)。LVSはポート80で使用できます。
- ❻ 実サーバがまだアクティブかどうかをテストするための確認のタイプ。
- ❼ このサービス用のすべての実サーバがダウンしている場合に、Webサービスのリダイレクト先にするサーバ。
- ❽ 使用するプロトコル。
- ❾ ポート80で利用できる2つの実サーバが定義されています。パケットの転送方法がgateなので、直接ルーティングが使用されます。
- ❿ 実サーバからの応答文字列内で一致する必要がある正規表現。
- ⓫ 確認間隔中に、各実サーバで要求されるオブジェクトへのURI。
- ⓬ 負荷分散に使用するスケジューラが選択されています。

⑬ 監視するサービスのタイプ

この設定を使用すると、次のような処理フローになります:ldirectordが、5秒ごとに各実サーバに接続し②、⑨と⑩で指定されているように、192.168.0.110:80/test.htmlまたは192.168.0.120:80/test.htmlを要求します。予期されたstill alive文字列⑩を、最後の確認から3秒以内③に実サーバから受信しない場合は、実サーバが使用可能なサーバのプールから削除されます。ただし、quiescent=yesが設定されているので④、実サーバは、LVSテーブルからは削除されず、代わりに、その重み付けが0に設定されます。その結果、この実サーバへの新しい接続は受け付けられなくなります。すでに確立されている接続は、タイムアウトするまで持続します。

10.3 追加設定

YaSTによるldirectordの設定に加えて、LVS設定を完了するには、次の条件を満たす必要があります。

- ・ 実サーバは、必要なサービスを提供するように正しく設定します。
- ・ 負荷分散サーバは、IP転送を使用して実サーバにトラフィックをルーティングできる必要があります。実サーバのネットワーク設定は、選択したパケット転送方法によって左右されます。
- ・ 負荷分散サーバをシステム全体のシングルポイント障害にしないため、ロードバランサのバックアップを1つ以上セットアップする必要があります。クラスタ設定では、ldirectordにプリミティブリソースを設定して、ハードウェア障害の場合にldirectordが他のサーバにフェールオーバーできるようにします。
- ・ ロードバランサのバックアップにも、そのタスクを達成するために、ldirectord設定ファイルが必要なので、ロードバランサのバックアップとして使用するすべてのサーバ上で/etc/ha.d/ldirectord.cfが使用できるようにします。設定ファイルは、3.2.3項「すべてのノードへの設定の転送」(26 ページ)で説明されているように、Csync2で同期できます。

10.4 詳細情報

Linux Virtual Serverの詳細については、プロジェクトのホームページ(<http://www.linuxvirtualserver.org/>)を参照してください。

ldirectordの詳細については、その総合的なマニュアルページを参照してください。

ネットワークデバイスボンディング

11

多くのシステムで、通常のEthernetデバイスの標準データセキュリティ/可用性の要件を超えるネットワーク接続の実装が望ましいことがあります。その場合、数台のEthernetデバイスを集めて1つのボンディングデバイスを構成できます。

ボンディングデバイスの構成には、ボンディングモジュールオプションを使用します。ボンディングデバイスの動作は、ボンディングデバイスのモードによって決定されます。デフォルトの動作は、mode=active-backupであり、アクティブなスレーブに障害が発生すると、別のスレーブデバイスがアクティブになります。

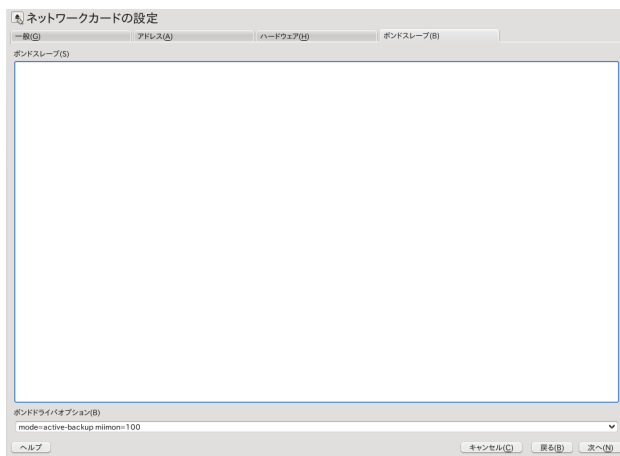
OpenAISの使用時は、クラスタソフトウェアでボンディングデバイスが管理されることはありません。したがって、ボンディングデバイスにアクセスする可能性のあるクラスタノードごとに、ボンディングデバイスを設定する必要があります。

11.1 YaSTによるボンディングデバイスの設定

ボンディングデバイスを設定するには、次の手順に従います。

- 1 rootとしてYaSTを開始し、[ネットワークデバイス] > [ネットワーク設定] の順に選択します。

- 2 [追加] をクリックして、新しいネットワークカードを設定し、[デバイスの型] を [ボンド] に変更します。[次へ] で続行します。



- 3 IPアドレスをボンディングデバイスに割り当てる方法を選択します。3つの方法から選択できます。

- IPアドレスなし
- 可変IPアドレス(DHCPまたは Zeroconf)
- 固定IPアドレス

ご使用の環境に適合する方法を使用します。OpenAISが仮想IPアドレスを管理する場合は、[固定IPアドレス] を選択し、インタフェースに基本IPアドレスを割り当てます。

- 4 [ボンドスレーブ] タブに切り替えます。
- 5 ボンドに含めるイーサネットデバイスを選択するため、[ボンドスレーブ] の該当するオプションのチェックボックスをオンにします。
- 6 [ボンドドライバオプション] を編集します。次のモードを使用できます。

`balance-rr`

負荷分散と耐障害性を提供します。

`active-backup`

耐障害性を提供します。

`balance-xor`

負荷分散と耐障害性を提供します。

ブロードキャスト

耐障害性を提供します。

`802.3ad`

接続されるスイッチでサポートされる場合は、ダイナミックリンク集合を提供します。

`balance-tlb`

発信トラフィックの負荷分散を提供します。

`balance-alb`

使用中にハードウェアアドレスの変更が可能なネットワークデバイスを使用する場合は、着信トラフィックと発信トラフィックの負荷分散を提供します。

- 7 パラメータ`miimon=100`が [ボンドドライバオプション] に追加されていることを確認します。このパラメータがないと、データの整合性が定期的にチェックされません。
- 8 [次へ] をクリックし、[OK] でYaSTを終了して、デバイスを作成します。

11.2 詳細情報

すべてのモードと他の多数のオプションの詳細は、「[*Linux Ethernet Bonding Driver HOWTO*]」に記載されています。このドキュメントは、`kernel-source`パッケージをインストールすれば、`/usr/src/linux/Documentation/networking/bonding.txt`で読むことができます。

パート III. ストレージおよびデータレプリケーション

Oracle Cluster File System 2

OCFS2 (Oracle Cluster File System 2) は、Linux 2.6以降のカーネルに完全に統合されている汎用ジャーナリングファイルシステムです。Oracle Cluster File System 2を利用すれば、アプリケーションバイナリファイル、データファイル、およびデータベースを、共有ストレージ中のデバイスに保管することができます。このファイルシステムには、クラスタ中のすべてのノードが同時に読み書きすることができます。ユーザスペース管理デーモンは、クローンリソースを介して管理され、HAスタック(特に、OpenAIS/CorosyncおよびDLM (Distributed Lock Manager))との統合を実現します。

12.1 特長と利点

OCFS2は、たとえば、次のストレージソリューションに使用できます。

- 一般のアプリケーションとワークロード。
- クラスタ内のXenイメージストア。Xen仮想マシンと仮想サーバは、クラスタサーバによってマウントされたOCFS2ボリュームに保存できます。これによって、サーバ間でXen仮想マシンを素早く容易に移植できます。
- LAMP(Linux、Apache、MySQL、およびPHP | PERL | Python)スタック。

OCF2は、高パフォーマンスでシンメトリックなパラレルクラスタファイルシステムとして、次の機能をサポートします。

- アプリケーションのファイルを、クラスタ内のすべてのノードで使用できます。ユーザは、クラスタ中のOracle Cluster File System 2ボリュームに1回インストールするだけで構いません。
- すべてのノードが、標準ファイルシステムインタフェースを介して、同時並行的に、ストレージに直接読み書きできるので、クラスタ全体に渡って実行されるアプリケーションの管理が容易になります。
- ファイルアクセスがDLMを介して調整されます。ほとんどの場合、DLMによる制御は適切に機能しますが、アプリケーションの設計によっては、アプリケーションとDLMがファイルアクセスの調整で競合すると、スケラビリティが制限されることがあります。
- すべてのバックエンドストレージで、ストレージのバックアップ機能を利用することができます。共有アプリケーションファイルのイメージを簡単に作成することができるため、災害発生時でも素早くデータを復元することができます。

Oracle Cluster File System 2には、次の機能も用意されています。

- メタデータのキャッシュ処理。
- メタデータのジャーナル処理。
- ノード間にまたがるファイルデータの整合性 j。
- 最大16TBまでのボリュームで、最高4KBまでの複数ブロックサイズをサポート(各ボリュームで異なるブロックサイズを使用可能)。
- 16台までのクラスタノードをサポート。
- データベースのパフォーマンスを向上する非同期、直接I/Oのサポート。

12.2 OCFS2のパッケージと管理ユーティリティ

OCFS2カーネルモジュール(ocfs2)は、自動的に、SUSE® Linux Enterprise Server 11 SP1上のHigh Availability Extensionにインストールされます。OCFS2

を使用するには、ocfs2-toolsと、ご使用のカーネルに適合するocfs2-kmp-*パッケージが、クラスタの各ノードにインストールされていることを確認してください。

ocfs2-toolsパッケージには、次に示すOCFS2ボリュームの管理ユーティリティがあります。構文については、各マニュアルページを参照してください。

表 12.1 OCFS2ユーティリティ

OCFS2ユーティリティ	説明
debugfs.ocfs2	デバッグの目的で、Oracle Cluster File System 2のファイルシステムの状態を調査します。
fsck.ocfs2	ファイルシステムにエラーがないかをチェックし、必要に応じてエラーを修復します。
mkfs.ocfs2	デバイス上にOCFS2ファイルシステムを作成します。通常は、共有物理/論理ディスク上のパーティションに作成します。
mounted.ocfs2	クラスタシステム上のすべてのOCFS2ボリュームを検出、表示します。OCFS2デバイスをマウントしているシステム上のすべてのノードを検出、表示するか、またはすべてのOCFS2デバイスを表示します。
tuneufs.ocfs2	ボリュームラベル、ノードスロット数、すべてのノードスロットのジャーナルサイズ、およびボリュームサイズなど、OCFS2ファイルのシステムパラメータを変更します。

12.3 OCFS2サービスの設定

OCFS2ボリュームを作成する前に、次のリソースをクラスタ内のサービスとして設定する必要があります: DLMおよびO2CBOCFS2はPacemakerからのクラスタメンバーシップサービスを使用し、それらのサービスはユーザスペース

で実行されます。したがって、**DLM**と**O2CB**は、クラスタ内の各ノードに存在するクローンリソースとして設定する必要があります。

手順 12.1 DLMリソースとO2CBリソースを設定する

次の手順では、`crm`シェルを使用してクラスタリソースを設定します。クラスタ内の1つのノードについて、次の手順を実行してください。リソースの設定には、**Heartbeat**を使用することもできます。

- 1 端末ウィンドウを開いて、`root`またはそれと同等のユーザとしてログインします。
- 2 **DLM (Distributed Lock Manager)**をリソースとして追加するには、次の手順に従います。

2a `crm`シェルを起動し、新しい設定を最初から作成します。

```
crm
cib new stack-glue
```

2b **DLM**サービスを作成し、クラスタ内のすべてのコンピュータで実行させます。

```
configure
primitive dlm ocf:pacemaker:controld op monitor interval=120s
clone dlm-clone dlm meta globally-unique=false interleave=true
end
```

`dlm`クローンリソースが、分散ロックマネージャサービスを制御し、クラスタ内のすべてのノードでこのサービスが開始するようにします。

2c 変更内容を検証後、それらを**CIB**にコミットします。

```
cib diff
configure verify
```

2d 設定をクラスタにアップロードし、シェルを終了します。

```
cib commit stack-glue
quit
```

3 **O2CB**設定を追加するには、次の手順に従います。

3a crmシェルを起動し、新しい設定を最初から作成します。

```
crm
cib new oracle-glue
```

3b O2CBサービスをクラスタ内のすべてのノードで開始させます。

```
configure
primitive o2cb ocf:ocfs2:o2cb op monitor interval=120s
clone o2cb-clone o2cb meta globally-unique=false interleave=true
```

3c O2CBサービスを、すでに実行中のdlmサービスのコピーを持つノードでのみ開始させるには、コロケーションの制約を追加します。

```
colocation o2cb-with-dlm INFINITY: o2cb-clone dlm-clone
order start-o2cb-after-dlm mandatory: dlm-clone o2cb-clone
```

3d 設定をクラスタにアップロードし、シェルを終了します。

```
cib commit oracle-glue
quit
```

4 フェンシングデバイスを設定するには、次の手順に従います。

4a crmシェルを起動し、新しい設定を最初から作成します。

```
crm
cib new fencing
```

4b external/sdbをフェンシングデバイスとして設定し、/dev/sdb2を共有ストレージ上のハートビートとフェンシング専用のパーティションにします。

```
configure
primitive sbd_stonith stonith:external/sbd \
meta target-role="Started"op monitor \
interval=15 timeout=15 start-delay=15 \
params sbd_device=/dev/sdb2
```

4c 設定をクラスタにアップロードし、シェルを終了します。

```
cib commit fencing
quit
```

12.4 OCFS2ボリュームの作成

12.3項「OCFS2サービスの設定」(163ページ)で説明されているように、DLMとO2CBをクラスタリソースとして設定したら、システムがOCFS2を使用できるように設定し、OCFS2ボリュームを作成します。

注記: アプリケーションファイルとデータファイル用のOCFS2ボリューム

一般に、アプリケーションファイルとデータファイルは、異なるOCFS2ボリュームに保存することを推奨します。アプリケーションボリュームとデータボリュームのマウント要件が異なる場合は、必ず、異なるボリュームに保存します。

作業を始める前に、OCFS2ボリュームに使用するブロックデバイスを準備します。デバイスは空き領域のままにしてください。

次に、手順12.2「OCFS2ボリュームを作成し、フォーマットする」(168ページ)で説明されているように、mkfs.ocfs2で、OCFS2ボリュームを作成し、フォーマットします。そのコマンドの最も重要なパラメータは、表12.2「重要なOCFS2パラメータ」(166ページ)に一覧されています。詳細情報とコマンド構文については、mkfs.ocfs2のマニュアルページを参照してください。

表 12.2 重要なOCFS2パラメータ

OCFS2パラメータ	説明と推奨設定
ボリュームラベル(-L)	異なるノードへのマウント時に、正しく識別できるように、一意のわかりやすいボリューム名を指定します。ラベルを変更するには、tunefs.ocfs2ユーティリティを使用します。
クラスタサイズ(-C)	クラスタサイズは、ファイルに割り当てられる、データ保管領域の最小単位です。使用できるオプションと推奨事項については、mkfs.ocfs2のマニュアルページを参照してください。

OCFS2パラメータ	説明と推奨設定
ノードスロット数(-N)	<p data-bbox="505 224 1091 483">同時にボリュームをマウントできる最大ノード数を指定します。各ノードについて、OCFS2はジャーナルなどの個別のシステムファイルを作成します。ボリュームにアクセスするノードに、リトルエンディアン形式のノード(x86、x86-64、およびia64など)とビッグエンディアン形式のノード(ppc64やs390xなど)が混在しても構いません。</p> <p data-bbox="505 521 1091 683">ノード固有のファイルは、ローカルファイルとして参照されます。ローカルファイルには、ノードスロット番号が付加されます。たとえば、journal:0000は、スロット番号0に割り当てられたノードに属します。</p> <p data-bbox="505 721 1091 948">各ボリュームを同時にマウントすると予期されるノード数に従って、各ボリュームの作成時に、そのボリュームの最大ノードスロット数を設定します。tunefs.ocfs2ユーティリティを使用して、必要に応じてノードスロットの数を増やします。ただし、この値は減らすことはできません。</p>
ブロックサイズ(-b)	<p>ファイルシステムがアドレス可能な領域の最小単位を指定します。ブロックサイズは、ボリュームの作成時に指定します。使用できるオプションと推奨事項については、mkfs.ocfs2のマニュアルページを参照してください。</p>
特定機能のオン/オフ (--fs-features)	<p>カンマで区切った機能フラグリストを指定できます。mkfs.ocfs2は、そのリストに従って、それらの機能セットを含むファイルシステムを作成をしようとします。機能をオンにするには、その機能をリストに入れます。機能をオフにするには、その名前の前にnoを付けます。</p>

OCFS2パラメータ	説明と推奨設定
	使用できるすべてのフラグの概要については、 <code>mkfs.ocfs2</code> のマニュアルページを参照してください。
事前定義機能 (<code>--fs-feature-level</code>)	事前定義されたファイルシステム機能セットから選択できます。使用できるオプションについては、 <code>mkfs.ocfs2</code> のマニュアルページを参照してください。

`mkfs.ocfs2`によるボリュームの作成およびフォーマット時に特定の機能を指定しない場合は、次の機能がデフォルトで有効になります。

`backup-super`、`sparse`、`inline-data`、`unwritten`、`metaecc`、`indexed-dirs`、および`xattr`。

手順 12.2 OCFS2ボリュームを作成し、フォーマットする

クラスタノードの1つだけで、次の手順を実行します。

- 1 端末ウィンドウを開いて、`root`.としてログインします。
- 2 クラスタがオンラインであることを`crm_mon`で確認します。
- 3 `mkfs.ocfs2`ユーティリティを使用して、ボリュームを作成およびフォーマットします。このコマンドの指定形式については、`mkfs.ocfs2`マニュアルページを参照してください。

たとえば、最大16台のクラスタノードをサポートする新しいOCFS2ファイルシステムを`/dev/sdb1`上に作成するには、次のコマンドを使用します。

```
mkfs.ocfs2 -N 16 /dev/sdb1
```

12.5 OCFS2ボリュームのマウント

OCFS2ボリュームは、手動でマウントするか、クラススタマネージャでマウントできます(手順12.4「クラススタマネージャでOCFS2ボリュームをマウントする」(169 ページ)参照)。

手順 12.3 OCFS2ボリュームを手動でマウントする

- 1 端末ウィンドウを開いて、root.としてログインします。
- 2 クラスタがオンラインであることをcrm_monで確認します。
- 3 コマンドラインから、mountコマンドを使ってボリュームをマウントします。

警告: 手動マウントによるOCFS2デバイス

OCFS2ファイルシステムをテスト目的で手動マウントした場合、そのファイルシステムは、いったんマウント解除してから、OpenAISで使用してください。

手順 12.4 クラススタマネージャでOCFS2ボリュームをマウントする

High Availability ソフトウェアでOCFS2ボリュームをマウントするには、クラスタ内でocfFile Systemリソースを設定します。次の手順では、crmシェルを使用してクラスタリソースを設定します。リソースの設定には、Heartbeatを使用することもできます。

- 1 crmシェルを起動し、新しい設定を最初から作成します。

```
crm
cib new filesystem
```

- 2 OCFS2ファイルシステムをクラスタ内のすべてのノードにマウントするように、Pacemakerを設定します。

```
configure
primitive fs ocf:heartbeat:Filesystem \
  params device="/dev/sdb1" directory="/mnt/shared" fstype="ocfs2" \
  op monitor interval=120s
```

```
clone fs-clone fs meta interleave="true" ordered="true"
```

- 3 Pacemakerが、すでに実行中のo2cbリソースのクローンを持つノードでのみ、fsクローンリソースを開始させるには、次のコロケーション制約を追加します。

```
colocation fs-with-o2cb INFINITY: fs-clone o2cb-clone  
order start-fs-after-o2cb mandatory: o2cb-clone fs-clone
```

- 4 設定をCIBにアップロードし、シェルを終了します。

```
cib commit filesystem  
quit
```

12.6 詳細情報

OCFS2の詳細については、次のリンクを参照してください。

<http://oss.oracle.com/projects/ocfs2/>

OracleサイトにあるOCFS2プロジェクトのホームページ

<http://oss.oracle.com/projects/ocfs2/documentation>

プロジェクトドキュメントのホームページにある『*OCFS2 User's Guide*』

Distributed Replicated Block Device (DRBD)

13

DRBDを使用すると、IPネットワーク内の2つの異なるサイトに位置する2つのブロックデバイスのミラーを作成できます。OpenAISと共に使用すると、DRBDは分散高可用性Linuxクラスタをサポートします。この章では、DRBDのインストールとセットアップの方法を示します。

13.1 概念の概要

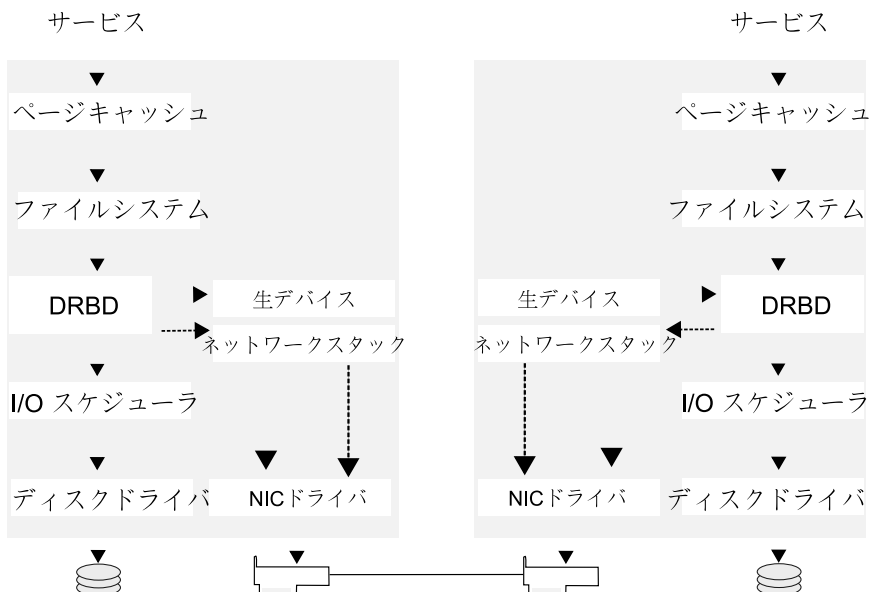
DRBDは、プライマリデバイス上のデータをセカンダリデバイスに、データの両方のコピーが同一に保たれるような方法で複製します。これは、ネットワーク型のRAID 1と考えてください。DRBDは、データをリアルタイムでミラーリングするので、そのレプリケーションは連続的に起こります。アプリケーションは、実際そのデータがさまざまなディスクに保存されるということを知る必要はありません。

重要項目: 暗号化されないデータ

ミラー間のデータトラフィックは暗号化されません。データ交換を安全にするには、接続に仮想プライベートネットワーク(VPN)ソリューションを導入する必要があります。

DRBDは、Linuxカーネルモジュールであり、下端のI/Oスケジューラと上端のファイルシステムの間には存在しています(図13.1「Linux内でのDRBDの位置」(172 ページ)参照)。DRBDと通信するには、高レベルのコマンドdrbdadmを使用します。柔軟性を最大にするため、DRBDには、低レベルのツールdrbdsetupが付いてきます。

図 13.1 Linux内でのDRBDの位置



DRBDでは、Linuxでサポートされる任意のブロックデバイスを使用できます。通常は次のデバイスです。

- パーティションまたは完全なハードディスク
- ソフトウェアRAID
- LVM (Logical Volume Manager)
- EVMS (Enterprise Volume Management System)

DRBDは、デフォルトでは、DRBDノード間の通信にTCPポート7780以上を使用します。ファイアウォールでこのポートの通信が許可されていることを確認してください。

まず、DRBDデバイスを設定してから、その上にファイルシステムを作成する必要があります。ユーザデータに関することはすべて、rawデバイスではなく、`/dev/drbd_R`デバイスを介してのみ実行される必要があります。これは、DRBDが、メタデータ用にrawデバイスの最後の128MBを使用するからで

す。ファイルシステムは、`raw`デバイス上ではなく、`/dev/drbd<n>`デバイス上にのみ作成するようにしてください。

たとえば、`raw`デバイスのサイズが1024MBの場合、DRBDデバイスは、896MBしかデータ用に使用できません。128MBは隠され、メタデータ用に予約されています。896MB～1024MBのスペースへのアクセスは、そのスペースがユーザデータ用でないので、すべて失敗します。

13.2 DRBDサービスのインストール

DRBDに必要なパッケージをインストールするには、パートI「インストールおよび管理」(1 ページ)で説明されているように、High Availability Extension アドオン製品をネットワーククラスタの両方のSUSE Linux Enterprise Serverコンピュータにインストールします。High Availability Extensionをインストールすると、DRBDプログラムファイルもインストールされます。

完全なクラスタスタックは必要なく、DRBDを使用したいだけの場合は、表13.1「DRBD RPMパッケージ」(173 ページ)に、DRBDのすべてのRPMパッケージが一覧されています。最新のバージョンで、`drbd`パッケージは、複数のパッケージに分割されました。

表 13.1 DRBD RPMパッケージ

ファイル名	説明
<code>drbd</code>	いくつかに分割された便利なパッケージ
<code>drbd-bash-completion</code>	プログラマブル <code>bash</code> 補完のサポート(<code>drbdadm</code> 用)
<code>drbd-heartbeat</code>	DRBD用Heartbeatリソースエージェント (Heartbeat用にのみ必要)
<code>drbd-kmp-default</code>	DRBD用カーネルモジュール(必要)
<code>drbd-kmp-xen</code>	DRBD用Xenカーネルモジュール

ファイル名	説明
drbd-udev	DRBD用のudev統合スクリプト。/dev/drbd/by-resと/dev/drbd/by-diskでDRBDデバイスへのシンボリックリンクを管理します。
drbd-utils	DRBD用管理ユーティリティ(必要)
drbd-pacemaker	DRBD用Pacemakerリソースエージェント
drbd-xen	DRBD用Xenブロックデバイス管理スクリプト
yast2-drbd	YaST DRBD環境設定(推奨)

drbdadmの操作を簡素化するには、RPMパッケージdrbd-bash-completionにあるBash補完サポートを使用します。現在のシェルセッションでこのサポートを有効にするには、次のコマンドを挿入します。

```
source /etc/bash_completion.d/drbdadm.sh
```

root用に永続的に使用するには、ファイル/root/.bashrcを作成し、上記の行を挿入します。

13.3 DRBDサービスの設定

注記

次の手順では、サーバ名としてjupiterとvenusを使用し、クラスタリソース名としてr0を使用します。jupiterは、プライマリノードとして設定します。必ず、手順を変更して、ご使用のノード名とファイル名を使用するようにしてください。

DRBDの設定を始める前に、Linuxノード内のブロックデバイスを準備し、(必要な場合は)パーティション分割しておいてください。次の手順では、jupiterとvenusという2つのノードがあり、それらがTCPポート7780を使用すると想定します。ファイアウォールでこのポートが開いているようにしてください。

DRBDを手動で設定するには、次の手順に従います。

手順 13.1 DRBDを手動で設定する

1 rootとしてログインします。

2 DRBDの環境設定ファイルを変更します。

2a ファイル/etc/drbd.confを開き、次の行を挿入します(これらの行がない場合)。

```
include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```

DRBD 8.3以降、環境設定ファイルは、複数のファイルに分割され、/etc/drbd.dディレクトリに保存されています。

2b /etc/drbd.d/global_common.confファイルを開きます。このファイルには、すでいくつかの事前定義値が含まれています。startupセクションに移動し、次の3行を挿入します。

```
startup {
    # wfc-timeout degr-wfc-timeout outdated-wfc-timeout
    # wait-after-sb;
    wfc-timeout 1;
    degr-wfc-timeout 1;
}
```

これらのオプションは、ブート時のタイムアウトを減らすために使
用します。詳細については、[http://www.drbd.org/
users-guide-emb/re-drbdconf.html](http://www.drbd.org/users-guide-emb/re-drbdconf.html)を参照してください。

2c ファイル/etc/drbd.d/r0.resを作成し、状況に合わせて行を変
更し、ファイルを保存します。

```
resource r0 { ❶
    device /dev/drbd_r0 minor 0; ❷
    disk /dev/sdal; ❸
    meta-disk internal; ❹
    on jupiter { ❺
        address 192.168.1.10:7780; ❻
    }
    on venus { ❺ (page 175)
        address 192.168.1.11:7780; ❻ (page 175)
    }
    syncer {
        rate 7M; ❼
    }
}
```

- ❶ リソースの名前。r0、r1などのリソース名の使用を推奨します。
 - ❷ DRBD用デバイス名とそのマイナー番号。/dev/drbdで始め、リソース名(この場合、r0)を付加することを推奨します。
 - ❸ ノード間で複製されるデバイス。ただし、この例では、デバイスは両方のノードで同じです。異なるデバイスが必要な場合は、diskパラメータをonセクションに移動します。
 - ❹ meta-diskパラメータには、通常、値internalが含まれますが、メタデータを保持する明示的なデバイスを指定することも可能です。詳細については、<http://www.drbd.org/users-guide-emb/ch-internals.html#s-metadata>を参照してください。
 - ❺ onセクションには、ノードのホスト名が含まれます。
 - ❻ それぞれのノードのIPアドレスとポート番号。リソースごとに、通常、7780から始まる別個のポートが必要です。
 - ❼ 同期レート。このレートは、ご使用の帯域幅の3分の1に設定します。これは、再同期を制限するだけで、ミラーリングは制限しません。
- 3 環境設定ファイルの構文をチェックします。次のコマンドがエラーを返す場合は、ファイルを検証します。

```
drbdadm dump all
```

- 4 DRBD環境設定ファイルをもう一方のノードにコピーします。

```
scp /etc/drbd.conf venus:/etc/  
scp /etc/drbd.d/* venus:/etc/drbd.d/
```

- 5 各ノードで次のコマンドを入力することにより、両方のシステムでメタデータを初期化します。

```
drbdadm -- --ignore-sanity-checks create-md r0  
rcdrbd start
```

ディスクに、必要のなくなったファイルシステムがすでに含まれている場合は、次のコマンドでファイルシステムの構造を破壊し、このステップを繰り返します。

```
dd if=/dev/zero of=/dev/sdb1 count=10000
```

- 6** 次のコマンドを各ノードで入力して、DRBDステータスをチェックします。

```
rcdrbd status
```

次のような出力が表示されます。

```
drbd driver loaded OK; device status:
version: 8.3.7 (api:88/proto:86-91)
GIT-hash: ea9e28dbff98e331a62bcbcc63a6135808fe2917 build by phil@fat-tyre, 2010-01-13
17:17:27
m:res  cs          ro          ds          p  mounted  fstype
0:r0   Connected  Secondary/Secondary  Inconsistent/Inconsistent  C
```

- 7** プライマリにしたいノード(この場合、**jupiter**)で再同期プロセスを開始します。

```
drbdadm -- --overwrite-data-of-peer primary r0
```

- 8** `rcdrbd status`を使用して、再びステータスをチェックすると、次のような結果が得られます。

```
...
m:res  cs          ro          ds          p  mounted  fstype
0:r0   Connected  Primary/Secondary  UpToDate/UpToDate  C
```

`ds`行のステータス(ディスクステータス)は、両方のノードで`UpToDate`である必要があります。

- 9** **jupiter**をプライマリノードとして設定します。

```
drbdadm primary r0
```

- 10** DRBDデバイスの上にファイルシステムを作成します。たとえば、次のように指定します。

```
mkfs.ext3 /dev/drbd_r0
```

- 11** ファイルシステムをマウントして使用します。

```
mount /dev/drbd_r0 /mnt/
```

13.4 DRBDサービスのテスト

インストールと設定のプロシージャが予期どおりの結果となった場合は、DRBD機能の基本的なテストを実行できます。このテストは、DRBDソフトウェアの機能を理解する上でも役立ちます。

1 **jupiter**でDRBDサービスをテストします。

1a 端末コンソールを開き、**root**としてログインします。

1b **jupiter**にマウントポイント(**/srv/r0mount**など)を作成します。

```
mkdir -p /srv/r0mount
```

1c **drbd**デバイスをマウントします。

```
mount -o rw /dev/drbd0 /srv/r0mount
```

1d プライマリノードからファイルを作成します。

```
touch /srv/r0mount/from_node1
```

2 **venus**でDRBDサービスをテストします。

2a 端末コンソールを開き、**root**としてログインします。

2b **jupiter**でディスクをマウント解除します。

```
umount /srv/r0mount
```

2c **jupiter**で次のコマンドを入力することにより、**jupiter**上のDRBDサービスを降格します。

```
drbdadm secondary r0
```

2d **venus**で、DRBDサービスをプライマリに昇格します。

```
drbdadm primary r0
```

2e **venus**で、**venus**がプライマリかどうかチェックします。

```
rcdrbd status
```

2f **venus**で、`/srv/r0mount`などのマウントポイントを作成します。

```
mkdir /srv/r0mount
```

2g **venus**で、**DRBD**デバイスをマウントします。

```
mount -o rw /dev/drbd0 /srv/r0mount
```

2h **jupiter**で作成したファイルを表示できることを確認します。

```
ls /srv/r0mount
```

`/srv/r0mount/from_node1`ファイルがリストされるはずです。

3 サービスが両方のノードで稼動していれば、**DRBD**の設定は完了です。

4 再度、**jupiter**をプライマリとして設定します。

4a 次のコマンドを**venus**で入力して、**venus**のディスクをディスマウントします。

```
umount /srv/r0mount
```

4b 次のコマンドを**venus**で入力して、**venus**上の**DRBD**サービスを降格します。

```
drbdadm secondary r0
```

4c **jupiter**で、**DRBD**サービスをプライマリに昇格します。

```
drbdadm primary r0
```

4d **jupiter**で、**jupiter**がプライマリかどうかチェックします。

```
rcdrbd status
```

5 サービスを自動的に起動させ、サーバに問題が発生した場合はフェールオーバーさせるためには、**OpenAIS**で**DRBD**を高可用性サービスとして設定できます。**OpenAIS for SUSE Linux Enterprise 11**のインストールと構成の詳細は、パートII「設定および管理」(35 ページ)を参照してください。

13.5 DRBDのチューニング

DRBDをチューニングするには、いくつかの方法があります。

1. メタデータ用には外部ディスクを使用します。これによって、接続速度が向上します。
2. DRBDデバイスの先読み設定を変更するudevルールを作成します。次の行をファイル/etc/udev/rules.d/82-dm-ra.rulesに保存し、`read_ahead_kb`値を独自の作業負荷に変更します。

```
ACTION=="add", KERNEL=="dm-*", ATTR{bdi/read_ahead_kb}="4100"
```

この行は、LVMの使用時のみ機能します。

3. LinuxソフトウェアRAIDシステムで**bmbv**をアクティブにします。次の行をDRBD環境設定の**common disk**セクション(通常、/etc/drbd.d/global_common.confにある)で使します。

```
disk {  
    use-bmbv;  
}
```

13.6 DRBDのトラブルシューティング

drbdセットアップには、多数の異なるコンポーネントが使用され、別のソースから問題が発生することがあります。以降のセクションでは、一般的なシナリオをいくつか示し、さまざまなソリューションを推奨します。

13.6.1 設定

初期のdrbdセットアップが予期どおりに機能しない場合は、おそらく、環境設定に問題があります。

環境設定の情報を取得するには:

- 1 端末コンソールを開き、`root`としてログインします。

- 2 drbdadmに-dオプションを指定して、環境設定ファイルをテストします。次のコマンドを入力します。

```
drbdadm -d adjust r0
```

adjustオプションのドライランでは、drbdadmは、DRBDリソースの実際の設定をご使用のDRBD環境設定ファイルと比較しますが、コールは実行しません。出力をレビューして、エラーのソースおよび原因を確認してください。

- 3 /etc/drbd.d/*ファイルとdrbd.confファイルにエラーがある場合は、そのエラーを修正してから続行してください。
- 4 パーティションと設定が正しい場合は、drbdadmを-dオプションなしで、再度実行します。

```
drbdadm adjust r0
```

このコマンドは、環境設定ファイルをDRBDリソースに適用します。

13.6.2 ホスト名

DRBDの場合、ホスト名では大文字小文字を区別します(たとえば、Node0は、node0とは異なるホストです)。

複数のネットワークデバイスがあり、専用ネットワークデバイスを使用したい場合、おそらく、ホスト名は使用されたIPアドレスに解決されません。この場合は、パラメータdisable-ip-verificationを使用します。

13.6.3 TCPポート 7788

システムがピアに接続できない場合は、ローカルファイアウォールに問題のある可能性があります。DRBDは、デフォルトでは、TCPポート7788を使用して、もう一方のノードにアクセスします。このポートを両方のノードからアクセスできるかどうか確認してください。

13.6.4 DRBDデバイスが再起動後に破損した

DRBDサブシステムが実際のどのデバイスが最新データを保持しているか認識していない場合、スプリットブレイン受験に変更されます。この場合、それぞれのDRBDサブシステムがセカンダリとして機動され、互いに接続しません。この場合、次のメッセージが/var/log/messagesに書き込まれます。

```
Split-Brain detected, dropping connection!
```

この状況を解決するには、廃棄するデータを持つノードで、次のコマンドを入力します。

```
drbdadm secondary r0  
drbdadm -- --discard-my-data connect r0
```

最新のデータを持つノードで、次のコマンドを入力します。

```
drbdadm connect r0
```

13.7 詳細情報

DRBDについては、次のオープンソースリソースを利用できます。

- プロジェクトホームページ<http://www.drbd.org>。
- http://clusterlabs.org/wiki/DRBD_HowTo_1.0(Linux Pacemaker Cluster Stack Projectによる)。
- ディストリビューションで利用できるDRBDのマニュアルページは次のとおりです。drbd(8)、drbddisk(8)、drbdsetup(8)、drbdsetup(8)、drbdadm。(8)、drbd.conf(5)
- コメント付きのDRBD構成例が、/usr/share/doc/packages/drbd/drbd.confにあります。

クラスタLVM

クラスタ上の共有ストレージを管理する場合、ストレージサブシステムに行った変更を各ノードに伝える必要があります。Linux Volume Manager 2 (LVM2) はローカルストレージの管理に多用されており、クラスタ全体のボリュームグループのトランスペアレントな管理をサポートするために拡張されています。クラスタ化されたボリュームグループを、ローカルストレージと同じコマンドで管理できます。

14.1 概念の概要

クラスタLVMは、さまざまなツールと連携します。

分散ロックマネージャ(DLM:Distributed Lock Manager)
c LVMのためにディスクアクセスを調整します。

論理ボリュームマネージャ2(LVM2: Logical Volume Manager2)
1つのファイルシステムをいくつかのディスクに柔軟に分散することができます。LVMは、ディスクスペースの仮想プールを提供します。

クラスタ化論理ボリュームマネージャ(c LVM: Clustered Logical Volume Manager)
すべてのノードが変更を知ることができるように、LVMメタデータへのアクセスを調整します。cLVMは、共有データ自体へのアクセスは調整しません。これをc LVMができるようにするには、OCFS2などのクラスタ対応アプリケーションをcLVMの管理対象ストレージの上に設定する必要があります。

14.2 cLVMの環境設定

ご使用のシナリオによっては、次のレイヤを使用して、cLVMでRAID 1デバイスを作成することができます。

- **LVM** ファイルシステムのサイズを増減したり、物理ストレージを追加したり、ファイルシステムのスナップショットを作成する場合に、高い柔軟性を提供するソリューションです。この方法については、14.2.1項「シナリオ - SAN上でiSCSIを使用するcLVM」(185 ページ)に説明があります。
- **DRBD** RAID 0 (ストライピング)とRAID 1 (ミラーリング)のみを提供します。最後の方式については、14.2.2項「シナリオ - DRBDを使用するcLVM」(190 ページ)に説明があります。
- **MDデバイス(LinuxソフトウェアRAIDまたはmdadm)** このソリューションは、すべてのRAIDレベルを提供しますが、まだクラスタはサポートしていません。

次の前提条件を満たしていることを確認してください。

- 共有ストレージデバイス(Fibre Channel、FCoE、SCSI、iSCSI SAN、DRBD)で提供されているデバイスなどが使用できること
- DRBDの場合は、両方のノードがプライマリであること(以降の手順で説明)。
- LVM2のロックタイプがクラスタを認識するかどうか確認すること。/etc/lvm/lvm.conf内のキーワードlocking_typeに値3がデフォルトで含まれている必要があります。必要な場合は、この設定をすべてのノードにコピーします。

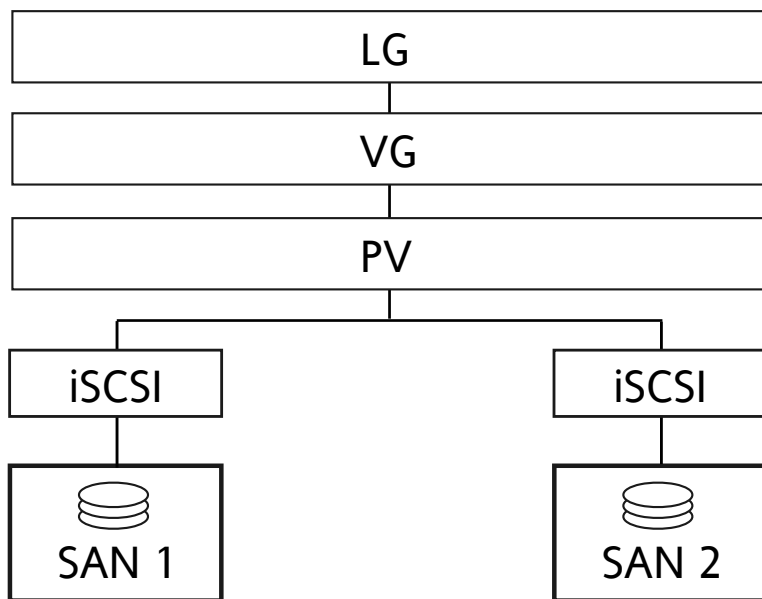
注記: 最初にクラスタリソースを作成する

まず、クラスタリソースを作成してから、LVMボリュームを作成してください。そうしないと、後でボリュームを削除できなくなります。

14.2.1 シナリオ - SAN上でiSCSIを使用する c LVM

次のシナリオでは、iSCSIターゲットをいくつかのクライアントにエクスポートする2つのSANボックスを使用します。一般的なアイデアが、図14.1「c LVMによるiSCSIのセットアップ」(185 ページ)で説明されています。

図 14.1 c LVMによるiSCSIのセットアップ



警告: データ損失

以降の手順を実行すると、ディスク上のデータはすべて破壊されます。

まず、1つのSANボックスだけ設定します。各SANボックスは、そのiSCSIターゲットをエクスポートする必要があります。次の手順に従います。

手順 14.1 iSCSIターゲット(SAN上)を設定する

- 1 YaSTを実行し、[ネットワークサービス] > [iSCSIターゲット] の順にクリックしてiSCSIサーバモジュールを起動します。

- 2 コンピュータがブートするたびにiSCSIターゲットを起動したい場合は、
[ブート時] を選択し、そうでない場合は、[手動] を選択します。
- 3 ファイアウォールが実行中の場合は、[ファイアウォールでポートを開く] を有効にします。
- 4 [グローバル] タブに切り替えます。認証が必要な場合は、受信または送信(あるいはその両方の)認証を有効にします。この例では、[認証なし] を選択します。
- 5 新しいiSCSIターゲットを追加します。
 - 5a [ターゲット] タブに切り替えます。
 - 5b [追加] をクリックします。
 - 5c ターゲットの名前を入力します。名前は、次のようにフォーマットされます。
`iqn.DATE.DOMAIN`
 - 5d より説明的な名前にしたい場合は、さまざまなターゲット間で一意であれば、識別子を変更できます。
 - 5e [追加] をクリックします。
 - 5f [パス] にデバイス名を入力し、[Scsiid] を使用します。
 - 5g [次へ] を2回クリックします。
- 6 警告ボックスで [はい] を選択して確認します。
- 7 環境設定ファイル/etc/iscsi/iscsi.confを開き、パラメータ node.startupをautomaticに変更します。

次の手順に従って、iSCSIイニシエータを設定します。

手順 14.2 iSCSIイニシエータを設定する

- 1 YaSTを実行し、[ネットワークサービス] > [iSCSIイニシエータ]の順にクリックします。
- 2 コンピュータがブートするたびに、iSCSIイニシエータを起動したい場合は、[ブート時]を選択し、そうでない場合は、[手動]を選択します。
- 3 [検出] タブに切り替え、[検出] ボタンをクリックします。
- 4 自分のIPアドレスとiSCSIターゲットのポートを追加します(手順14.1「iSCSIターゲット(SAN上)を設定する」(185ページ)参照)。通常は、ポートを既定のままにし、デフォルト値を使用できます。
- 5 認証を使用する場合は、受信および送信用のユーザ名およびパスワードを挿入します。そうでない場合は、[認証なし]を選択します。
- 6 [次へ] を選択します。検出された接続が一覧されます。
- 7 [完了] をクリックして続行します。
- 8 シェルを開き、rootとしてログインします。
- 9 iSCSIイニシエータが正常に起動しているかどうかテストします。

```
iscsiadm -m discovery -t st -p 192.168.3.100  
192.168.3.100:3260,1 iqn.2010-03.de.jupiter:san1
```

- 10 セッションを確立します。

```
iscsiadm -m node -l  
Logging in to [iface: default, target: iqn.2010-03.de.jupiter:san2,  
portal: 192.168.3.100,3260]  
Logging in to [iface: default, target: iqn.2010-03.de.venus:san1,  
portal: 192.168.3.101,3260]  
Login to [iface: default, target: iqn.2010-03.de.jupiter:san2, portal:  
192.168.3.100,3260]: successful  
Login to [iface: default, target: iqn.2010-03.de.venus:san1, portal:  
192.168.3.101,3260]: successful
```

lsscsiでデバイス名を表示します。

```
...
[4:0:0:2]    disk    IET        ...    0    /dev/sdd
[5:0:0:1]    disk    IET        ...    0    /dev/sde
```

3番目の列にIETを含むエントリを捜します。この場合、該当するデバイスは、/dev/sddと/dev/sdeです。

手順 14.3 DLMリソースを作成する

- 1 シェルを起動し、rootとしてログインします。
- 2 crm configureを実行します。
- 3 次のコマンドを入力します。

```
primitive dlm ocf:pacemaker:controld
primitive clvm ocf:lvm2:clvmd \
    params daemon_timeout="30"
group dlm-clvm dlm clvm
clone dlm-clvm-clone dlm-clvm \
    meta interleave="true" ordered="true"
```

- 4 showで変更内容をレビューします。
- 5 すべて正しい場合は、commitを入力し、exitでcrmを終了します。

手順 14.4 LVMボリュームグループを作成する

- 1 手順14.2「iSCSIイニシエータを設定する」(187 ページ)のiSCSIイニシエータを実行したノードの1つで、rootシェルを開きます。
- 2 ディスク/dev/sddおよび/dev/sdeでコマンドpvcreateを使用して、LVM用に物理ボリュームを準備します。

```
pvcreate /dev/sdd
pvcreate /dev/sde
```

- 3 pvdisplayで、すべて正しいかどうかチェックします。

```
--- Physical volume ---
PV Name                /dev/sdd
VG Name                clustervg
PV Size                509,88 MB / not usable 1,88 MB
Allocatable            yes
PE Size (KByte)        4096
```



```

Total PE          127
Free PE           127
Allocated PE      0
PV UUID           52okH4-nv3z-2AUL-GhAN-8DAZ-GMtU-Xrn9Kh

--- Physical volume ---
PV Name           /dev/sde
VG Name           clustervg
PV Size           509,84 MB / not usable 1,84 MB
Allocatable       yes
PE Size (KByte)   4096
Total PE          127
Free PE           127
Allocated PE      0
PV UUID           Ouj3Xm-AI58-lxB1-mWm2-xn51-agM2-0UuHFC

```

4 両方のディスク上でクラスタ対応のボリウムグループを作成します。

```
vgcreate --clustered y clustervg /dev/sdd /dev/sde
```

5 vgdisplayで、すべて正しいかどうかチェックします。

```

--- Volume group ---
VG Name           clustervg
System ID
Format            lvm2
Metadata Areas    2
Metadata Sequence No 1
VG Access         read/write
VG Status         resizable
Clustered         yes
Shared            no
MAX LV            0
Cur LV           0
Open LV           0
Max PV            0
Cur PV           2
Act PV            2
VG Size           1016,00 MB
PE Size           4,00 MB
Total PE          254
Alloc PE / Size   0 / 0
Free PE / Size    254 / 1016,00 MB
VG UUID           UCyWw8-2jqV-enuT-KH4d-NXQI-JhH3-J24anD

```

6 必要に応じて、論理ボリウムを作成します。

```
lvcreate --name clusterlv --size 500M clustervg
```

ボリュームを作成してリソースを起動すると、/dev/dm-0とい名前の新しいデバイスができています。LVMリソースの上でクラスタ化されたファイルシステム(たとえば、OCFS)を使用することをお勧めします。詳細については、第12章 *Oracle Cluster File System 2* (161 ページ)を参照してください。

14.2.2 シナリオ - DRBDを使用する c LVM

市、国、または大陸の各所にデータセンタが分散している場合は、次のシナリオを使用できます。

手順 14.5 DRBDでクラスタ対応ボリュームグループを作成する

1 プライマリ/プライマリDRBDリソースを作成する

1a まず、手順13.1「DRBDを手動で設定する」(175 ページ)の説明に従って、DRBDデバイスをプライマリ/セカンダリとしてセットアップします。ディスクの状態が両方のノードでup-to-dateであることを確認します。これは、`cat /proc/drbd`または`rcdrbd status`で確認します。

1b 次のオプションを環境設定ファイル(通常は、`/etc/drbd.d/r0.res`)に追加します。

```
resource r0 {
    startup {
        become-primary-on both;
    }

    net {
        allow-two-primaries;
    }
    ...
}
```

1c 変更した設定ファイルをもう一方のノードにコピーします。たとえば、次のように指定します。

```
scp /etc/drbd.d/r0.res venus:/etc/drbd.d/
```

1d 両方のノードで、次のコマンドを実行します。

```
drbdadm disconnect r0
drbdadm connect r0
drbdadm primary r0
```

1e ノードのステータスをチェックします。

```
cat /proc/drbd
...
0: cs:Connected ro:Primary/Primary ds:UpToDate/UpToDate C r----
```

2 `clvmd`リソースをペースメーカーの環境設定でクローンとして保存し、DLMクローンリソースに依存させます。詳細については、手順14.3「DLMリソースを作成する」(188ページ)を参照してください。次に進む前に、クラスタでこれらのリソースが正しく機動していることを確認してください。`crm_mon`またはGUIを使用して、実行中のサービスを確認できます。

3 `pvcreate`コマンドで、LVM用に物理ボリュームを準備します。たとえば、`/dev/drbd_r0`デバイスでは、コマンドは次のようになります。

```
pvcreate /dev/drbd_r0
```

4 クラスタ対応のボリュームグループを作成します。

```
vgcreate --clustered y myclusterfs /dev/drbd_r0
```

5 必要に応じて、論理ボリュームを作成します。論理ボリュームのサイズは変更できます。たとえば、次のコマンドで、4ギガバイトの論理ボリュームを作成します。

```
lvcreate --name testlv -L 4G myclusterfs
```

6 ボリュームグループがクラスタ全体でアクティブ化されるようにするには、LVMリソースを次のように設定します。

```
primitive vg1 ocf:heartbeat:LVM \
    params volgrpname="myclusterfs"
clone vg1-clone vg1 \
    meta interleave="true" ordered="true"
colocation colo-vg1 inf: vg1-clone dlm-clvm-clone
order order-vg1 inf: dlm-clvm-clone vg1-clone
```

- 7 ボリュームグループを1ノードだけで排他的にアクティブ化したい場合は、次の例を使用します。この場合、クラスタ化されていないアプリケーション保護の追加対策として、**cLVM**は**VG**内のすべての論理ボリュームが複数ノードでアクティブ化されないように保護します。

```
primitive vg1 ocf:heartbeat:LVM \
    params volgrpname="myclusterfs" exclusive="yes"
colocation colo-vg1 inf: vg1 dlm-clvm-clone
order order-vg1 inf: dlm-clvm-clone vg1
```

- 8 **VG**内の論理ボリュームは、ファイルシステムのマウントまたは**raw**用として使用できるようになりました。論理ボリュームを使用しているサービスに कोरोケシヨンのための正しい依存性があることを確認し、**VG**をアクティブ化したら論理ボリュームの順序付けを行います。

このような設定手順を終了すると、**LVM2**の環境設定は他のスタンドアロンワークステーションと同様に行えます。

14.3 有効な**LVM2**デバイスの明示的な設定

複数のデバイスが同じ物理ボリュームの署名を共有していると思われる場合(マルチパスデバイスやdrbdなどのように)、**LVM2**が**PV**を走査するデバイスを明示的に設定しておくことをお勧めします。

たとえばコマンド**vgcreate**がミラーブロックデバイスの代わりに物理デバイスを使用すると、**DRBD**は混乱してしまい、**DRBD**のスプリットブレイン状態が発生する場合があります。

LVM2用の単一のデバイスを非アクティブ化するには、次の手順に従います。

- 1 ファイル/etc/lvm/lvm.confを編集し、**filter**から始まる行を検索します。
- 2 そこに記載されているパターンは正規表現として処理されます。冒頭の「a」は走査にデバイスパターンを受け入れることを、冒頭の「r」はそのデバイスパターンのデバイスを拒否することを意味します。

- 3 /dev/sdb1という名前のデバイスを削除するには、次の表現をフィルタルールに追加します。

```
"r|^/dev/sdb1$|"
```

完全なフィルタ行は次のようになります。

```
filter = [ "r|^/dev/sdb1$|", "r|/dev/.*/by-path/.*/",  
           "r|/dev/.*/by-id/.*/", "a/.*/" ]
```

DRBDとMPIOデバイスは受け入れ、その他のすべてのデバイスは拒否するフィルタ行は次のようになります。

```
filter = [ "a|/dev/drbd.*|", "a|/dev/.*/by-id/dm-uuid-mpath-.*/", "r/.*/"  
          ]
```

- 4 環境設定ファイルを書き込み、すべてのクラスタノードにコピーします。

14.4 詳細情報

詳細な情報は、<http://www.clusterlabs.org/wiki/Help:Contents>にあるPacemakerメーリングリストから取得できます。

cLVMのFAQのオフィシャルサイトは<http://sources.redhat.com/cluster/wiki/FAQ/CLVM>です。

ストレージ保護

High Availability クラスタスタックでは、データの整合性の保護が最優先されます。これは、データストレージへの未調整の同時アクセスを防止することによって達成されます。たとえば、**ext3** ファイルは、クラスタに一回だけマウントされ、**OCFS2** ボリュームは、他のクラスタノードとの調整が可能になるまでマウントされません。正常に機能するクラスタでは、**Pacemaker** によって、リソースがそれらの同時並行性の制限を超えてアクティブであるかどうかを検出され、復元が開始されます。さらに、そのポリシーエンジンがそれらの制限を超えることは決してありません。

ただし、ネットワークのパーティション分割やソフトウェアの誤動作により、いくつかのコーディネータが選択される状況となる可能性があります。このいわゆるスプリットブレインシナリオが発生した場合は、データが破損することがあります。したがって、このリスクを軽減するため、クラスタスタックには、いくつかの保護レイヤが追加されています。

この目的に貢献する第一のコンポーネントは、**IO** フェンシング/**STONITH** です。これにより、ストレージアクティベーション以前の他のすべてのアクセスが確実に終了されるからです。その他のメカニズムとしては、管理やアプリケーションの欠陥に対してシステムを保護する **cLVM2** の排他的アクティベーションや **OCFS2** のファイルロッキングサポートがあります。これらをご使用のセットアップに適合するように組み合わせると、スプリットブレインシナリオの悪影響を確実に防止できます。

この章では、ストレージ自体を活用する **IO** フェンシングについて説明し、次に、排他的ストレージアクセスを確保する追加保護レイヤについて説明します。これら2つのメカニズムを組み合わせると、より高度な保護を実現できます。

15.1 ストレージベースのフェンシング

SBD (Split Brain Detector)、watchdogサポート、およびexternal/sbd STONITHエージェントの使用により、スプリットブレインシナリオを確実に回避できます。

15.1.1 概要

すべてのノードが共有ストレージへのアクセスを持つ環境で、小さなパーティション(1MB)をSBDでできるようにフォーマットします。SBDは、そのデーモンの設定後、クラスタスタックの他のコンポーネントが起動される前に各ノードでオンラインになります。SBDデーモンは、他のすべてのクラスタコンポーネントがシャットダウンされた後に終了されます。したがって、クラスタリソースがSBDの監督なしでアクティブになることはありません。

このデーモンは、自動的に、パーティション上のメッセージスロットの1つを自分自身に割り当て、自分へのメッセージがないかどうか、そのスロットを絶えず監視します。デーモンは、メッセージを受信すると、ただちに要求に従います(フェンシングのための電源切断やリブートサイクルの開始など)。

デーモンは、ストレージデバイスへの接続を絶えず監視し、パーティションが到達不能になった場合は、デーモン自体が終了します。このため、デーモンがフェンシングメッセージから切断されることはありません。これは、クラスタデータが別のパーティション上の同じ論理ユニットにある場合、追加障害ポイントになることはありません。ストレージ接続を失えば、ワークロードは終了します。

保護は、watchdogサポートによって増大します。最近のシステムでは、hardware watchdogをサポートしています。この機能は、ソフトウェアクライアントによって更新される必要があり、更新されない場合は、ハードウェアがシステムの再起動を強制します。この機能は、SBDプロセス自体の障害(IOエラーで終了またはスタックするなど)に対する保護を提供します。

15.1.2 ストレージベースの保護のセットアップ

ストレージベースの保護を設定するには、次の手順に従う必要があります。

- 1 SBDパーティションの作成 (197 ページ)
- 2 ソフトウェアウォッチドッグのセットアップ (199 ページ)
- 3 SBDデーモンの起動 (199 ページ)
- 4 SBDのテスト (200 ページ)
- 5 フェンシングリソースの設定 (200 ページ)

次の手順は、すべて、`root`として実行する必要があります。手順を開始する前に、次の要件が満たされているかどうか確認してください。

重要項目: 要件

- 環境内にすべてのノードが到達できる共有ストレージが存在する必要があります。
- 共有ストレージセグメントが、ホストベースのRAID、cLVM2、DRBDを使用してはなりません。
- ただし、信頼性向上のため、ストレージベースのRAIDとマルチパスの使用は推奨されます。

SBDパーティションの作成

デバイスの起動時に1MBのパーティションを作成することを推奨します。SBDデバイスがマルチパスグループ上にある場合は、MPIOのバスダウン検出によって遅延が発生することがあるので、SBDが使用するタイムアウトを調整する必要があります。msgwaitタイムアウト後、メッセージがノードに配信されたと想定されます。この時間は、マルチパスの場合、MPIOがパスの障害を検出し、次のパスに切り替えるために必要な時間です。これは、ご使用の

環境でテストする必要があるかもしれません。ノードは、ウォッチドッグタイムを十分な速度で更新しなかった場合は、ノード自体が終了します。ウォッチドッグタイムアウトは、msgwaitタイムアウトより短くする必要があります。半分ぐらいの値が適当です。

次の手順では、このSBDパーティションを/dev/SBDで参照します。これは、ご使用の実際のパス名で置き換えてください(たとえば、/dev/sdc1)。

重要項目: 既存データの上書き

SBD用に使用するデバイスには、何もデータがないようにしてください。sdbコマンドは、さらに確認を要求せずに、デバイスを上書きします。

- 1 次のコマンドで、SBDデバイスを初期化します。

```
sbd -d /dev/SBD create
```

これによって、デバイスにヘッダが書き込まれ、デフォルトのタイミングでこのデバイスを共有する最大255ノードのスロットが作成されます。

- 2 SBDデバイスがマルチパスグループ上にある場合は、SBDが使用するタイムアウトを調整します。これは、SBDデバイスの初期化時に指定できます(すべてのタイムアウトは秒単位で指定)。

```
/usr/sbin/sbd -d /dev/SBD -4 $msgwait -1 $watchdogtimeout create
```

- 3 次のコマンドで、デバイスに何が書き込まれたか確認します。

```
sbd -d /dev/SBD dump
Header version      : 2
Number of slots     : 255
Sector size        : 512
Timeout (watchdog)  : 5
Timeout (allocate)  : 2
Timeout (loop)      : 1
Timeout (msgwait)   : 10
```

ご覧のように、タイムアウトがヘッダにも保存され、それらに関するすべての参加ノードの合意が確保されます。

ソフトウェアウォッチドッグのセットアップ

ご使用のLinuxシステムにウォッチドッグを使用するように設定することを強く推奨します。これには、システムブート時に正しいウォッチドッグドライバをロードする必要があります。

- HPハードウェアでは、これは、hpwdtモジュールで行います。
- Intel TCOを含むシステムでは、iTCO_wdtを使用できます。softdogが最も一般的なドライバですが、実際のハードウェア統合のあるドライバの使用を推奨します。

選択リストについては、カーネルパッケージ内のdrivers/watchdogを参照してください。

SBDデーモンの起動

SBDデーモンは、クラスタスタックの不可欠なコンポーネントです。このデーモンは、クラスタスタックの実行中や、あるいはクラスタスタックの一部がクラッシュしたときでも、実行している必要があります。そうすれば、クラスタスタックをフェンシングできます。

- 1 OpenAIS initスクリプトにSDBを開始/停止させるには、次のコマンドを/etc/sysconfig/sbdに追加します。

```
SBD_DEVICE="/dev/SBD"
# The next line enables the watchdog support:
SBD_OPTS="-W"
```

SBDデバイスがアクセス不能な場合は、SBDデーモンが開始できなくなり、OpenAISの起動を抑止します。

注記

SBDデバイスがノードからアクセスできなくなった場合は、ノードが無限のリブートサイクルに入ることがあります。これは、技術的には正しい結果ですが、ご使用の管理ポリシーによっては、問題とみなされることがあります。そのような場合は、ブート時にOpenAISを自動的に起動したくないことがあります。

- 2 先に進む前に、`rcopenais restart`を実行して、SBDがすべてのノード上で開始しているようにします。

SBDのテスト

- 1 次のコマンドを使用すると、ノードスロットとそれらの現在のメッセージがSBDデバイスからダンプされます。

```
sbd -d /dev/SBD list
```

ここに、SBDとともに起動されたすべてのクラスタノードが一覧され、メッセージスロットにはclearが表示されるはずです。

- 2 ノードの1つにテストメッセージを送信してみます。

```
sbd -d /dev/SBD message nodea test
```

- 3 ノードがシステムログにメッセージの受信を記録します。

```
Aug 29 14:10:00 nodea sbd: [13412]: info: Received command test from nodeb
```

これによって、SBDがノード上で実際に機能し、メッセージを受信できることが確認されます。

フェンシングリソースの設定

- 1 SBDの設定を完了するには、次のように、SBDをCIB内でSTONITH/フェンシングメカニズムとしてアクティブにする必要があります。

```
crm configure
crm(live)configure# property stonith-enabled="true"
crm(live)configure# property stonith-timeout="30s"
crm(live)configure# primitive stonith:external/sbd params
sbd_device="/dev/SBD"
crm(live)configure# commit
crm(live)configure# quit
```

ノードスロットは自動的に割り当てられるので、手動ホットリストを定義する必要はありません。

- 2 以前設定した他のフェンシングデバイスを無効にします。現在、この機能にはSBDメカニズムが使用されるからです。

一度リソースが開始すれば、クラススタは共有ストレージフェンシング用に正常に設定されています。クラススタは、ノードのフェンシングが必要になると、この方法を使用します。

15.2 排他的ストレージアクティベーションの確保

このセクションでは、共有ストレージへのアクセスを1つのノードに排他的にロックする低レベルの追加メカニズムであるsfexを紹介します。ただし、sfexは、STONITHに置き換わらないので注意してください。sfexには共有ストレージが必要なので、上記で説明したexternal/sbdフェンシングメカニズムは、ストレージの別のパーティションでを使用することをお勧めします。

設計上、sfexは、同時並行性を必要とするワークロード(OCFS2など)とともに使用することはできませんが、従来のフェールオーバースタイルのワークロードの保護レイヤとして機能します。これは、実際にはSCSI-2予約と似ていますが、もっと一般的です。

15.2.1 概要

共有ストレージ環境では、ストレージの小さなパーティションが1つ以上のロックの保存用に確保されます。

ノードは、保護されたリソースを取得する前に、まず、保護ロックを取得する必要があります。順序は、Pacemakerによって強制され、sfexコンポーネントは、Pacemakerがスプリットブレイン条件に制約されても、ロックが2回以上付与されないことを保証します。

ノードのダウンが永続的にロックをブロックせず、他のノードが続行できるように、これらのロックも定期的に更新される必要があります。

15.2.2 設定

次の手順では、sfexで使用する共有パーティションの作成方法と、CIBでsfexロック用にリソースを設定する方法を学習します。1つのsfexパーティション

は任意の数のロックを保持できますが、デフォルトは1に設定されています。ロックごとに1KBのストレージスペースを割り当てる必要があります。

重要項目: 要件

- **sfex**用の共有パーティションは、保護するデータと同じ論理ユニットにある必要があります。
- 共有された**sfex**パーティションは、ホストベースの**RAID**や**DRBD**を使用してはなりません。
- **cLVM2**論理ボリュームの使用は可能です。

手順 15.1 *sfex*パーティションを作成する

- 1 **sfex**で使用する共有パーティションを作成します。このパーティションの名前を書き留め、以降の手順の/dev/**sfex**をこの名前で置き換えます。

- 2 次のコマンドで、**sfex**メタデータを作成します。

```
sfex_init -i 1 /dev/sfex
```

- 3 メタデータが正しく作成されたかどうか確認します。

```
sfex_stats -i 1 /dev/sfex ; echo $?
```

現在、ロックがかかっていないので、このコマンドは、2を返すはずで
す。

手順 15.2 *sfex*ロック用リソースを設定する

- 1 **sfex**ロックは、**CIB**内のリソースを介して表現され、次のように設定されます。

```
primitive sfex_1 ocf:heartbeat:sfex \  
# params device="/dev/sfex" index="1" collision_timeout="1" \  
lock_timeout="70" monitor_interval="10" \  
# op monitor interval="10s" timeout="30s" on_fail="fence"
```

- 2** **sfex**ロックによってリソースを保護するには、保護対象と**sfex**リソース間の必須の順序付けと配置の制約を作成します。保護対象のリソースが **filesystem1** というIDを持つ場合は、次のようになります。

```
# order order-sfex-1 inf: sfex_1 filesystem1
# colocation colo-sfex-1 inf: filesystem1 sfex_1
```

- 3** グループ構文を使用する場合は、**sfex**リソースを最初のリソースとしてグループに追加します。

```
# group LAMP sfex_1 filesystem1 apache ipaddr
```


Sambaクラスタリング

クラススタ対応のSambaサーバは、異種混合ネットワークにHigh Availabilityソリューションを提供します。この章では、背景情報とクラススタ対応Sambaサーバの設定方法を説明します。

16.1 概念の概要

TDB (Trivial Databas)は、長年に渡って、Sambaによって使用されてきました。TDBでは、複数のアプリケーションを同時に書き込むことができます。すべての書き込み操作を正常に実行し、互いに衝突させないため、TDBは、内部ロッキングメカニズムを使用しています。

CTDB(Cluster Trivial Database)は、既存TDBの小規模な拡張です。CTDBは、プロジェクト自身によって、「一時データの保存のために、Sambaなどのプロジェクトによって使用されるTDBデータベースのクラスタ実装」として説明されています。

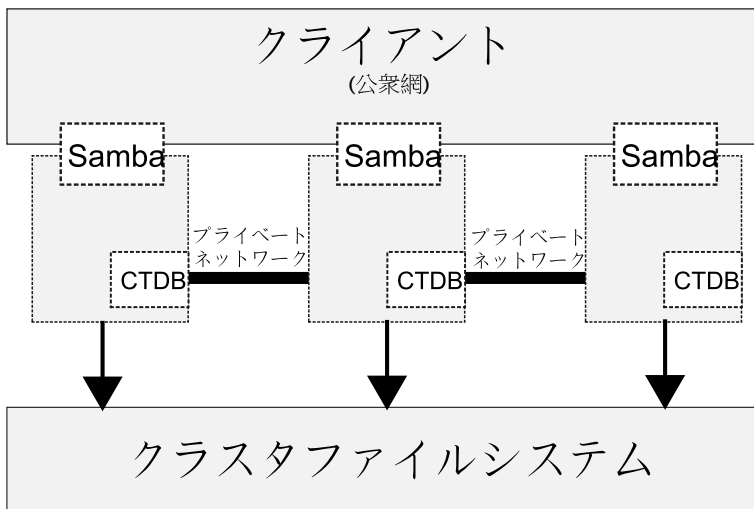
各クラスタノードは、ローカルCTDBデーモンを実行します。Sambaは、そのTDBに直接書き込むのではなく、そのローカルCTDBデーモンと通信します。それらのデーモンは、ネットワークを介してメタデータを交換しますが、実際の読み取り/書き込み操作は、高速ストレージでローカルコピー上で行われます。CTDBの概念は、図16.1「CTDBクラスタの構造」(206 ページ)に表示されています。

注記: Samba専用CTDB

CTDBリソースエージェントの現在の実装では、Sambaの管理のためだけにCTDBを設定します。他の機能(IPフェールオーバーなど)はすべて、Pacemakerで設定する必要があります。

さらに、CTDBは、完全に同種のクラスタに関してのみサポートされます。たとえば、クラスタのすべてのノードが同じアーキテクチャを持つ必要があります。i586とx86_64を混合できません。

図 16.1 CTDBクラスタの構造



クラスタ対応Sambaサーバは、一定のデータを共有する必要があります。

- UnixのユーザとグループIDをWindowsのユーザとグループに関連付けるマッピングテーブル。
- ユーザデータベースをすべてのノード間で同期する必要があります。
- Windowsドメイン内のメンバサーバの参加情報をすべてのノードで利用できる必要があります。
- メタデータ(アクティブSMBセッション、共有接続、各種ロックなど)をすべてのノードで利用できる必要があります。

N+1ノードを持つクラスタ対応SambaサーバがNノードだけのサーバより高速になることを目的としています。1つのノードは、クラスタ非対応のSambaサーバより遅くなることはありません。

16.2 基本的な設定

注記: 変更された設定ファイル

CTDBリソースエージェントは、自動的に、`/etc/sysconfig/ctdb`と`/etc/samba/smb.conf`を変更します。`crminfo` CTDBを使用して、CTDBリソースに指定できるすべてのパラメータを一覧してください。

クラスタ対応Sambaサーバをセットアップするには、次の手順に従います。

1 クラスタを準備します。

1a 本書のパートII「設定および管理」(35 ページ)で説明されているように、クラスタを設定します(`OpenAIS`、`Pacemaker`、`OCFS2`など使用)。

1b `OCFS2`などの共有ファイルシステムを設定し、マウントします(たとえば、`/shared`にマウント)。

1c `POSIX ACL`をオンにする場合は、それを有効にします。

- 新しい`OCFS2`ファイルシステムの場合は、次のコマンドを使用します。

```
mkfs.ocfs2 --fs-features=xattr ...
```

- 既存の`OCFS2`ファイルシステムの場合は、次のコマンドを使用します。

```
tuneefs.ocfs2 --fs-feature=xattr DEVICE
```

ファイルシステムリソースには、必ず、`acl`オプションを指定します。次のように、`crm`シェルを使用します。

```
crm(live)configure# primary ocfs2-3 ocf:heartbeat:Filesystem
options="acl" ...
```

- 1d** ctdb、smb、nmb、winbindの各サービスが無効になるようにします。

```
chkconfig ctdb off
chkconfig smb off
chkconfig nmb off
chkconfig winbind off
```

- 2** 共有ファイルシステムにCTDBロックとSamba状態のディレクトリを作成します。

```
mkdir -p /shared/samba/private
```

- 3** /etc/ctdb/nodesに、クラスタ内の各ノードの全プライベートIPアドレスを含むすべてのノードを挿入します。

```
192.168.1.10
192.168.1.11
```

- 4** CTDBリソースをクラスタに追加します。

```
crm configure
crm(live)configure# primitive ctdb ocf:heartbeat:CTDB params \
    ctdb_recovery_lock="/shared/samba/ctdb.lock" \
    smb_private_dir="/shared/samba/private" \
    op monitor timeout=20 interval=10
crm(live)configure# clone ctdb-clone ctdb \
    meta globally-unique="false" interleave="true"
crm(live)configure# colocation ctdb-with-fs inf: ctdb-clone fs-clone
crm(live)configure# order start-ctdb-after-fs inf: fs-clone ctdb-clone
crm(live)configure# commit
```

- 5** クラスタ対応のIPアドレスを追加します。

```
crm(live)configure# primitive ip ocf:heartbeat:IPAddr2 params
ip=192.168.2.222 \
    clusterip_hash="sourceip-sourceport" op monitor interval=60s
crm(live)configure# clone ip-clone ip meta globally-unique="true"
crm(live)configure# colocation ip-with-ctdb inf: ip-clone ctdb-clone
crm(live)configure# order start-ip-after-ctdb inf: ctdb-clone ip-clone
crm(live)configure# commit
```

- 6** 結果を確認します。

```
crm status
Clone Set: dlm-clone
    Started: [ hex-14 hex-13 ]
```

```

Clone Set: o2cb-clone
  Started: [ hex-14 hex-13 ]
Clone Set: c-ocfs2-3
  Started: [ hex-14 hex-13 ]
Clone Set: ctdb-clone
  Started: [ hex-14 hex-13 ]
Clone Set: ip-clone (unique)
  ip:0      (ocf::heartbeat:IPaddr2):      Started hex-13
  ip:1      (ocf::heartbeat:IPaddr2):      Started hex-14

```

- 7 クライアントコンピュータからテストを行います。次のコマンドをLinuxクライアントで実行して、システムからファイルをコピーしたり、システムにファイルをコピーできるかどうか確認します。

```
smbclient //192.168.2.222/myshare
```

16.3 クラスタ対応Sambaのデバッグとテスト

クライアント対応Sambaサーバのデバッグには、次のツールを使用できます。これらのツールは、さまざまなレベルで動作します。

ctdb_diagnostics

このツールを実行して、クラスタ対応Sambaサーバを診断します。これによって、問題の追跡に役立つ多数のデバッグメッセージが表示されます。

ctdb_diagnosticsコマンドは、次のファイルを検索します。これらのファイルは、すべてのノードで利用する必要があります。

```

/etc/krb5.conf
/etc/hosts
/etc/ctdb/nodes
/etc/sysconfig/ctdb
/etc/resolv.conf
/etc/nsswitch.conf
/etc/sysctl.conf
/etc/samba/smb.conf
/etc/fstab
/etc/multipath.conf
/etc/pam.d/system-auth
/etc/sysconfig/nfs
/etc/exports
/etc/vsftpd/vsftpd.conf

```

/etc/ctdb/public_addressesファイルと/etc/ctdb/static-routesファイルが存在する場合は、それらもチェックされます。

ping_pong

ping_pongでは、ファイルシステムがCTDBに適合しているかどうかチェックできます。このコマンドは、クラスタファイルシステムの一定のテスト(コヒーレンスやパフォーマンスなどのテスト)を実行して(http://wiki.samba.org/index.php/Ping_pong参照)、高負荷の状況下におけるクラスタの動作を示す情報を提供します。

クラスタファイルシステムの特定の側面をテストするには、次の手順に従います。

手順 16.1 クラスタファイルシステムのコヒーレンスとパフォーマンスをテストする

- 1 1つのノードでping_pongコマンドを開始します。ブレースホルダ N はノード数+1で置き換えます。ファイル名は、共有ストレージ内で利用可能なので、すべてのノードでアクセスできます。

```
ping_pong data.txt N
```

1つのノードでだけ実行しているので、ロッキングレートは非常に高いと予想してください。プログラムがロッキングレートを印刷しない場合は、クラスタファイルシステムを置き換えます。

- 2 同じパラメータを使用して、別のノードでping_pongの2つ目のコピーを開始します。

ロッキングレートが大幅に下がることを予想してください。使用のクラスタファイルシステムに次のどれかが当てはまる場合は、クラスタファイルシステムを置き換えます。

- ping_pongがロッキングレート(秒単位)を印刷しません。
- 2つのインスタンスのロッキングレートがほぼ同じではありません。
- 2つ目のインスタンスの開始後にロッキングレートが下がりました。

- 3 ping_pongの3つ目のコピーを開始します。もう1つノードを追加し、ロッキングレートの変化に注目します。
- 4 ping_pongコマンドを段階的に終了します。単一ノードの状態に戻るまで、ロッキングレートの増加が観察されるはずです。予想どおりの動作が起こらなかった場合は、クラスタファイルシステムを置き換えます。

16.4 詳細情報

- [http://linux-ha.org/wiki/CTDB_\(resource_agent\)](http://linux-ha.org/wiki/CTDB_(resource_agent))
- http://wiki.samba.org/index.php/CTDB_Setup
- <http://ctdb.samba.org>
- http://wiki.samba.org/index.php/Samba_%26_Clustering

パート IV. トラブルシューティング と参照情報

トラブルシューティング

しばしば、容易に理解しがたい奇妙な問題が発生することがあります(特に、High Availabilityで実験を開始した場合)。ただし、High Availabilityの内部プロセスを詳しく調べるために使用できる可能性のあるいくつかのユーティリティがあります。この章では、さまざまなソリューションを推奨します。

17.1 インストールの問題

パッケージのインストールやクラスタのオンライン化では、次のように問題をトラブルシュートします。

HAパッケージはインストールされているか

クラスタの構成を管理に必要なパッケージは、High Availability Extensionで利用できる高可用性インストールパターンに付属しています。

High Availability Extensionが各クラスタノードにアドオンとしてSUSE Linux Enterprise Server 11 SP1にインストールされているか、[高可用性] パターンが3.1項「High Availability Extensionのインストール」(21 ページ)で説明するように各マシンにインストールされているか、確認します。

初期構成がすべてのクラスタノードについて同一か

相互に通信するため、3.2項「クラスタの初期セットアップ」(22 ページ)で説明するように、同じクラスタに属するすべてのノードは同じbindnetaddr、mcastaddr、mcastportを使用する必要があります。

/etc/corosync/corosync.confで設定されている通信チャネルとオプションがすべてのクラスタノードに関して同一かどうか確認します。

暗号化通信を使用する場合は、/etc/corosync/authkeyファイルがすべてのクラスタノードで使用可能かどうかを確認します。

すべてのcorosync.conf設定(nodeid以外)が同一で、すべてのノードのauthkeyファイルが同一でなければなりません。

ファイアウォールでmcastportによる通信が許可されているか
クラスタノード間の通信に使用されるmcastportがファイアウォールでブロックされている場合、ノードは相互に認識できません。3.1項「High Availability Extensionのインストール」(21 ページ)で示すように、YaSTで初期セットアップを構成しているときに、ファイアウォール設定は通常、自動的に調整されます。

mcastportがファイアウォールでブロックされないようにするには、各ノードの/etc/sysconfig/SuSEfirewall2の設定を確認します。または、各クラスタノードのYaSTファイアウォールモジュールを起動します。[許可されるサービス] > [詳細] をクリックして、mcastportを許可された[UDPポート] のリストに追加し、変更を確定します。

OpenAISが各クラスタノードで起動されているか
各クラスタノードのOpenAISの状態を/etc/init.d/openais statusで確認します。OpenAISが実行されていない場合、/etc/init.d/openais startを実行して起動します。

17.2 HAクラスタの「デバッグ」

次のコマンドは、リソース操作履歴(-oオプション)と非アクティブなリソース(-r)を表示します。

```
crm_mon -o -r
```

表示内容は、ステータスが変わると、更新されます(これをキャンセルするには、Ctrl + Cを押します)。次に例を示します

例 17.1 停止されたリソース

Refresh in 10s...

```
=====
Last updated: Mon Jan 19 08:56:14 2009
Current DC: d42 (d42)
3 Nodes configured.
3 Resources configured.
=====

Online: [ d230 d42 ]
OFFLINE: [ clusternode-1 ]

Full list of resources:

Clone Set: o2cb-clone
    Stopped: [ o2cb:0 o2cb:1o2cb:2 ]
Clone Set: dlm-clone
    Stopped [ dlm:0 dlm:1 dlm:2 ]
mySecondIP      (ocf::heartbeat:IPaddr):      Stopped

Operations:
* Node d230:
    aa: migration-threshold=1000000
    + (5) probe: rc=0 (ok)
    + (37) stop: rc=0 (ok)
    + (38) start: rc=0 (ok)
    + (39) monitor: interval=15000ms rc=0 (ok)
* Node d42:
    aa: migration-threshold=1000000
    + (3) probe: rc=0 (ok)
    + (12) stop: rc=0 (ok)
```

まず、ノードをオンラインにします(17.3項 (218 ページ)を参照)。その後、リソースと操作を確認します。

『*Configuration Explained*』のPDF(<http://clusterlabs.org/wiki/Documentation>)では、「*How Does the Cluster Interpret the OCF Return Codes?*」セクションで3つの異なる復元タイプを説明しています。

17.3 FAQ

クラスタはどのような状態でしょうか

クラスタの現在の状態を確認するには、`crm_mon`か`crmstatus`のどちらかを使用します。これによって、現在のDCと、現在のノードに認識されているすべてのノードとリソースが表示されます。

クラスタのノードの一部が互いに認識しません。

これにはいくつかの理由が考えられます。

- まず設定ファイル`/etc/corosync/corosync.conf`を調べて、マルチキャストアドレスがクラスタ内のすべてのノードで同一かどうか確認します(キー`mcastaddr`を含む`interface`セクションを調べてください)。
- ファイアウォール設定を確認します。
- スイッチがマルチキャストアドレスをサポートしているか確認します。
- ノード間の接続が切断されてるかどうか確認します。これはファイアウォールの構成が正しくないことが大半の原因です。また、これはスプリットブレインの理由にもなり、クラスタがパーティション化されます。

現在わかっているリソースを一覧表示したい。

コマンド`crm_resource -L`を使用して、現在のリソースの情報を取得できます。

リソースを構成しましたが、いつも失敗します。

OCFスクリプトを確認するには、たとえば、次の`ocf-tester`コマンドを使用します。

```
ocf-tester -n ipl -o ip=YOUR_IP_ADDRESS \  
/usr/lib/ocf/resource.d/heartbeat/IPaddr
```

パラメータを増やすには、`-o`を何回も使用します。必須パラメータとオプションパラメータのリストは、`crm ra info AGENT`の実行によって取得できます。たとえば、次のようにします。

```
crm ra info ocf:heartbeat:IPaddr
```

ocf-testerを実行する場合は、その前に、リソースがクラスタで管理されていないことを確認してください。

エラーメッセージを受け取りました。詳細情報を取得できますか。

コマンドには、いつでも、`--verbose`パラメータを追加できます。これを複数回行くと、デバッグ出力の情報量が増加します。は、`/var/log/messages`に、役に立つヒントがあります。

リソースはどのようにクリーンアップしますか。

次のコマンドを使用してください。

```
crm resource list
crm resource cleanup rsoid [node]
```

ノードを指定しないと、すべてのノードでリソースがクリーンアップされます。詳細については、6.4.2項「リソースのクリーンアップ」(116 ページ)を参照してください。

ocfs2デバイスをマウントできません。

`/var/log/message`に次の行があるかどうか確認してください。

```
Jan 12 09:58:55 clusternode2 lrmd: [3487]: info: RA output:
(o2cb:l:start:stderr) 2009/01/12_09:58:55
    ERROR: Could not load ocfs2_stackglue
Jan 12 16:04:22 clusternode2 modprobe: FATAL: Module ocfs2_stackglue not
found.
```

この場合、カーネルモジュール`ocfs2_stackglue.ko`がありません。インストールされたカーネルに応じて、パッケージ`ocfs2-kmp-default`、`ocfs2-kmp-pae`、または`ocfs2-kmp-xen`をインストールします。

17.4 その他の情報

LinuxおよびHeartbeatの、クラスタリソースの設定、およびHeartbeatクラスタの管理とカスタマイズなど、高可用性に関するその他の情報については、<http://clusterlabs.org/wiki/Documentation>を参照してください。

クラスタ管理ツール

High Availability Extensionには、クラスタをコマンドラインから管理する際に役立つ、包括的なツールセットが付属しています。この章では、CIBおよびクラスタリソースでのクラスタ構成を管理するために必要なツールを紹介します。リソースエージェントを管理する他のコマンドラインツールや、セットアップのデバッグ(およびトラブルシューティング)に使用するツールについては、第17章 **トラブルシューティング**(215 ページ)で説明されています。

次のリストは、クラスタ管理に関連するいくつかのタスクを示しており、これらのタスクを実行するために使用するツールを簡単に説明しています。

クラスタの状態の監視

`crm_mon`コマンドでは、クラスタの状態と構成を監視できます。出力には、ノード数、`uname`、`uuid`、状態、クラスタで構成されたリソース、それぞれの現在の状態が含まれます。`crm_mon`の出力は、コンソールに表示したり、HTMLファイルに出力したりできます。`status`セクションのないクラスタ設定ファイルが指定された場合、`crm_mon`はファイルに指定されたノードとリソースの概要を作成します。このツールの使用方法とコマンド構文の詳細については、`crm_mon(8)` (244 ページ)を参照してください。

CIBの管理

`cibadmin`コマンドは、Heartbeat CIBを操作するための低レベル管理コマンドです。CIBのすべてまたは一部のダンプ、CIBのすべてまたは一部の更新、すべてまたは一部の変更、CIB全体の削除、その他のCIB管理操作に使用できます。このツールの使用方法とコマンド構文の詳細については、`cibadmin(8)` (224 ページ)を参照してください。

設定の変更の管理

`crm_diff` コマンドは、XMLパッチの作成と適用をサポートします。クラスタ設定の2つのバージョン間の差異を表示する、または後で `cibadmin(8)` (224 ページ) を使用して適用できるように変更を保存する場合に有効です。このツールの使用方法とコマンド構文の詳細については、`crm_diff(8)` (236 ページ) を参照してください。

CIB属性の操作

`crm_attribute` コマンドで、CIBで使用されているノード属性およびクラスタ構成オプションを問い合わせる操作ができます。このツールの使用方法とコマンド構文の詳細については、`crm_attribute(8)` (233 ページ) を参照してください。

クラスタ構成の検証

`crm_verify` コマンドは、構成データベース(CIB)の整合性およびその他の問題を確認します。構成を含むファイルを確認したり、実行中のクラスタに接続したりできます。2種類の問題を報告します。エラーは、解決しないと `Heartbeat` が正常に機能できず、警告の解決は管理者が担当します。`cm_verify` は新規または変更された構成の作成を支援します。実行中のクラスタのCIBのローカルコピーを作成し、編集し、`crm_verify` を使用して検証し、新規構成を `cibadmin` を使用して適用できます。このツールの使用方法とコマンド構文の詳細については、`crm_verify(8)` (272 ページ) を参照してください。

リソース構成の管理

`crm_resource` コマンドは、クラスタ上でリソース関連のさまざまなアクションを実行します。構成されたリソースの定義の変更、リソースの開始と停止、リソースの削除およびノード間での移行を実行できます。このツールの使用方法とコマンド構文の詳細については、`crm_resource(8)` (248 ページ) を参照してください。

リソースの失敗回数の管理

`crm_failcount` コマンドは、所定のノードのリソースごとの失敗回数を問い合わせます。このツールは、失敗回数のリセットにも使用でき、リソースが頻繁に失敗したノード上で再度実行できるようにします。このツールの使用方法とコマンド構文の詳細については、`crm_failcount(8)` (239 ページ) を参照してください。

ノードのスタンバイ状態の管理

`crm_standby` コマンドは、ノードのスタンバイ属性を操作します。スタンバイモードのノードはすべて、リソースをホストすることができず、そのノードにあるリソースは削除する必要があります。スタンバイモードはカーネルアップデートなどの保守タスクに有効です。ノードを再びクラスタの完全にアクティブなメンバーにするには、ノードからスタンバイ属性を削除します。このツールの使用方法とコマンド構文の詳細については、`crm_standby(8)` (269 ページ)を参照してください。

cibadmin (8)

cibadmin — Provides direct access to the cluster configuration

Synopsis

Allows the configuration, or sections of it, to be queried, modified, replaced and/or deleted.

```
cibadmin (--query|-Q) -[Vrwlsmfbp] [-i xml-object-id|-o
    xml-object-type] [-t t-flag-whatever] [-h hostname]

cibadmin (--create|-C) -[Vrwlsmfbp] [-X xml-string]
    [-x xml- filename] [-t t-flag-whatever] [-h hostname]

cibadmin (--replace|-R) -[Vrwlsmfbp] [-i xml-object-id|
    -o xml-object-type] [-X xml-string] [-x xml-filename] [-t
    t-flag- whatever] [-h hostname]

cibadmin (--update|-U) -[Vrwlsmfbp] [-i xml-object-id|
    -o xml-object-type] [-X xml-string] [-x xml-filename] [-t
    t-flag- whatever] [-h hostname]

cibadmin (--modify|-M) -[Vrwlsmfbp] [-i xml-object-id|
    -o xml-object-type] [-X xml-string] [-x xml-filename] [-t
    t-flag- whatever] [-h hostname]

cibadmin (--delete|-D) -[Vrwlsmfbp] [-i xml-object-id|
    -o xml-object-type] [-t t-flag-whatever] [-h hostname]

cibadmin (--delete_alt|-d) -[Vrwlsmfbp] -o
    xml-object-type [-X xml-string|-x xml-filename]
    [-t t-flag-whatever] [-h hostname]

cibadmin --erase (-E)

cibadmin --bump (-B)

cibadmin --ismaster (-m)

cibadmin --master (-w)

cibadmin --slave (-r)

cibadmin --sync (-S)

cibadmin --help (-?)
```

Description

The `cibadmin` command is the low-level administrative command for manipulating the Heartbeat CIB. Use it to dump all or part of the CIB, update all or part of it, modify all or part of it, delete the entire CIB, or perform miscellaneous CIB administrative operations.

`cibadmin` operates on the XML trees of the CIB, largely without knowledge of the purpose of the updates or queries performed. This means that shortcuts that seem natural to users who understand the meaning of the elements in the XML tree are impossible to use with `cibadmin`. It requires a complete lack of ambiguity and can only deal with valid XML subtrees (tags and elements) for both input and output.

注記

`cibadmin` should always be used in preference to editing the `cib.xml` file by hand—especially if the cluster is active. The cluster goes to great lengths to detect and discourage this practice so that your data is not lost or corrupted.

Options

`--obj_type object-type, -o object-type`

Specify the type of object on which to operate. Valid values are `nodes`, `resources`, `constraints`, `crm_status`, and `status`.

`--verbose, -V`

Turn on debug mode. Additional `-V` options increase the detail and frequency of the output.

`--help, -?`

Obtain a help message from `cibadmin`.

`--xpath PATHSPEC, -A PATHSPEC`

Supply a valid XPath to use instead of an `obj_type`.

Commands

`--bump, -B`

Increase the `epoch` version counter in the CIB. Normally this value is increased automatically by the cluster when a new leader is elected. Manually increasing it can be useful if you want to make an older configuration obsolete (such as one stored on inactive cluster nodes).

`--create, -C`

Create a new CIB from the XML content of the argument.

`--delete, -D`

Delete the first object matching the supplied criteria, for example, `<op id="rscl_op1" name="monitor"/>`. The tag name and all attributes must match in order for the element to be deleted

`--erase, -E`

Erase the contents of the entire CIB.

`--ismaster, -m`

Print a message indicating whether or not the local instance of the CIB software is the master instance or not. Exits with return code 0 if it is the master instance or 35 if not.

`--modify, -M`

Find the object somewhere in the CIB's XML tree and update it.

`--query, -Q`

Query a portion of the CIB.

`--replace, -R`

Recursively replace an XML object in the CIB.

`--sync, -S`

Force a resync of all nodes with the CIB on the specified host (if `-h` is used) or with the DC (if no `-h` option is used).

XML Data

`--xml-text string, -X string`

Specify an XML tag or fragment on which `crmadmin` should operate. It must be a complete tag or XML fragment.

`--xml-file filename, -x filename`

Specify the XML from a file on which `cibadmin` should operate. It must be a complete tag or an XML fragment.

`--xml_pipe, -p`

Specify that the XML on which `cibadmin` should operate comes from standard input. It must be a complete tag or an XML fragment.

Advanced Options

`--host hostname, -h hostname`

Send command to specified host. Applies to `query` and `sync` commands only.

`--local, -l`

Let a command take effect locally (rarely used, advanced option).

`--no-bcast, -b`

Command will not be broadcast even if it altered the CIB.

重要項目

Use this option with care to avoid ending up with a divergent cluster.

`--sync-call, -s`

Wait for call to complete before returning.

Examples

To get a copy of the entire active CIB (including status section, etc.) delivered to stdout, issue this command:

```
cibadmin -Q
```

To add an IPaddr2 resource to the *resources* section, first create a file `foo` with the following contents:

```
<primitive id="R_10.10.10.101" class="ocf" type="IPaddr2"
  provider="heartbeat">
  <instance_attributes id="RA_R_10.10.10.101">
    <attributes>
      <nvpair id="R_ip_P_ip" name="ip" value="10.10.10.101"/>
      <nvpair id="R_ip_P_nic" name="nic" value="eth0"/>
    </attributes>
  </instance_attributes>
</primitive>
```

Then issue the following command:

```
cibadmin --obj_type resources -U -x foo
```

To change the IP address of the IPaddr2 resource previously added, issue the command below:

```
cibadmin -M -X '<nvpair id="R_ip_P_ip" name="ip" value="10.10.10.102"/>'
```

注記

This does not change the resource name to match the new IP address. To do that, delete then re-add the resource with a new ID tag.

To stop (disable) the IP address resource added previously, and without removing it, create a file called `bar` with the following content in it:

```
<primitive id="R_10.10.10.101">
  <instance_attributes id="RA_R_10.10.10.101">
    <attributes>
      <nvpair id="stop_R_10.10.10.101" name="target-role" value="Stopped"/>
    </attributes>
  </instance_attributes>
</primitive>
```

Then issue the following command:

```
cibadmin --obj_type resources -U -x bar
```

To restart the IP address resource stopped by the previous step, issue:

```
cibadmin -D -X '<nvpair id="stop_R_10.10.10.101">'
```

To completely remove the IP address resource from the CIB, issue this command:


```
cibadmin -D -X '<primitive id="R_10.10.10.101"/>'
```

To replace the CIB with a new manually-edited version of the CIB, use the following command:

```
cibadmin -R -x $HOME/cib.xml
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk.

See Also

`crm_resource(8)` (248 ページ), `crmadmin(8)` (230 ページ), `lrmadmin(8)`, `heartbeat(8)`

Caveats

Avoid working on the automatically maintained copy of the CIB on the local disk. Whenever anything in the cluster changes, the CIB is updated. Therefore using an outdated backup copy of the CIB to propagate your configuration changes might result in an inconsistent cluster.

crmadmin (8)

crmadmin — controls the Cluster Resource Manager

Synopsis

```
crmadmin [-V|-q] [-i|-d|-K|-S|-E] node
crmadmin [-V|-q] -N -B
crmadmin [-V|-q] -D
crmadmin -v
crmadmin -?
```

Description

`crmadmin` was originally designed to control most of the actions of the CRM daemon. However, the largest part of its functionality has been made obsolete by other tools, such as `crm_attribute` and `crm_resource`. Its remaining functionality is mostly related to testing and the status of the `crmd` process.

警告

Some `crmadmin` options are geared towards testing and cause trouble if used incorrectly. In particular, do not use the `--kill` or `--election` options unless you know exactly what you are doing.

Options

`--help, -?`
Print the help text.

`--version, -v`
Print version details for HA, CRM, and CIB feature set.

`--verbose, -V`
Turn on command debug information.

注記

Increase the level of verbosity by providing additional instances.

`--quiet, -q`

Do not provide any debug information at all and reduce the output to a minimum.

`--bash-export, -B`

Create bash export entries of the form `export uname=uuid`. This applies only to the `crmadmin -N node` command.

注記

The `-B` functionality is rarely useful and may be removed in future versions.

Commands

`--debug_inc node, -i node`

Incrementally increase the CRM daemon's debug level on the specified node. This can also be achieved by sending the USR1 signal to the `crmd` process.

`--debug_dec node, -d node`

Incrementally decrease the CRM daemon's debug level on the specified node. This can also be achieved by sending the USR2 signal to the `crmd` process.

`--kill node, -K node`

Shut down the CRM daemon on the specified node.

警告

Use this with extreme caution. This action should normally only be issued by Heartbeat and may have unintended side effects.

`--status node, -S node`

Query the status of the CRM daemon on the specified node.

The output includes a general health indicator and the internal FSM state of the `crmd` process. This can be helpful when determining what the cluster is doing.

`--election node, -E node`

Initiate an election from the specified node.

警告

Use this with extreme caution. This action is normally initiated internally and may have unintended side effects.

`--dc_lookup, -D`

Query the uname of the current DC.

The location of the DC is only of significance to the `crmd` internally and is rarely useful to administrators except when deciding on which node to examine the logs.

`--nodes, -N`

Query the uname of all member nodes. The results of this query may include nodes in `offline` mode.

注記

The `-i`, `-d`, `-K`, and `-E` options are rarely used and may be removed in future versions.

See Also

`crm_attribute(8)` (233 ページ), `crm_resource(8)` (248 ページ)

crm_attribute (8)

crm_attribute — Allows node attributes and cluster options to be queried, modified and deleted

Synopsis

```
crm_attribute [options]
```

Description

The `crm_attribute` command queries and manipulates node attributes and cluster configuration options that are used in the CIB.

Options

`--help, -?`

Print a help message.

`--verbose, -V`

Turn on debug information.

注記

Increase the level of verbosity by providing additional instances.

`--quiet, -Q`

When doing an attribute query using `-G`, print just the value to stdout. Use this option with `-G`.

`--get-value, -G`

Retrieve, rather than set, the preference.

`--delete-attr, -D`

Delete, rather than set, the attribute.

`--attr-id string, -i string`
For advanced users only. Identifies the id attribute.

`--attr-value string, -v string`
Value to set. This is ignored when used with `-G`.

`--node node_name, -N node_name`
The uname of the node to change

`--set-name string, -s string`
Specify the set of attributes in which to read or write the attribute.

`--attr-name string, -n string`
Specify the attribute to set or query.

`--type string, -t type`
Determine to which section of the CIB the attribute should be set or to which section of the CIB the attribute that is queried belongs. Possible values are `nodes`, `status`, or `crm_config`.

Examples

Query the value of the `location` attribute in the `nodes` section for the host *myhost* in the CIB:

```
crm_attribute -G -t nodes -U myhost -n location
```

Query the value of the `cluster-delay` attribute in the `crm_config` section in the CIB:

```
crm_attribute -G -t crm_config -n cluster-delay
```

Query the value of the `cluster-delay` attribute in the `crm_config` section in the CIB. Print just the value:

```
crm_attribute -G -Q -t crm_config -n cluster-delay
```

Delete the `location` attribute for the host *myhost* from the `nodes` section of the CIB:

```
crm_attribute -D -t nodes -U myhost -n location
```

Add a new attribute called `location` with the value of `office` to the `set` subsection of the `nodes` section in the CIB (settings applied to the host *myhost*):

```
crm_attribute -t nodes -U myhost -s set -n location -v office
```

Change the value of the `location` attribute in the `nodes` section for the *myhost* host:

```
crm_attribute -t nodes -U myhost -n location -v backoffice
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk. Editing this file directly is strongly discouraged.

See Also

`cibadmin(8)` (224 ページ)

crm_diff (8)

`crm_diff` — identify changes to the cluster configuration and apply patches to the configuration files

Synopsis

```
crm_diff [-?|-V] [-o filename] [-O string] [-p filename] [-n filename] [-N string]
```

Description

The `crm_diff` command assists in creating and applying XML patches. This can be useful for visualizing the changes between two versions of the cluster configuration or saving changes so they can be applied at a later time using `cibadmin`.

Options

`--help, -?`

Print a help message.

`--original filename, -o filename`

Specify the original file against which to diff or apply patches.

`--new filename, -n filename`

Specify the name of the new file.

`--original-string string, -O string`

Specify the original string against which to diff or apply patches.

`--new-string string, -N string`

Specify the new string.

`--patch filename, -p filename`

Apply a patch to the original XML. Always use with `-o`.

`--cib, -c`

Compare or patch the inputs as a CIB. Always specify the base version with `-o` and provide either the patch file or the second version with `-p` or `-n`, respectively.

`--stdin, -s`

Read the inputs from stdin.

Examples

Use `crm_diff` to determine the differences between various CIB configuration files and to create patches. By means of patches, easily reuse configuration parts without having to use the `cibadmin` command on every single one of them.

- 1 Obtain the two different configuration files by running `cibadmin` on the two cluster setups to compare:

```
cibadmin -Q > cib1.xml
cibadmin -Q > cib2.xml
```

- 2 Determine whether to diff the entire files against each other or compare just a subset of the configurations.

- 3 To print the difference between the files to stdout, use the following command:

```
crm_diff -o cib1.xml -n cib2.xml
```

- 4 To print the difference between the files to a file and create a patch, use the following command:

```
crm_diff -o cib1.xml -n cib2.xml > patch.xml
```

- 5 Apply the patch to the original file:

```
crm_diff -o cib1.xml -p patch.xml
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk. Editing this file directly is strongly discouraged.

See Also

[cibadmin\(8\) \(224 ページ\)](#)

crm_failcount (8)

crm_failcount — Manage the counter recording each resource's failures

Synopsis

```
crm_failcount [-?|-V] -D -u|-U node -r resource
crm_failcount [-?|-V] -G -u|-U node -r resource
crm_failcount [-?|-V] -v string -u|-U node -r resource
```

Description

Heartbeat implements a sophisticated method to compute and force failover of a resource to another node in case that resource tends to fail on the current node. A resource carries a `resource-stickiness` attribute to determine how much it prefers to run on a certain node. It also carries a `migration-threshold` that determines the threshold at which the resource should failover to another node.

The `failcount` attribute is added to the resource and increased on resource monitoring failure. The value of `failcount` multiplied by the value of `migration-threshold` determines the *failover score* of this resource. If this number exceeds the preference set for this resource, the resource is moved to another node and not run again on the original node until the failure count is reset.

The `crm_failcount` command queries the number of failures per resource on a given node. This tool can also be used to reset the failcount, allowing the resource to run again on nodes where it had previously failed too many times.

Options

`--help, -?`
Print a help message.

`--verbose, -V`
Turn on debug information.

Increase the level of verbosity by providing additional instances.

`--quiet, -Q`

When doing an attribute query using `-G`, print just the value to stdout. Use this option with `-G`.

`--get-value, -G`

Retrieve rather than set the preference.

`--delete-attr, -D`

Specify the attribute to delete.

`--attr-id string, -i string`

For advanced users only. Identifies the id attribute.

`--attr-value string, -v string`

Specify the value to use. This option is ignored when used with `-G`.

`--node node_uname, -U node_uname`

Specify the uname of the node to change.

`--resource-id resource name, -r resource name`

Specify the name of the resource on which to operate.

Examples

Reset the failcount for the resource `my_rsc` on the node `node1`:

```
crm_failcount -D -U node1 -r my_rsc
```

Query the current failcount for the resource `my_rsc` on the node `node1`:

```
crm_failcount -G -U node1 -r my_rsc
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk. Editing this file directly is strongly discouraged.

See Also

`crm_attribute(8)` (233 ページ), `cibadmin(8)` (224 ページ), and the Linux High Availability FAQ Web site [http://www.linux-ha.org/v2/faq/forced_failover]

crm_master (8)

`crm_master` — Manage a master/slave resource's preference for being promoted on a given node

Synopsis

```
crm_master [-V|-Q] -D [-l lifetime]  
crm_master [-V|-Q] -G [-l lifetime]  
crm_master [-V|-Q] -v string [-l string]
```

Description

`crm_master` is called from inside the resource agent scripts to determine which resource instance should be promoted to master mode. It should never be used from the command line and is just a helper utility for the resource agents. RAs use `crm_master` to promote a particular instance to master mode or to remove this preference from it. By assigning a lifetime, determine whether this setting should survive a reboot of the node (set lifetime to `forever`) or whether it should not survive a reboot (set lifetime to `reboot`).

A resource agent needs to determine on which resource `crm_master` should operate. These queries must be handled inside the resource agent script. The actual calls of `crm_master` follow a syntax similar to those of the `crm_attribute` command.

Options

`--help, -?`
Print a help message.

`--verbose, -V`
Turn on debug information.

注記

Increase the level of verbosity by providing additional instances.

`--quiet, -Q`

When doing an attribute query using `-G`, print just the value to stdout. Use this option with `-G`.

`--get-value, -G`

Retrieve rather than set the preference to be promoted.

`--delete-attr, -D`

Delete rather than set the attribute.

`--attr-id string, -i string`

For advanced users only. Identifies the id attribute.

`--attr-value string, -v string`

Value to set. This is ignored when used with `-G`.

`--lifetime string, -l string`

Specify how long the preference lasts. Possible values are `reboot` or `forever`.

Environment Variables

`OCF_RESOURCE_INSTANCE`—the name of the resource instance

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk.

See Also

`cibadmin(8)` (224 ページ), `crm_attribute(8)` (233 ページ)

crm_mon (8)

crm_mon — monitor the cluster's status

Synopsis

```
crm_mon [-V] -d -pfilename -h filename
crm_mon [-V] [-l|-n|-r] -h filename
crm_mon [-V] [-n|-r] -X filename
crm_mon [-V] [-n|-r] -c|-l
crm_mon [-V] -i interval
crm_mon -?
```

Description

The `crm_mon` command allows you to monitor your cluster's status and configuration. Its output includes the number of nodes, uname, uuid, status, the resources configured in your cluster, and the current status of each. The output of `crm_mon` can be displayed at the console or printed into an HTML file. When provided with a cluster configuration file without the status section, `crm_mon` creates an overview of nodes and resources as specified in the file.

Options

`--help, -?`

Provide help.

`--verbose, -V`

Increase the debug output.

`--interval seconds, -i seconds`

Determine the update frequency. If `-i` is not specified, the default of 15 seconds is assumed.

`--group-by-node, -n`
Group resources by node.

`--inactive, -r`
Display inactive resources.

`--simple-status, -s`
Display the cluster status once as a simple one line output (suitable for nagios).

`--one-shot, -l`
Display the cluster status once on the console then exit (does not use ncurses).

`--as-html filename, -h filename`
Write the cluster's status to the specified file.

`--web-cgi, -w`
Web mode with output suitable for CGI.

`--daemonize, -d`
Run in the background as a daemon.

`--pid-file filename, -p filename`
Specify the daemon's pid file.

Examples

Display your cluster's status and get an updated listing every 15 seconds:

```
crm_mon
```

Display your cluster's status and get an updated listing after an interval specified by `-i`. If `-i` is not given, the default refresh interval of 15 seconds is assumed:

```
crm_mon -i interval[s]
```

Display your cluster's status on the console:

```
crm_mon -c
```

Display your cluster's status on the console just once then exit:

```
crm_mon -l
```

Display your cluster's status and group resources by node:

```
crm_mon -n
```

Display your cluster's status, group resources by node, and include inactive resources in the list:

```
crm_mon -n -r
```

Write your cluster's status to an HTML file:

```
crm_mon -h filename
```

Run `crm_mon` as a daemon in the background, specify the daemon's pid file for easier control of the daemon process, and create HTML output. This option allows you to constantly create HTML output that can be easily processed by other monitoring applications:

```
crm_mon -d -p filename -h filename
```

Display the cluster configuration laid out in an existing cluster configuration file (*filename*), group the resources by node, and include inactive resources. This command can be used for dry runs of a cluster configuration before rolling it out to a live cluster.

```
crm_mon -r -n -X filename
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk. Editing this file directly is strongly discouraged.

crm_node (8)

crm_node — Lists the members of a cluster

Synopsis

```
crm_node [-V] [-p|-e|-q]
```

Description

Lists the members of a cluster.

Options

- V
be verbose
- partition, -p
print the members of this partition
- epoch, -e
print the epoch this node joined the partition
- quorum, -q
print a 1 if our partition has quorum

crm_resource (8)

crm_resource — Perform tasks related to cluster resources

Synopsis

```
crm_resource [-?|-V|-S] -L|-Q|-W|-D|-C|-P|-p [options]
```

Description

The `crm_resource` command performs various resource-related actions on the cluster. It can modify the definition of configured resources, start and stop resources, and delete and migrate resources between nodes.

`--help, -?`

Print the help message.

`--verbose, -V`

Turn on debug information.

注記

Increase the level of verbosity by providing additional instances.

`--quiet, -Q`

Print only the value on stdout (for use with `-W`).

Commands

`--list, -L`

List all resources.

`--query-xml, -x`

Query a resource.

Requires: `-r`

`--locate, -W`
Locate a resource.

Requires: `-r`

`--migrate, -M`
Migrate a resource from its current location. Use `-N` to specify a destination.

If `-N` is not specified, the resource is forced to move by creating a rule for the current location and a score of `-INFINITY`.

注記

This prevents the resource from running on this node until the constraint is removed with `-U`.

Requires: `-r`, Optional: `-N, -f`

`--un-migrate, -U`
Remove all constraints created by `-M`

Requires: `-r`

`--delete, -D`
Delete a resource from the CIB.

Requires: `-r, -t`

`--cleanup, -C`
Delete a resource from the LRM.

Requires: `-r`. Optional: `-H`

`--reprobe, -P`
Recheck for resources started outside the CRM.

Optional: `-H`

`--refresh, -R`
Refresh the CIB from the LRM.

Optional: -H

`--set-parameter string, -p string`

Set the named parameter for a resource.

Requires: -r, -v. Optional: -i, -s, and --meta

`--get-parameter string, -g string`

Get the named parameter for a resource.

Requires: -r. Optional: -i, -s, and --meta

`--delete-parameter string, -d string`

Delete the named parameter for a resource.

Requires: -r. Optional: -i, and --meta

`--list-operations string, -O string`

List the active resource operations. Optionally filtered by resource, node, or both.

Optional: -N, -r

`--list-all-operations string, -o string`

List all resource operations. Optionally filtered by resource, node, or both. Optional:

-N, -r

Options

`--resource string, -r string`

Specify the resource ID.

`--resource-type string, -t string`

Specify the resource type (primitive, clone, group, etc.).

`--property-value string, -v string`

Specify the property value.

`--node string, -N string`

Specify the hostname.

`--meta`

Modify a resource's configuration option rather than one which is passed to the resource agent script. For use with `-p`, `-g` and `-d`.

`--lifetime string, -u string`

Lifespan of migration constraints.

`--force, -f`

Force the resource to move by creating a rule for the current location and a score of `-INFINITY`

This should be used if the resource's stickiness and constraint scores total more than `INFINITY` (currently 100,000).

注記

This prevents the resource from running on this node until the constraint is removed with `-U`.

`-s string`

(Advanced Use Only) Specify the ID of the `instance_attributes` object to change.

`-i string`

(Advanced Use Only) Specify the ID of the `nvpair` object to change or delete.

Examples

Listing all resources:

```
crm_resource -L
```

Checking where a resource is running (and if it is):

```
crm_resource -W -r my_first_ip
```

If the `my_first_ip` resource is running, the output of this command reveals the node on which it is running. If it is not running, the output shows this.

Start or stop a resource:

```
crm_resource -r my_first_ip -p target_role -v started
crm_resource -r my_first_ip -p target_role -v stopped
```

Query the definition of a resource:

```
crm_resource -Q -r my_first_ip
```

Migrate a resource away from its current location:

```
crm_resource -M -r my_first_ip
```

Migrate a resource to a specific location:

```
crm_resource -M -r my_first_ip -H c001n02
```

Allow a resource to return to its normal location:

```
crm_resource -U -r my_first_ip
```

注記

The values of `resource_stickiness` and `default_resource_stickiness` may mean that it does not move back. In such cases, you should use `-M` to move it back before running this command.

Delete a resource from the CRM:

```
crm_resource -D -r my_first_ip -t primitive
```

Delete a resource group from the CRM:

```
crm_resource -D -r my_first_group -t group
```

Disable resource management for a resource in the CRM:

```
crm_resource -p is-managed -r my_first_ip -t primitive -v off
```

Enable resource management for a resource in the CRM:

```
crm_resource -p is-managed -r my_first_ip -t primitive -v on
```

Reset a failed resource after having been manually cleaned up:

```
crm_resource -C -H c001n02 -r my_first_ip
```


Recheck all nodes for resources started outside the CRM:

```
crm_resource -P
```

Recheck one node for resources started outside the CRM:

```
crm_resource -P -H c001n02
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk.
Editing this file directly is strongly discouraged.

See Also

`cibadmin(8)` (224 ページ), `crmadmin(8)` (230 ページ), `lrmadmin(8)`, `heartbeat(8)`

crm_shadow (8)

crm_shadow — Perform Configuration Changes in a Sandbox Before Updating The Live Cluster

Synopsis

```
crm_shadow [-V] [-p|-e|-q]
```

Description

Sets up an environment in which configuration tools (`cibadmin`, `crm_resource`, etc) work offline instead of against a live cluster, allowing changes to be previewed and tested for side-effects.

Options

```
--verbose, -V
    turn on debug info. additional instance increase verbosity

--which, -w
    indicate the active shadow copy

--display, -p
    display the contents of the shadow copy

--diff, -d
    display the changes in the shadow copy

--create-empty, -eNAME
    create the named shadow copy with an empty cluster configuration

--create, -cNAME
    create the named shadow copy of the active cluster configuration
```

`--reset, -rNAME`
recreate the named shadow copy from the active cluster configuration

`--commit, -cNAME`
upload the contents of the named shadow copy to the cluster

`--delete, -dNAME`
delete the contents of the named shadow copy

`--edit, -eNAME`
Edit the contents of the named shadow copy with your favorite editor

`--batch, -b`
do not spawn a new shell

`--force, -f`
do not spawn a new shell

`--switch, -s`
switch to the named shadow copy

Internal Commands

To work with a shadow configuration, you need to create one first:

```
crm_shadow --create-empty YOUR_NAME
```

It gives you an internal shell like the one from the `crm` tool. Use `help` to get an overview of all internal commands, or `help subcommand` for a specific command.

表 18.1 *Overview of Internal Commands*

Command	Syntax/Description
<code>alias</code>	<code>alias [-p] [name[=value] ...]</code> <code>alias</code> with no arguments or with the <code>-p</code> option prints the list of aliases in the form <code>alias NAME=VALUE</code> on standard output. Otherwise, an alias is defined for each <code>NAME</code> whose <code>VALUE</code> is given. A trailing space in <code>VALUE</code> causes the next word to be checked for alias

Command	Syntax/Description
	substitution when the alias is expanded. Alias returns true unless a NAME is given for which no alias has been defined.
bg	bg [JOB_SPEC ...] Place each JOB_SPEC in the background, as if it had been started with &. If JOB_SPEC is not present, the shell's notion of the current job is used.
bind	bind [-lpvsPVS] [-m keymap] [-f filename] [-q name] [-u name] [-r keyseq] [-x keyseq:shell-command] [keyseq:readline-function or readline-command] Bind a key sequence to a Readline function or a macro, or set a Readline variable. The non-option argument syntax is equivalent to that found in ~/ .inputrc, but must be passed as a single argument: bind "\C-x\C-r": re-read-init-file.
break	break [N] Exit from within a for, while or until loop. If N is specified, break N levels.
builtin	builtin [shell-builtin [arg ...]] Run a shell builtin. This is useful when you wish to rename a shell builtin to be a function, but need the functionality of the builtin within the function itself.
caller	caller [EXPR] Returns the context of the current subroutine call. Without EXPR, returns \$line \$filename. With EXPR, returns \$line \$subroutine \$filename; this extra information can be used to provide a stack trace.
case	case WORD in [PATTERN [PATTERN] [COMMANDS;;] ... esac

Command	Syntax/Description
	Selectively execute <i>COMMANDS</i> based upon <i>WORD</i> matching <i>PATTERN</i> . The <code>` '</code> is used to separate multiple patterns.
<code>cd</code>	<code>cd [-L -P] [dir]</code> Change the current directory to <i>DIR</i> .
<code>command</code>	<code>command [-pVv]</code> <code>command [arg ...]</code> Runs <i>COMMAND</i> with <i>ARGS</i> ignoring shell functions. If you have a shell function called <code>`ls'</code> , and you wish to call the command <code>`ls'</code> , you can say "command ls". If the <code>-p</code> option is given, a default value is used for <i>PATH</i> that is guaranteed to find all of the standard utilities. If the <code>-V</code> or <code>-v</code> option is given, a string is printed describing <i>COMMAND</i> . The <code>-V</code> option produces a more verbose description.
<code>compgen</code>	<code>compgen [-abcdefgjkuv] [-o option] [-A action]</code> <code>[-G globpat] [-W wordlist] [-P prefix]</code> <code>[-S suffix] [-X filterpat] [-F function]</code> <code>[-C command] [WORD]</code> Display the possible completions depending on the options. Intended to be used from within a shell function generating possible completions. If the optional <i>WORD</i> argument is supplied, matches against <i>WORD</i> are generated.
<code>complete</code>	<code>complete [-abcdefgjkuv] [-pr] [-o option]</code> <code>[-A action] [-G globpat] [-W wordlist] [-P prefix]</code> <code>[-S suffix] [-X filterpat] [-F function] [-C command]</code> <code>[name ...]</code> For each <i>NAME</i> , specify how arguments are to be completed. If the <code>-p</code> option is supplied, or if no options are supplied, existing completion specifications are printed in a way that allows them to be reused as input. The <code>-r</code> option removes a completion specification for each <i>NAME</i> , or, if no <i>NAMES</i> are supplied, all completion specifications.
<code>continue</code>	<code>continue [N]</code>

Command	Syntax/Description
	Resume the next iteration of the enclosing FOR, WHILE or UNTIL loop. If <i>N</i> is specified, resume at the <i>N</i> -th enclosing loop.
declare	declare [-afFirtx] [-p] [name[=value] ...] Declare variables and/or give them attributes. If no <i>NAMES</i> are given, then display the values of variables instead. The <i>-p</i> option will display the attributes and values of each <i>NAME</i> .
dirs	dirs [-clpv] [+N] [-N] Display the list of currently remembered directories. Directories find their way onto the list with the <i>pushd</i> command; you can get back up through the list with the <i>popd</i> command.
disown	disown [-h] [-ar] [JOBSPEC ...] By default, removes each <i>JOBSPEC</i> argument from the table of active jobs. If the <i>-h</i> option is given, the job is not removed from the table, but is marked so that SIGHUP is not sent to the job if the shell receives a SIGHUP. The <i>-a</i> option, when <i>JOBSPEC</i> is not supplied, means to remove all jobs from the job table; the <i>-r</i> option means to remove only running jobs.
echo	echo [-neE] [arg ...] Output the ARGs. If <i>-n</i> is specified, the trailing newline is suppressed. If the <i>-e</i> option is given, interpretation of the following backslash-escaped characters is turned on: \ a (alert, bell) \ b (backspace) \ c (suppress trailing newline) \ E (escape character) \ f (form feed) \ n (new line)

Command	Syntax/Description
	<p> <code>\r</code> (carriage return) <code>\t</code> (horizontal tab) <code>\v</code> (vertical tab) <code>\\</code> (backslash) <code>\0nnn</code> (the character whose ASCII code is NNN (octal). NNN can be 0 to 3 octal digits) </p> <p>You can turn off the interpretation of the above characters with the <code>-E</code> option.</p>
<code>enable</code>	<p> <code>enable [-pnds] [-a] [-f filename] [name...]</code> </p> <p>Enable and disable builtin shell commands. This allows you to use a disk command which has the same name as a shell builtin without specifying a full pathname. If <code>-n</code> is used, the <i>NAMES</i> become disabled; otherwise <i>NAMES</i> are enabled. For example, to use the <code>test</code> found in <code>\$PATH</code> instead of the shell builtin version, type <code>enable -n test</code>. On systems supporting dynamic loading, the <code>-f</code> option may be used to load new builtins from the shared object <i>FILENAME</i>. The <code>-d</code> option will delete a builtin previously loaded with <code>-f</code>. If no non-option names are given, or the <code>-p</code> option is supplied, a list of builtins is printed. The <code>-a</code> option means to print every builtin with an indication of whether or not it is enabled. The <code>-s</code> option restricts the output to the POSIX.2 'special' builtins. The <code>-n</code> option displays a list of all disabled builtins.</p>
<code>eval</code>	<p> <code>eval [ARG ...]</code> </p> <p>Read <i>ARGS</i> as input to the shell and execute the resulting command(s).</p>
<code>exec</code>	<p> <code>exec [-cl] [-a name] file [redirection ...]</code> </p> <p>Exec <i>FILE</i>, replacing this shell with the specified program. If <i>FILE</i> is not specified, the redirections take effect in this shell. If the first argument is <code>-l</code>, then place a dash in the zeroth arg passed to <i>FILE</i>, as <code>login</code> does. If the <code>-c</code> option is supplied, <i>FILE</i> is executed with a null environment. The <code>-a</code> option means to make <code>set argv[0]</code> of the</p>

Command	Syntax/Description
	executed process to <i>NAME</i> . If the file cannot be executed and the shell is not interactive, then the shell exits, unless the shell option <code>execfail</code> is set.
<code>exit</code>	<code>exit [N]</code> Exit the shell with a status of <i>N</i> . If <i>N</i> is omitted, the exit status is that of the last command executed.
<code>export</code>	<code>export [-nf] [NAME[=value] ...]</code> <code>export -p</code> <i>NAMES</i> are marked for automatic export to the environment of subsequently executed commands. If the <code>-f</code> option is given, the <i>NAMES</i> refer to functions. If no <i>NAMES</i> are given, or if <code>-p</code> is given, a list of all names that are exported in this shell is printed. An argument of <code>-n</code> says to remove the export property from subsequent <i>NAMES</i> . An argument of <code>--</code> disables further option processing.
<code>false</code>	<code>false</code> Return an unsuccessful result.
<code>fc</code>	<code>fc [-e ename] [-nlr] [FIRST] [LAST]</code> <code>fc -s [pat=rep] [cmd]</code> <code>fc</code> is used to list or edit and re-execute commands from the history list. <i>FIRST</i> and <i>LAST</i> can be numbers specifying the range, or <i>FIRST</i> can be a string, which means the most recent command beginning with that string.
<code>fg</code>	<code>fg [JOB_SPEC]</code> Place <i>JOB_SPEC</i> in the foreground, and make it the current job. If <i>JOB_SPEC</i> is not present, the shell's notion of the current job is used.
<code>for</code>	<code>for NAME [in WORDS ... ;] do COMMANDS; done</code>

Command	Syntax/Description
	<p>The <code>for</code> loop executes a sequence of commands for each member in a list of items. If <code>in WORDS ... ;</code> is not present, then <code>in "\$@"</code> is assumed. For each element in <i>WORDS</i>, <i>NAME</i> is set to that element, and the <i>COMMANDS</i> are executed.</p>
<code>function</code>	<pre>function NAME { COMMANDS ; } function NAME () { COMMANDS ; }</pre> <p>Create a simple command invoked by <i>NAME</i> which runs <i>COMMANDS</i>. Arguments on the command line along with <i>NAME</i> are passed to the function as <code>\$0 .. \$n</code>.</p>
<code>getopts</code>	<pre>getopts OPTSTRING NAME [arg]</pre> <p>Getopts is used by shell procedures to parse positional parameters.</p>
<code>hash</code>	<pre>hash [-lr] [-p PATHNAME] [-dt] [NAME...]</pre> <p>For each <i>NAME</i>, the full pathname of the command is determined and remembered. If the <code>-p</code> option is supplied, <i>PATHNAME</i> is used as the full pathname of <i>NAME</i>, and no path search is performed. The <code>-r</code> option causes the shell to forget all remembered locations. The <code>-d</code> option causes the shell to forget the remembered location of each <i>NAME</i>. If the <code>-t</code> option is supplied the full pathname to which each <i>NAME</i> corresponds is printed. If multiple <i>NAME</i> arguments are supplied with <code>-t</code>, the <i>NAME</i> is printed before the hashed full pathname. The <code>-l</code> option causes output to be displayed in a format that may be reused as input. If no arguments are given, information about remembered commands is displayed.</p>
<code>history</code>	<pre>history [-c] [-d OFFSET] [n] history -ps arg [arg...] history -awrm [filename]</pre> <p>Display the history list with line numbers. Lines listed with with a <code>*</code> have been modified. Argument of <i>N</i> says to list only the last <i>N</i> lines. The <code>-c</code> option causes the history list to be cleared by deleting all of</p>

Command	Syntax/Description
	<p>the entries. The <code>-d</code> option deletes the history entry at offset <i>OFFSET</i>. The <code>-w</code> option writes out the current history to the history file; <code>-r</code> means to read the file and append the contents to the history list instead. <code>-a</code> means to append history lines from this session to the history file. Argument <code>-n</code> means to read all history lines not already read from the history file and append them to the history list.</p>
<code>jobs</code>	<pre>jobs [-lnprs] [JOBSPEC ...] job -x COMMAND [ARGS]</pre> <p>Lists the active jobs. The <code>-l</code> option lists process id's in addition to the normal information; the <code>-p</code> option lists process id's only. If <code>-n</code> is given, only processes that have changed status since the last notification are printed. <i>JOBSPEC</i> restricts output to that job. The <code>-r</code> and <code>-s</code> options restrict output to running and stopped jobs only, respectively. Without options, the status of all active jobs is printed. If <code>-x</code> is given, <i>COMMAND</i> is run after all job specifications that appear in <i>ARGS</i> have been replaced with the process ID of that job's process group leader.</p>
<code>kill</code>	<pre>kill [-s sigspec -n signum -sigspec] pid JOBSPEC ... kill -l [sigspec]</pre> <p>Send the processes named by PID (or <i>JOBSPEC</i>) the signal <i>SIGSPEC</i>. If <i>SIGSPEC</i> is not present, then <i>SIGTERM</i> is assumed. An argument of <code>-l</code> lists the signal names; if arguments follow <code>-l</code> they are assumed to be signal numbers for which names should be listed. Kill is a shell builtin for two reasons: it allows job IDs to be used instead of process IDs, and, if you have reached the limit on processes that you can create, you don't have to start a process to kill another one.</p>
<code>let</code>	<pre>let ARG [ARG ...]</pre> <p>Each <i>ARG</i> is a mathematical expression to be evaluated. Evaluation is done in fixed-width integers with no check for overflow, though division by 0 is trapped and flagged as an error. The following list of operators is grouped into levels of equal-precedence operators. The levels are listed in order of decreasing precedence.</p>

Command	Syntax/Description
<code>local</code>	<pre>local NAME [=VALUE] ...</pre> <p>Create a local variable called <i>NAME</i>, and give it <i>VALUE</i>. <code>local</code> can only be used within a function; it makes the variable <i>NAME</i> have a visible scope restricted to that function and its children.</p>
<code>logout</code>	<pre>logout</pre> <p>Logout of a login shell.</p>
<code>popd</code>	<pre>popd [+N -N] [-n]</pre> <p>Removes entries from the directory stack. With no arguments, removes the top directory from the stack, and <code>cd</code>'s to the new top directory.</p>
<code>printf</code>	<pre>printf [-v var] format [ARGUMENTS]</pre> <p><code>printf</code> formats and prints <i>ARGUMENTS</i> under control of the <i>FORMAT</i>. <i>FORMAT</i> is a character string which contains three types of objects: plain characters, which are simply copied to standard output, character escape sequences which are converted and copied to the standard output, and format specifications, each of which causes printing of the next successive argument. In addition to the standard <code>printf(1)</code> formats, <code>%b</code> means to expand backslash escape sequences in the corresponding argument, and <code>%q</code> means to quote the argument in a way that can be reused as shell input. If the <code>-v</code> option is supplied, the output is placed into the value of the shell variable <i>VAR</i> rather than being sent to the standard output.</p>
<code>pushd</code>	<pre>pushd [dir +N -N] [-n]</pre> <p>Adds a directory to the top of the directory stack, or rotates the stack, making the new top of the stack the current working directory. With no arguments, exchanges the top two directories.</p>
<code>pwd</code>	<pre>pwd [-LP]</pre>

Command	Syntax/Description
	<p>Print the current working directory. With the <code>-P</code> option, <code>pwd</code> prints the physical directory, without any symbolic links; the <code>-L</code> option makes <code>pwd</code> follow symbolic links.</p>
<code>read</code>	<pre>read [-ers] [-u fd] [-t timeout] [-p prompt] [-a array] [-n nchars] [-d delim] [NAME ...]</pre> <p>The given <i>NAMES</i> are marked readonly and the values of these <i>NAMES</i> may not be changed by subsequent assignment. If the <code>-f</code> option is given, then functions corresponding to the <i>NAMES</i> are so marked. If no arguments are given, or if <code>-p</code> is given, a list of all readonly names is printed. The <code>-a</code> option means to treat each <i>NAME</i> as an array variable. An argument of <code>--</code> disables further option processing.</p>
<code>readonly</code>	<pre>readonly [-af] [NAME[=VALUE] ...] readonly -p</pre> <p>The given <i>NAMES</i> are marked readonly and the values of these <i>NAMES</i> may not be changed by subsequent assignment. If the <code>-f</code> option is given, then functions corresponding to the <i>NAMES</i> are so marked. If no arguments are given, or if <code>-p</code> is given, a list of all readonly names is printed. The <code>-a</code> option means to treat each <i>NAME</i> as an array variable. An argument of <code>--</code> disables further option processing.</p>
<code>return</code>	<pre>return [N]</pre> <p>Causes a function to exit with the return value specified by <i>N</i>. If <i>N</i> is omitted, the return status is that of the last command.</p>
<code>select</code>	<pre>select NAME [in WORDS ... ;] do COMMANDS; done</pre> <p>The <i>WORDS</i> are expanded, generating a list of words. The set of expanded words is printed on the standard error, each preceded by a number. If <code>in WORDS</code> is not present, <code>in "\$@"</code> is assumed. The PS3 prompt is then displayed and a line read from the standard input. If the line consists of the number corresponding to one of the displayed words, then <i>NAME</i> is set to that word. If the line is empty, <i>WORDS</i> and</p>

Command	Syntax/Description
	<p>the prompt are redisplayed. If EOF is read, the command completes. Any other value read causes <i>NAME</i> to be set to null. The line read is saved in the variable <i>REPLY</i>. <i>COMMANDS</i> are executed after each selection until a break command is executed.</p>
set	<pre>set [--abefhkmnptuvxBCHP] [-o OPTION] [ARG...]</pre> <p>Sets internal shell options.</p>
shift	<pre>shift [n]</pre> <p>The positional parameters from $\\$N+1$. . . are renamed to $\\$1$. . . If <i>N</i> is not given, it is assumed to be 1.</p>
shopt	<pre>shopt [-pqsu] [-o long-option] OPTNAME [OPTNAME...]</pre> <p>Toggle the values of variables controlling optional behavior. The <i>-s</i> flag means to enable (set) each <i>OPTNAME</i>; the <i>-u</i> flag unsets each <i>OPTNAME</i>. The <i>-q</i> flag suppresses output; the exit status indicates whether each <i>OPTNAME</i> is set or unset. The <i>-o</i> option restricts the <i>OPTNAME</i>s to those defined for use with <code>set -o</code>. With no options, or with the <i>-p</i> option, a list of all settable options is displayed, with an indication of whether or not each is set.</p>
source	<pre>source FILENAME [ARGS]</pre> <p>Read and execute commands from <i>FILENAME</i> and return. The pathnames in $\\$PATH$ are used to find the directory containing <i>FILENAME</i>. If any <i>ARGS</i> are supplied, they become the positional parameters when <i>FILENAME</i> is executed.</p>
suspend	<pre>suspend [-f]</pre> <p>Suspend the execution of this shell until it receives a SIGCONT signal. The <i>-f</i> if specified says not to complain about this being a login shell if it is; just suspend anyway.</p>

Command	Syntax/Description
test	<pre>test [expr]</pre> <p>Exits with a status of 0 (true) or 1 (false) depending on the evaluation of <i>EXPR</i>. Expressions may be unary or binary. Unary expressions are often used to examine the status of a file. There are string operators as well, and numeric comparison operators.</p>
time	<pre>time [-p] PIPELINE</pre> <p>Execute <i>PIPELINE</i> and print a summary of the real time, user CPU time, and system CPU time spent executing <i>PIPELINE</i> when it terminates. The return status is the return status of <i>PIPELINE</i>. The <i>-p</i> option prints the timing summary in a slightly different format. This uses the value of the <i>TIMEFORMAT</i> variable as the output format.</p>
times	<pre>times</pre> <p>Print the accumulated user and system times for processes run from the shell.</p>
trap	<pre>trap [-lp] [ARG SIGNAL_SPEC ...]</pre> <p>The command <i>ARG</i> is to be read and executed when the shell receives signal(s) <i>SIGNAL_SPEC</i>. If <i>ARG</i> is absent (and a single <i>SIGNAL_SPEC</i> is supplied) or <i>-</i>, each specified signal is reset to its original value. If <i>ARG</i> is the null string each <i>SIGNAL_SPEC</i> is ignored by the shell and by the commands it invokes. If a <i>SIGNAL_SPEC</i> is <i>EXIT</i> (0) the command <i>ARG</i> is executed on exit from the shell. If a <i>SIGNAL_SPEC</i> is <i>DEBUG</i>, <i>ARG</i> is executed after every simple command. If the <i>-p</i> option is supplied then the trap commands associated with each <i>SIGNAL_SPEC</i> are displayed. If no arguments are supplied or if only <i>-p</i> is given, trap prints the list of commands associated with each signal. Each <i>SIGNAL_SPEC</i> is either a signal name in <i>signal.h</i> or a signal number. Signal names are case insensitive and the <i>SIG</i> prefix is optional. <code>trap -l</code> prints a list of signal names and their corresponding numbers. Note that a signal can be sent to the shell with <code>kill -signal \$\$</code>.</p>

Command	Syntax/Description
true	<pre>true</pre> <p>Return a successful result.</p>
type	<pre>type [-afptP] NAME [NAME ...]</pre> <p>Obsolete, see declare.</p>
typeset	<pre>typeset [-afFirtx] [-p] name[=value]</pre> <p>Obsolete, see declare.</p>
ulimit	<pre>ulimit [-SHacdfilmpqstuvx] [limit]</pre> <p>Ulimit provides control over the resources available to processes started by the shell, on systems that allow such control.</p>
umask	<pre>umask [-p] [-S] [MODE]</pre> <p>The user file-creation mask is set to <i>MODE</i>. If <i>MODE</i> is omitted, or if <i>-S</i> is supplied, the current value of the mask is printed. The <i>-S</i> option makes the output symbolic; otherwise an octal number is output. If <i>-p</i> is supplied, and <i>MODE</i> is omitted, the output is in a form that may be used as input. If <i>MODE</i> begins with a digit, it is interpreted as an octal number, otherwise it is a symbolic mode string like that accepted by <code>chmod(1)</code>.</p>
unalias	<pre>unalias [-a] NAME [NAME ...]</pre> <p>Remove <i>NAMES</i> from the list of defined aliases. If the <i>-a</i> option is given, then remove all alias definitions.</p>
unset	<pre>unset [-f] [-v] [NAME ...]</pre> <p>For each <i>NAME</i>, remove the corresponding variable or function. Given the <i>-v</i>, unset will only act on variables. Given the <i>-f</i> flag, unset will only act on functions. With neither flag, unset first tries to unset a variable. If that fails, it then tries to unset a function. Some variables cannot be unset; also see <code>readonly</code>.</p>

Command	Syntax/Description
<code>until</code>	<p><code>until COMMANDS; do COMMANDS; done</code></p> <p>Expand and execute <i>COMMANDS</i> as long as the final command in the <code>until COMMANDS</code> has an exit status which is not zero.</p>
<code>wait</code>	<p><code>wait [N]</code></p> <p>Wait for the specified process and report its termination status. If <i>N</i> is not given, all currently active child processes are waited for, and the return code is zero. <i>N</i> may be a process ID or a job specification; if a job spec is given, all processes in the job's pipeline are waited for.</p>
<code>while</code>	<p><code>while COMMANDS; do COMMANDS; done</code></p> <p>Expand and execute <i>COMMANDS</i> as long as the final command in the <code>while COMMANDS</code> has an exit status of zero.</p>

crm_standby (8)

`crm_standby` — manipulate a node's standby attribute to determine whether resources can be run on this node

Synopsis

```
crm_standby [-?|-V] -D -u|-U node -r resource
crm_standby [-?|-V] -G -u|-U node -r resource
crm_standby [-?|-V] -v string -u|-U node -r resource [-l string]
```

Description

The `crm_standby` command manipulates a node's standby attribute. Any node in standby mode is no longer eligible to host resources and any resources that are there must be moved. Standby mode can be useful for performing maintenance tasks, such as kernel updates. Remove the standby attribute from the node when it needs to become a fully active member of the cluster again.

By assigning a lifetime to the `standby` attribute, determine whether the standby setting should survive a reboot of the node (set lifetime to `forever`) or should be reset with reboot (set lifetime to `reboot`). Alternatively, remove the `standby` attribute and bring the node back from standby manually.

Options

`--help, -?`
Print a help message.

`--verbose, -V`
Turn on debug information.

注記

Increase the level of verbosity by providing additional instances.

`--quiet, -Q`

When doing an attribute query using `-G`, print just the value to stdout. Use this option with `-G`.

`--get-value, -G`

Retrieve rather than set the preference.

`--delete-attr, -D`

Specify the attribute to delete.

`--attr-value string, -v string`

Specify the value to use. This option is ignored when used with `-G`.

`--attr-id string, -i string`

For advanced users only. Identifies the id attribute..

`--node node_undef, -u node_undef`

Specify the uname of the node to change.

`--lifetime string, -l string`

Determine how long this preference lasts. Possible values are `reboot` or `forever`.

注記

If a `forever` value exists, it is always used by the CRM instead of any `reboot` value.

Examples

Have a local node go to standby:

```
crm_standby -v true
```

Have a node (`node1`) go to standby:

```
crm_standby -v true -U node1
```

Query the standby status of a node:

```
crm_standby -G -U node1
```

Remove the standby property from a node:

```
crm_standby -D -U node1
```

Have a node go to standby for an indefinite period of time:

```
crm_standby -v true -l forever -U node1
```

Have a node go to standby until the next reboot of this node:

```
crm_standby -v true -l reboot -U node1
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk.
Editing this file directly is strongly discouraged.

See Also

`cibadmin(8)` (224 ページ), `crm_attribute(8)` (233 ページ)

crm_verify (8)

crm_verify — check the CIB for consistency

Synopsis

```
crm_verify [-V] -x file
crm_verify [-V] -X string
crm_verify [-V] -L|-p
crm_verify [-?]
```

Description

crm_verify checks the configuration database (CIB) for consistency and other problems. It can be used to check a file containing the configuration or can it can connect to a running cluster. It reports two classes of problems, errors and warnings. Errors must be fixed before High Availability can work properly. However, it is left up to the administrator to decide if the warnings should also be fixed.

crm_verify assists in creating new or modified configurations. You can take a local copy of a CIB in the running cluster, edit it, validate it using crm_verify, then put the new configuration into effect using cibadmin.

Options

--help, -h
Print a help message.

--verbose, -V
Turn on debug information.

注記

Increase the level of verbosity by providing additional instances.

`--live-check, -L`

Connect to the running cluster and check the CIB.

`--crm_xml string, -X string`

Check the configuration in the supplied string. Pass complete CIBs only.

`--xml-file file, -x file`

Check the configuration in the named file.

`--xml-pipe, -p`

Use the configuration piped in via stdin. Pass complete CIBs only.

Examples

Check the consistency of the configuration in the running cluster and produce verbose output:

```
crm_verify -VL
```

Check the consistency of the configuration in a given file and produce verbose output:

```
crm_verify -Vx file1
```

Pipe a configuration into `crm_verify` and produce verbose output:

```
cat file1.xml | crm_verify -Vp
```

Files

`/var/lib/heartbeat/crm/cib.xml`—the CIB (minus status section) on disk.
Editing this file directly is strongly discouraged.

See Also

`cibadmin(8)` (224 ページ)

HA OCF Agents

All OCF agents require several parameters to be set when they are started. The following overview shows how to manually operate these agents. The data that is available in this appendix is directly taken from the `meta-data` invocation of the respective RA. Find all these agents in `/usr/lib/ocf/resource.d/heartbeat/`.

When configuring an RA, omit the `OCF_RESKEY_` prefix to the parameter name. Parameters that are in square brackets may be omitted in the configuration.

ocf:anything (7)

ocf:anything — Manages an arbitrary service

Synopsis

```
OCF_RESKEY_binfile=string [OCF_RESKEY_cmdline_options=string]  
[OCF_RESKEY_pidfile=string] [OCF_RESKEY_logfile=string]  
[OCF_RESKEY_errlogfile=string] [OCF_RESKEY_user=string]  
[OCF_RESKEY_monitor_hook=string] [OCF_RESKEY_stop_timeout=string]  
anything [start | stop | monitor | meta-data | validate-all]
```

Description

This is a generic OCF RA to manage almost anything.

Supported Parameters

OCF_RESKEY_binfile=Full path name of the binary to be executed
The full name of the binary to be executed. This is expected to keep running with the same pid and not just do something and exit.

OCF_RESKEY_cmdline_options=Command line options
Command line options to pass to the binary

OCF_RESKEY_pidfile=File to write STDOUT to
File to read/write the PID from/to.

OCF_RESKEY_logfile=File to write STDOUT to
File to write STDOUT to

OCF_RESKEY_errlogfile=File to write STDERR to
File to write STDERR to

OCF_RESKEY_user=User to run the command as
User to run the command as

OCF_RESKEY_monitor_hook=Command to run in monitor operation
Command to run in monitor operation

OCF_RESKEY_stop_timeout=Seconds to wait after having sent SIGTERM before
sending SIGKILL in stop operation

In the stop operation: Seconds to wait for kill -TERM to succeed before sending
kill -SIGKILL. Defaults to 2/3 of the stop operation timeout.

ocf:AoEtarget (7)

ocf:AoEtarget — Manages ATA-over-Ethernet (AoE) target exports

Synopsis

```
[OCF_RESKEY_device=string] [OCF_RESKEY_nic=string]  
[OCF_RESKEY_shelf=integer] [OCF_RESKEY_slot=integer]  
OCF_RESKEY_pid=string [OCF_RESKEY_binary=string] AoEtarget [start |  
stop | monitor | reload | meta-data | validate-all]
```

Description

This resource agent manages an ATA-over-Ethernet (AoE) target using vblade. It exports any block device, or file, as an AoE target using the specified Ethernet device, shelf, and slot number.

Supported Parameters

OCF_RESKEY_device=Device to export

The local block device (or file) to export as an AoE target.

OCF_RESKEY_nic=Ethernet interface

The local Ethernet interface to use for exporting this AoE target.

OCF_RESKEY_shelf=AoE shelf number

The AoE shelf number to use when exporting this target.

OCF_RESKEY_slot=AoE slot number

The AoE slot number to use when exporting this target.

OCF_RESKEY_pid=Daemon pid file

The file to record the daemon pid to.

OCF_RESKEY_binary=vblade binary

Location of the vblade binary.

ocf:apache (7)

ocf:apache — Manages an Apache web server instance

Synopsis

```
OCF_RESKEY_configfile=string [OCF_RESKEY_httpd=string]
[OCF_RESKEY_port=integer] [OCF_RESKEY_statusurl=string]
[OCF_RESKEY_testregex=string] [OCF_RESKEY_client=string]
[OCF_RESKEY_testurl=string] [OCF_RESKEY_testregex10=string]
[OCF_RESKEY_testconf=string] [OCF_RESKEY_testname=string]
[OCF_RESKEY_options=string] [OCF_RESKEY_envfiles=string] apache
[start | stop | status | monitor | meta-data | validate-all]
```

Description

This is the resource agent for the Apache web server. This resource agent operates both version 1.x and version 2.x Apache servers. The start operation ends with a loop in which monitor is repeatedly called to make sure that the server started and that it is operational. Hence, if the monitor operation does not succeed within the start operation timeout, the apache resource will end with an error status. The monitor operation by default loads the server status page which depends on the mod_status module and the corresponding configuration file (usually /etc/apache2/mod_status.conf). Make sure that the server status page works and that the access is allowed **only** from localhost (address 127.0.0.1). See the statusurl and testregex attributes for more details. See also <http://httpd.apache.org/>

Supported Parameters

OCF_RESKEY_configfile=configuration file path

The full pathname of the Apache configuration file. This file is parsed to provide defaults for various other resource agent parameters.

OCF_RESKEY_httpd=httpd binary path

The full pathname of the httpd binary (optional).

`OCF_RESKEY_port=httpd port`

A port number that we can probe for status information using the `statusurl`. This will default to the port number found in the configuration file, or 80, if none can be found in the configuration file.

`OCF_RESKEY_statusurl=url name`

The URL to monitor (the apache server status page by default). If left unspecified, it will be inferred from the apache configuration file. If you set this, make sure that it succeeds **only** from the localhost (127.0.0.1). Otherwise, it may happen that the cluster complains about the resource being active on multiple nodes.

`OCF_RESKEY_testregex=monitor regular expression`

Regular expression to match in the output of `statusurl`. Case insensitive.

`OCF_RESKEY_client=http client`

Client to use to query to Apache. If not specified, the RA will try to find one on the system. Currently, `wget` and `curl` are supported. For example, you can set this parameter to "curl" if you prefer that to `wget`.

`OCF_RESKEY_testurl=test url`

URL to test. If it does not start with "http", then it's considered to be relative to the Listen address.

`OCF_RESKEY_testregex10=extended monitor regular expression`

Regular expression to match in the output of `testurl`. Case insensitive.

`OCF_RESKEY_testconf file=test configuration file`

A file which contains test configuration. Could be useful if you have to check more than one web application or in case sensitive info should be passed as arguments (passwords). Furthermore, using a config file is the only way to specify certain parameters. Please see `README.webapps` for examples and file description.

`OCF_RESKEY_testname=test name`

Name of the test within the test configuration file.

`OCF_RESKEY_options=command line options`

Extra options to apply when starting apache. See `man httpd(8)`.

OCF_RESKEY_envfiles=environment settings files

Files (one or more) which contain extra environment variables. If you want to prevent script from reading the default file, set this parameter to empty string.

ocf:AudibleAlarm (7)

ocf:AudibleAlarm — Emits audible beeps at a configurable interval

Synopsis

```
[OCF_RESKEY_nodelist=string] AudibleAlarm [start | stop | restart | status |  
monitor | meta-data | validate-all]
```

Description

Resource script for AudibleAlarm. It sets an audible alarm running by beeping at a set interval.

Supported Parameters

OCF_RESKEY_nodelist=Node list

The node list that should never sound the alarm.

ocf:ClusterMon (7)

ocf:ClusterMon — Runs `crm_mon` in the background, recording the cluster status to an HTML file

Synopsis

```
[OCF_RESKEY_user=string] [OCF_RESKEY_update=integer]  
[OCF_RESKEY_extra_options=string] OCF_RESKEY_pidfile=string  
OCF_RESKEY_htmlfile=string ClusterMon [start | stop | monitor | meta-data |  
validate-all]
```

Description

This is a ClusterMon Resource Agent. It outputs current cluster status to the html.

Supported Parameters

OCF_RESKEY_user=The user we want to run `crm_mon` as
The user we want to run `crm_mon` as

OCF_RESKEY_update=Update interval
How frequently should we update the cluster status

OCF_RESKEY_extra_options=Extra options
Additional options to pass to `crm_mon`. Eg. `-n -r`

OCF_RESKEY_pidfile=PID file
PID file location to ensure only one instance is running

OCF_RESKEY_htmlfile=HTML output
Location to write HTML output to.

ocf:CTDB (7)

ocf:CTDB — CTDB Resource Agent

Synopsis

```
OCF_RESKEY_ctdb_recovery_lock=string
OCF_RESKEY_smb_private_dir=string
[OCF_RESKEY_ctdb_config_dir=string]
[OCF_RESKEY_ctdb_binary=string] [OCF_RESKEY_ctdbd_binary=string]
[OCF_RESKEY_ctdb_socket=string] [OCF_RESKEY_ctdb_dbdir=string]
[OCF_RESKEY_ctdb_logfile=string]
[OCF_RESKEY_ctdb_debuglevel=integer] [OCF_RESKEY_smb_conf=string]
CTDB [start | stop | monitor | meta-data | validate-all]
```

Description

This resource agent manages CTDB, allowing one to use Clustered Samba in a Linux-HA/Pacemaker cluster. You need a shared filesystem (e.g. OCFS2) on which CTDB lock and Samba state will be stored. Configure shares in `smb.conf` on all nodes, and create `/etc/ctdb/nodes` containing a list of private IP addresses of each node in the cluster. Configure this RA as a clone, and it will take care of the rest. For more information see [http://linux-ha.org/wiki/CTDB_\(resource_agent\)](http://linux-ha.org/wiki/CTDB_(resource_agent))

Supported Parameters

`OCF_RESKEY_ctdb_recovery_lock=CTDB shared lock file`

The location of a shared lock file, common across all nodes. This must be on shared storage, e.g.: `/shared-fs/samba/ctdb.lock`

`OCF_RESKEY_smb_private_dir=Samba private dir`

The directory for `smbd` to use for storing such files as `smbpasswd` and `secrets.tdb`. This must be on shared storage, e.g.: `/shared-fs/samba/private`

`OCF_RESKEY_ctdb_config_dir`=CTDB config file directory

The directory containing various CTDB configuration files. The "nodes" and "notify.sh" scripts are expected to be in this directory, as is the "events.d" subdirectory.

`OCF_RESKEY_ctdb_binary`=CTDB binary path

Full path to the CTDB binary.

`OCF_RESKEY_ctdbd_binary`=CTDB Daemon binary path

Full path to the CTDB cluster daemon binary.

`OCF_RESKEY_ctdb_socket`=CTDB socket location

Full path to the domain socket that ctdbd will create, used for local clients to attach and communicate with the ctdb daemon.

`OCF_RESKEY_ctdb_dbdir`=CTDB database directory

The directory to put the local CTDB database files in. Persistent database files will be put in ctdb_dbdir/persistent.

`OCF_RESKEY_ctdb_logfile`=CTDB log file location

Full path to log file. To log to syslog instead, use the value "syslog".

`OCF_RESKEY_ctdb_debuglevel`=CTDB debug level

What debug level to run at (0-10). Higher means more verbose.

`OCF_RESKEY_smb_conf`=Path to smb.conf

Path to default samba config file.

ocf:db2 (7)

ocf:db2 — Manages an IBM DB2 Universal Database instance

Synopsis

```
[OCF_RESKEY_instance=string] [OCF_RESKEY_admin=string] db2 [start | stop  
| status | monitor | validate-all | meta-data | methods]
```

Description

Resource script for db2. It manages a DB2 Universal Database instance as an HA resource.

Supported Parameters

OCF_RESKEY_instance=instance
The instance of database.

OCF_RESKEY_admin=admin
The admin user of the instance.

ocf:Delay (7)

ocf:Delay — Waits for a defined timespan

Synopsis

```
[OCF_RESKEY_startdelay=integer] [OCF_RESKEY_stopdelay=integer]  
[OCF_RESKEY_mondelay=integer] Delay [start | stop | status | monitor | meta-data  
| validate-all]
```

Description

This script is a test resource for introducing delay.

Supported Parameters

OCF_RESKEY_startdelay=Start delay
How long in seconds to delay on start operation.

OCF_RESKEY_stopdelay=Stop delay
How long in seconds to delay on stop operation. Defaults to "startdelay" if unspecified.

OCF_RESKEY_mondelay=Monitor delay
How long in seconds to delay on monitor operation. Defaults to "startdelay" if unspecified.

ocf:drbd (7)

ocf:drbd — Manages a DRBD resource (deprecated)

Synopsis

```
OCF_RESKEY_drbd_resource=string [OCF_RESKEY_drbdconf=string]
[OCF_RESKEY_clone_overrides_hostname=boolean]
[OCF_RESKEY_ignore_deprecation=boolean] drbd [start | promote | demote
| notify | stop | monitor | monitor | meta-data | validate-all]
```

Description

Deprecation warning: This agent is deprecated and may be removed from a future release. See the ocf:linbit:drbd resource agent for a supported alternative. -- This resource agent manages a Distributed Replicated Block Device (DRBD) object as a master/slave resource. DRBD is a mechanism for replicating storage; please see the documentation for setup details.

Supported Parameters

OCF_RESKEY_drbd_resource=drbd resource name
The name of the drbd resource from the drbd.conf file.

OCF_RESKEY_drbdconf=Path to drbd.conf
Full path to the drbd.conf file.

OCF_RESKEY_clone_overrides_hostname=Override drbd hostname
Whether or not to override the hostname with the clone number. This can be used to create floating peer configurations; drbd will be told to use node_<cloneno> as the hostname instead of the real uname, which can then be used in drbd.conf.

OCF_RESKEY_ignore_deprecation=Suppress deprecation warning
If set to true, suppresses the deprecation warning for this agent.

ocf:Dummy (7)

ocf:Dummy — Example stateless resource agent

Synopsis

`OCF_RESKEY_state=string Dummy [start | stop | monitor | reload | migrate_to | migrate_from | meta-data | validate-all]`

Description

This is a Dummy Resource Agent. It does absolutely nothing except keep track of whether its running or not. Its purpose in life is for testing and to serve as a template for RA writers.

Supported Parameters

`OCF_RESKEY_state=State file`
Location to store the resource state in.

ocf:eDir88 (7)

ocf:eDir88 — Manages a Novell eDirectory directory server

Synopsis

```
OCF_RESKEY_eDir_config_file=string  
[OCF_RESKEY_eDir_monitor_ldap=boolean]  
[OCF_RESKEY_eDir_monitor_idm=boolean]  
[OCF_RESKEY_eDir_jvm_initial_heap=integer]  
[OCF_RESKEY_eDir_jvm_max_heap=integer]  
[OCF_RESKEY_eDir_jvm_options=string] eDir88 [start | stop | monitor | meta-  
data | validate-all]
```

Description

Resource script for managing an eDirectory instance. Manages a single instance of eDirectory as an HA resource. The "multiple instances" feature of eDirectory has been added in version 8.8. This script will not work for any version of eDirectory prior to 8.8. This RA can be used to load multiple eDirectory instances on the same host. It is very strongly recommended to put eDir configuration files (as per the `eDir_config_file` parameter) on local storage on each node. This is necessary for this RA to be able to handle situations where the shared storage has become unavailable. If the eDir configuration file is not available, this RA will fail, and heartbeat will be unable to manage the resource. Side effects include STONITH actions, unmanageable resources, etc... Setting a high action timeout value is `_very__strongly_` recommended. eDir with IDM can take in excess of 10 minutes to start. If heartbeat times out before eDir has had a chance to start properly, mayhem `_WILL ENSUE_`. The LDAP module seems to be one of the very last to start. So this script will take even longer to start on installations with IDM and LDAP if the monitoring of IDM and/or LDAP is enabled, as the start command will wait for IDM and LDAP to be available.

Supported Parameters

OCF_RESKEY_eDir_config_file=eDir config file

Path to configuration file for eDirectory instance.

OCF_RESKEY_eDir_monitor_ldap=eDir monitor ldap

Should we monitor if LDAP is running for the eDirectory instance?

OCF_RESKEY_eDir_monitor_idm=eDir monitor IDM

Should we monitor if IDM is running for the eDirectory instance?

OCF_RESKEY_eDir_jvm_initial_heap=DHOST_INITIAL_HEAP value

Value for the DHOST_INITIAL_HEAP java environment variable. If unset, java defaults will be used.

OCF_RESKEY_eDir_jvm_max_heap=DHOST_MAX_HEAP value

Value for the DHOST_MAX_HEAP java environment variable. If unset, java defaults will be used.

OCF_RESKEY_eDir_jvm_options=DHOST_OPTIONS value

Value for the DHOST_OPTIONS java environment variable. If unset, original values will be used.

ocf:Evmsd (7)

ocf:Evmsd — Controls clustered EVMS volume management (deprecated)

Synopsis

[OCF_RESKEY_ignore_deprecation=**boolean**] Evmsd [start | stop | monitor | meta-data]

Description

Deprecation warning: EVMS is no longer actively maintained and should not be used. This agent is deprecated and may be removed from a future release. -- This is a Evmsd Resource Agent.

Supported Parameters

OCF_RESKEY_ignore_deprecation=Suppress deprecation warning
If set to true, suppresses the deprecation warning for this agent.

ocf:EvmsSCC (7)

ocf:EvmsSCC — Manages EVMS Shared Cluster Containers (SCCs) (deprecated)

Synopsis

```
[OCF_RESKEY_ignore_deprecation=boolean] EvmsSCC [start | stop | notify  
| status | monitor | meta-data]
```

Description

Deprecation warning: EVMS is no longer actively maintained and should not be used. This agent is deprecated and may be removed from a future release. -- Resource script for EVMS shared cluster container. It runs `evms_activate` on one node in the cluster.

Supported Parameters

`OCF_RESKEY_ignore_deprecation=Suppress deprecation warning`
If set to true, suppresses the deprecation warning for this agent.

ocf:Filesystem (7)

ocf:Filesystem — Manages filesystem mounts

Synopsis

```
[OCF_RESKEY_device=string] [OCF_RESKEY_directory=string]  
[OCF_RESKEY_fstype=string] [OCF_RESKEY_options=string]  
[OCF_RESKEY_statusfile_prefix=string] Filesystem [start | stop | notify  
| monitor | validate-all | meta-data]
```

Description

Resource script for Filesystem. It manages a Filesystem on a shared storage medium. The standard monitor operation of depth 0 (also known as probe) checks if the filesystem is mounted. If you want deeper tests, set `OCF_CHECK_LEVEL` to one of the following values: 10: read first 16 blocks of the device (raw read) This doesn't exercise the filesystem at all, but the device on which the filesystem lives. This is noop for non-block devices such as NFS, SMBFS, or bind mounts. 20: test if a status file can be written and read The status file must be writable by root. This is not always the case with an NFS mount, as NFS exports usually have the "root_squash" option set. In such a setup, you must either use read-only monitoring (depth=10), export with "no_root_squash" on your NFS server, or grant world write permissions on the directory where the status file is to be placed.

Supported Parameters

`OCF_RESKEY_device=block device`

The name of block device for the filesystem, or -U, -L options for mount, or NFS mount specification.

`OCF_RESKEY_directory=mount point`

The mount point for the filesystem.

OCF_RESKEY_fstype=filesystem type

The optional type of filesystem to be mounted.

OCF_RESKEY_options=options

Any extra options to be given as -o options to mount. For bind mounts, add "bind" here and set fstype to "none". We will do the right thing for options such as "bind,ro".

OCF_RESKEY_statusfile_prefix=status file prefix

The prefix to be used for a status file for resource monitoring with depth 20. If you don't specify this parameter, all status files will be created in a separate directory.

ocf:ICP (7)

ocf:ICP — Manages an ICP Vortex clustered host drive

Synopsis

```
[OCF_RESKEY_driveid=string] [OCF_RESKEY_device=string] ICP [start | stop  
| status | monitor | validate-all | meta-data]
```

Description

Resource script for ICP. It Manages an ICP Vortex clustered host drive as an HA resource.

Supported Parameters

OCF_RESKEY_driveid=ICP cluster drive ID
The ICP cluster drive ID.

OCF_RESKEY_device=device
The device name.

ocf:ids (7)

ocf:ids — Manages an Informix Dynamic Server (IDS) instance

Synopsis

```
[OCF_RESKEY_informixdir=string] [OCF_RESKEY_informixserver=string]  
[OCF_RESKEY_onconfig=string] [OCF_RESKEY_dbname=string]  
[OCF_RESKEY_sqltestquery=string] ids [start | stop | status | monitor | validate-  
all | meta-data | methods | usage]
```

Description

OCF resource agent to manage an IBM Informix Dynamic Server (IDS) instance as an High-Availability resource.

Supported Parameters

OCF_RESKEY_informixdir= INFORMIXDIR environment variable

The value the environment variable INFORMIXDIR has after a typical installation of IDS. Or in other words: the path (without trailing '/') where IDS was installed to. If this parameter is unspecified the script will try to get the value from the shell environment.

OCF_RESKEY_informixserver= INFORMIXSERVER environment variable

The value the environment variable INFORMIXSERVER has after a typical installation of IDS. Or in other words: the name of the IDS server instance to manage. If this parameter is unspecified the script will try to get the value from the shell environment.

OCF_RESKEY_onconfig= ONCONFIG environment variable

The value the environment variable ONCONFIG has after a typical installation of IDS. Or in other words: the name of the configuration file for the IDS instance specified in INFORMIXSERVER. The specified configuration file will be searched

at '/etc/'. If this parameter is unspecified the script will try to get the value from the shell environment.

OCF_RESKEY_dbname= database to use for monitoring, defaults to 'sysmaster'
This parameter defines which database to use in order to monitor the IDS instance. If this parameter is unspecified the script will use the 'sysmaster' database as a default.

OCF_RESKEY_sqltestquery= SQL test query to use for monitoring, defaults to 'SELECT COUNT(*) FROM systables;'
SQL test query to run on the database specified by the parameter 'dbname' in order to monitor the IDS instance and determine if it's functional or not. If this parameter is unspecified the script will use 'SELECT COUNT(*) FROM systables;' as a default.

ocf:IPAddr2 (7)

ocf:IPAddr2 — Manages virtual IPv4 addresses (Linux specific version)

Synopsis

```
OCF_RESKEY_ip=string [OCF_RESKEY_nic=string]
[OCF_RESKEY_cidr_netmask=string] [OCF_RESKEY_broadcast=string]
[OCF_RESKEY_iflabel=string] [OCF_RESKEY_lvs_support=boolean]
[OCF_RESKEY_mac=string] [OCF_RESKEY_clusterip_hash=string]
[OCF_RESKEY_unique_clone_address=boolean]
[OCF_RESKEY_arp_interval=integer] [OCF_RESKEY_arp_count=integer]
[OCF_RESKEY_arp_bg=string] [OCF_RESKEY_arp_mac=string] IPAddr2 [start
| stop | status | monitor | meta-data | validate-all]
```

Description

This Linux-specific resource manages IP alias IP addresses. It can add an IP alias, or remove one. In addition, it can implement Cluster Alias IP functionality if invoked as a clone resource.

Supported Parameters

OCF_RESKEY_ip=IPv4 address

The IPv4 address to be configured in dotted quad notation, for example "192.168.1.1".

OCF_RESKEY_nic=Network interface

The base network interface on which the IP address will be brought online. If left empty, the script will try and determine this from the routing table. Do NOT specify an alias interface in the form eth0:1 or anything here; rather, specify the base interface only.

OCF_RESKEY_cidr_netmask=CIDR netmask

The netmask for the interface in CIDR format (e.g., 24 and not 255.255.255.0) If unspecified, the script will also try to determine this from the routing table.

OCF_RESKEY_broadcast=Broadcast address

Broadcast address associated with the IP. If left empty, the script will determine this from the netmask.

OCF_RESKEY_iflabel=Interface label

You can specify an additional label for your IP address here. This label is appended to your interface name. If a label is specified in nic name, this parameter has no effect.

OCF_RESKEY_lvs_support=Enable support for LVS DR

Enable support for LVS Direct Routing configurations. In case a IP address is stopped, only move it to the loopback device to allow the local node to continue to service requests, but no longer advertise it on the network.

OCF_RESKEY_mac=Cluster IP MAC address

Set the interface MAC address explicitly. Currently only used in case of the Cluster IP Alias. Leave empty to chose automatically.

OCF_RESKEY_clusterip_hash=Cluster IP hashing function

Specify the hashing algorithm used for the Cluster IP functionality.

OCF_RESKEY_unique_clone_address=Create a unique address for cloned instances

If true, add the clone ID to the supplied value of ip to create a unique address to manage

OCF_RESKEY_arp_interval=ARP packet interval in ms

Specify the interval between unsolicited ARP packets in milliseconds.

OCF_RESKEY_arp_count=ARP packet count

Number of unsolicited ARP packets to send.

OCF_RESKEY_arp_bg=ARP from background

Whether or not to send the arp packets in the background.

OCF_RESKEY_arp_mac=ARP MAC

MAC address to send the ARP packets too. You really shouldn't be touching this.

ocf:IPaddr (7)

ocf:IPaddr — Manages virtual IPv4 addresses (portable version)

Synopsis

```
OCF_RESKEY_ip=string [OCF_RESKEY_nic=string]
[OCF_RESKEY_cidr_netmask=string] [OCF_RESKEY_broadcast=string]
[OCF_RESKEY_iflabel=string] [OCF_RESKEY_lvs_support=boolean]
[OCF_RESKEY_local_stop_script=string]
[OCF_RESKEY_local_start_script=string]
[OCF_RESKEY_ARP_INTERVAL_MS=integer]
[OCF_RESKEY_ARP_REPEAT=integer]
[OCF_RESKEY_ARP_BACKGROUND=boolean]
[OCF_RESKEY_ARP_NETMASK=string] IPaddr [start | stop | monitor | validate-all
| meta-data]
```

Description

This script manages IP alias IP addresses It can add an IP alias, or remove one.

Supported Parameters

OCF_RESKEY_ip=IPv4 address

The IPv4 address to be configured in dotted quad notation, for example "192.168.1.1".

OCF_RESKEY_nic=Network interface

The base network interface on which the IP address will be brought online. If left empty, the script will try and determine this from the routing table. Do NOT specify an alias interface in the form eth0:1 or anything here; rather, specify the base interface only.

OCF_RESKEY_cidr_netmask=Netmask

The netmask for the interface in CIDR format. (ie, 24), or in dotted quad notation 255.255.255.0). If unspecified, the script will also try to determine this from the routing table.

OCF_RESKEY_broadcast=Broadcast address

Broadcast address associated with the IP. If left empty, the script will determine this from the netmask.

OCF_RESKEY_iflabel=Interface label

You can specify an additional label for your IP address here.

OCF_RESKEY_lvs_support=Enable support for LVS DR

Enable support for LVS Direct Routing configurations. In case a IP address is stopped, only move it to the loopback device to allow the local node to continue to service requests, but no longer advertise it on the network.

OCF_RESKEY_local_stop_script=Script called when the IP is released

Script called when the IP is released

OCF_RESKEY_local_start_script=Script called when the IP is added

Script called when the IP is added

OCF_RESKEY_ARP_INTERVAL_MS=milliseconds between gratuitous ARPs

milliseconds between ARPs

OCF_RESKEY_ARP_REPEAT=repeat count

How many gratuitous ARPs to send out when bringing up a new address

OCF_RESKEY_ARP_BACKGROUND=run in background

run in background (no longer any reason to do this)

OCF_RESKEY_ARP_NETMASK=netmask for ARP

netmask for ARP - in nonstandard hexadecimal format.

ocf:IPsrcaddr (7)

ocf:IPsrcaddr — Manages the preferred source address for outgoing IP packets

Synopsis

```
[OCF_RESKEY_ipaddress=string] IPsrcaddr [start | stop | stop | monitor |  
validate-all | meta-data]
```

Description

Resource script for IPsrcaddr. It manages the preferred source address modification.

Supported Parameters

OCF_RESKEY_ipaddress=IP address
The IP address.

ocf:IPv6addr (7)

ocf:IPv6addr — Manages IPv6 aliases

Synopsis

```
[OCF_RESKEY_ipv6addr=string] [OCF_RESKEY_cidr_netmask=string]  
[OCF_RESKEY_nic=string] IPv6addr [start | stop | status | monitor | validate-all |  
meta-data]
```

Description

This script manages IPv6 alias IPv6 addresses, It can add an IP6 alias, or remove one.

Supported Parameters

OCF_RESKEY_ipv6addr=IPv6 address
The IPv6 address this RA will manage

OCF_RESKEY_cidr_netmask=Netmask
The netmask for the interface in CIDR format. (ie, 24). The value of this parameter overwrites the value of `_prefix_` of `ipv6addr` parameter.

OCF_RESKEY_nic=Network interface
The base network interface on which the IPv6 address will be brought online.

ocf:iSCSILogicalUnit (7)

ocf:iSCSILogicalUnit — Manages iSCSI Logical Units (LUs)

Synopsis

```
[OCF_RESKEY_implementation=string] [OCF_RESKEY_target_iqn=string]  
[OCF_RESKEY_lun=integer] [OCF_RESKEY_path=string]  
OCF_RESKEY_scsi_id=string OCF_RESKEY_scsi_sn=string  
[OCF_RESKEY_vendor_id=string] [OCF_RESKEY_product_id=string]  
[OCF_RESKEY_additional_parameters=string] iSCSILogicalUnit [start  
| stop | monitor | meta-data | validate-all]
```

Description

Manages iSCSI Logical Unit. An iSCSI Logical unit is a subdivision of an SCSI Target, exported via a daemon that speaks the iSCSI protocol.

Supported Parameters

OCF_RESKEY_implementation=iSCSI target daemon implementation

The iSCSI target daemon implementation. Must be one of "iet", "tgt", or "lio". If unspecified, an implementation is selected based on the availability of management utilities, with "iet" being tried first, then "tgt", then "lio".

OCF_RESKEY_target_iqn=iSCSI target IQN

The iSCSI Qualified Name (IQN) that this Logical Unit belongs to.

OCF_RESKEY_lun=Logical Unit number (LUN)

The Logical Unit number (LUN) exposed to initiators.

OCF_RESKEY_path=Block device (or file) path

The path to the block device exposed. Some implementations allow this to be a regular file, too.

OCF_RESKEY_scsi_id=SCSI ID

The SCSI ID to be configured for this Logical Unit. The default is the resource name, truncated to 24 bytes.

OCF_RESKEY_scsi_sn=SCSI serial number

The SCSI serial number to be configured for this Logical Unit. The default is a hash of the resource name, truncated to 8 bytes.

OCF_RESKEY_vendor_id=SCSI vendor ID

The SCSI vendor ID to be configured for this Logical Unit.

OCF_RESKEY_product_id=SCSI product ID

The SCSI product ID to be configured for this Logical Unit.

OCF_RESKEY_additional_parameters=List of iSCSI LU parameters

Additional LU parameters. A space-separated list of "name=value" pairs which will be passed through to the iSCSI daemon's management interface. The supported parameters are implementation dependent. Neither the name nor the value may contain whitespace.

ocf:iSCSITarget (7)

ocf:iSCSITarget — iSCSI target export agent

Synopsis

```
[OCF_RESKEY_implementation=string] OCF_RESKEY_iqn=string  
OCF_RESKEY_tid=integer [OCF_RESKEY_portals=string]  
[OCF_RESKEY_allowed_initiators=string]  
OCF_RESKEY_incoming_username=string  
[OCF_RESKEY_incoming_password=string]  
[OCF_RESKEY_additional_parameters=string] iSCSITarget [start | stop  
| monitor | meta-data | validate-all]
```

Description

Manages iSCSI targets. An iSCSI target is a collection of SCSI Logical Units (LUs) exported via a daemon that speaks the iSCSI protocol.

Supported Parameters

`OCF_RESKEY_implementation=`Manages an iSCSI target export

The iSCSI target daemon implementation. Must be one of "iet", "tgt", or "lio". If unspecified, an implementation is selected based on the availability of management utilities, with "iet" being tried first, then "tgt", then "lio".

`OCF_RESKEY_iqn=`iSCSI target IQN

The target iSCSI Qualified Name (IQN). Should follow the conventional "iqn.yyyy-mm.<reversed domain name>[:identifier]" syntax.

`OCF_RESKEY_tid=`iSCSI target ID

The iSCSI target ID. Required for tgt.

OCF_RESKEY_portals=iSCSI portal addresses

iSCSI network portal addresses. Not supported by all implementations. If unset, the default is to create one portal that listens on .

OCF_RESKEY_allowed_initiators=List of iSCSI initiators allowed to connect to this target

Allowed initiators. A space-separated list of initiators allowed to connect to this target. Initiators may be listed in any syntax the target implementation allows. If this parameter is empty or not set, access to this target will be allowed from any initiator.

OCF_RESKEY_incoming_username=Incoming account username

A username used for incoming initiator authentication. If unspecified, allowed initiators will be able to log in without authentication.

OCF_RESKEY_incoming_password=Incoming account password

A password used for incoming initiator authentication.

OCF_RESKEY_additional_parameters=List of iSCSI target parameters

Additional target parameters. A space-separated list of "name=value" pairs which will be passed through to the iSCSI daemon's management interface. The supported parameters are implementation dependent. Neither the name nor the value may contain whitespace.

ocf:iscsi (7)

ocf:iscsi — Manages a local iSCSI initiator and its connections to iSCSI targets

Synopsis

```
[OCF_RESKEY_portal=string] OCF_RESKEY_target=string  
[OCF_RESKEY_discovery_type=string] [OCF_RESKEY_iscsiadm=string]  
[OCF_RESKEY_udev=string] iscsi [start | stop | status | monitor | validate-all |  
methods | meta-data]
```

Description

OCF Resource Agent for iSCSI. Add (start) or remove (stop) iSCSI targets.

Supported Parameters

OCF_RESKEY_portal=portal

The iSCSI portal address in the form: {ip_address|hostname}[:"port"]

OCF_RESKEY_target=target

The iSCSI target.

OCF_RESKEY_discovery_type=discovery_type

Discovery type. Currently, with open-iscsi, only the sendtargets type is supported.

OCF_RESKEY_iscsiadm=iscsiadm

iscsiadm program path.

OCF_RESKEY_udev=udev

If the next resource depends on the udev creating a device then we wait until it is finished. On a normally loaded host this should be done quickly, but you may be unlucky. If you are not using udev set this to "no", otherwise we will spin in a loop until a timeout occurs.

ocf:ldirectord (7)

ocf:ldirectord — Wrapper OCF Resource Agent for ldirectord

Synopsis

```
OCF_RESKEY_configfile=string [OCF_RESKEY_ldirectord=string]  
ldirectord [start | stop | monitor | meta-data | validate-all]
```

Description

It's a simple OCF RA wrapper for ldirectord and uses the ldirectord interface to create the OCF compliant interface. You win monitoring of ldirectord. Be warned: Asking ldirectord status is an expensive action.

Supported Parameters

OCF_RESKEY_configfile=configuration file path
The full pathname of the ldirectord configuration file.

OCF_RESKEY_ldirectord=ldirectord binary path
The full pathname of the ldirectord.

ocf:LinuxSCSI (7)

ocf:LinuxSCSI — Enables and disables SCSI devices through the kernel SCSI hot-plug subsystem (deprecated)

Synopsis

```
[OCF_RESKEY_scsi=string] [OCF_RESKEY_ignore_deprecation=boolean]  
LinuxSCSI [start | stop | methods | status | monitor | meta-data | validate-all]
```

Description

Deprecation warning: This agent makes use of Linux SCSI hot-plug functionality which has been superseded by SCSI reservations. It is deprecated and may be removed from a future release. See the `scsi2reservation` and `sfex` agents for alternatives. -- This is a resource agent for LinuxSCSI. It manages the availability of a SCSI device from the point of view of the linux kernel. It make Linux believe the device has gone away, and it can make it come back again.

Supported Parameters

`OCF_RESKEY_scsi=SCSI instance`
The SCSI instance to be managed.

`OCF_RESKEY_ignore_deprecation=Suppress deprecation warning`
If set to true, suppresses the deprecation warning for this agent.

ocf:LVM (7)

ocf:LVM — Controls the availability of an LVM Volume Group

Synopsis

```
[OCF_RESKEY_volgrpname=string] [OCF_RESKEY_exclusive=string] LVM  
[start | stop | status | monitor | methods | meta-data | validate-all]
```

Description

Resource script for LVM. It manages an Linux Volume Manager volume (LVM) as an HA resource.

Supported Parameters

OCF_RESKEY_volgrpname=Volume group name
The name of volume group.

OCF_RESKEY_exclusive=Exclusive activation
If set, the volume group will be activated exclusively.

ocf:MailTo (7)

ocf:MailTo — Notifies recipients by email in the event of resource takeover

Synopsis

```
[OCF_RESKEY_email=string] [OCF_RESKEY_subject=string] MailTo [start |  
stop | status | monitor | meta-data | validate-all]
```

Description

This is a resource agent for MailTo. It sends email to a sysadmin whenever a takeover occurs.

Supported Parameters

OCF_RESKEY_email=Email address
The email address of sysadmin.

OCF_RESKEY_subject=Subject
The subject of the email.

ocf:ManageRAID (7)

ocf:ManageRAID — Manages RAID devices

Synopsis

[OCF_RESKEY_raidname=string] ManageRAID [start | stop | status | monitor |
validate-all | meta-data]

Description

Manages starting, stopping and monitoring of RAID devices which are preconfigured in /etc/conf.d/HB-ManageRAID.

Supported Parameters

OCF_RESKEY_raidname=RAID name

Name (case sensitive) of RAID to manage. (preconfigured in /etc/conf.d/HB-
ManageRAID)

ocf:ManageVE (7)

ocf:ManageVE — Manages an OpenVZ Virtual Environment (VE)

Synopsis

```
[OCF_RESKEY_veid=integer] ManageVE [start | stop | status | monitor | validate-all  
| meta-data]
```

Description

This OCF complaint resource agent manages OpenVZ VEs and thus requires a proper OpenVZ installation including a recent vzctl util.

Supported Parameters

OCF_RESKEY_veid=OpenVZ ID of VE

OpenVZ ID of virtual environment (see output of vzlist -a for all assigned IDs)

ocf:mysql-proxy (7)

ocf:mysql-proxy — Manages a MySQL Proxy daemon

Synopsis

```
[OCF_RESKEY_binary=string] OCF_RESKEY_defaults_file=string
[OCF_RESKEY_proxy_backend_addresses=string]
[OCF_RESKEY_proxy_read_only_backend_addresses=string]
[OCF_RESKEY_proxy_address=string] [OCF_RESKEY_log_level=string]
[OCF_RESKEY_heartbeat=string] [OCF_RESKEY_admin_address=string]
[OCF_RESKEY_admin_username=string]
[OCF_RESKEY_admin_password=string]
[OCF_RESKEY_admin_lua_script=string]
[OCF_RESKEY_parameters=string] OCF_RESKEY_pidfile=string
mysql-proxy [start | stop | reload | monitor | validate-all | meta-data]
```

Description

This script manages MySQL Proxy as an OCF resource in a high-availability setup. Tested with MySQL Proxy 0.7.0 on Debian 5.0.

Supported Parameters

OCF_RESKEY_binary=Full path to MySQL Proxy binary
Full path to the MySQL Proxy binary. For example, "/usr/sbin/mysql-proxy".

OCF_RESKEY_defaults_file=Full path to configuration file
Full path to a MySQL Proxy configuration file. For example, "/etc/mysql-proxy.conf".

OCF_RESKEY_proxy_backend_addresses=MySQL Proxy backend-servers
Address:port of the remote backend-servers (default: 127.0.0.1:3306).

OCF_RESKEY_proxy_read_only_backend_addresses=MySQL Proxy read only backend-servers

Address:port of the remote (read only) slave-server (default:).

OCF_RESKEY_proxy_address=MySQL Proxy listening address

Listening address:port of the proxy-server (default: :4040). You can also specify a socket like "/tmp/mysql-proxy.sock".

OCF_RESKEY_log_level=MySQL Proxy log level.

Log all messages of level (error|warning|info|message|debug|) or higher. An empty value disables logging.

OCF_RESKEY_keepalive=Use keepalive option

Try to restart the proxy if it crashed (default:). Valid values: true or false. An empty value equals "false".

OCF_RESKEY_admin_address=MySQL Proxy admin-server address

Listening address:port of the admin-server (default: 127.0.0.1:4041).

OCF_RESKEY_admin_username=MySQL Proxy admin-server username

Username to allow to log in (default:).

OCF_RESKEY_admin_password=MySQL Proxy admin-server password

Password to allow to log in (default:).

OCF_RESKEY_admin_lua_script=MySQL Proxy admin-server lua script

Script to execute by the admin plugin.

OCF_RESKEY_parameters=MySQL Proxy additional parameters

The MySQL Proxy daemon may be called with additional parameters. Specify any of them here.

OCF_RESKEY_pidfile=PID file

PID file

ocf:mysql (7)

ocf:mysql — Manages a MySQL database instance

Synopsis

```
[OCF_RESKEY_binary=string] [OCF_RESKEY_config=string]
[OCF_RESKEY_datadir=string] [OCF_RESKEY_user=string]
[OCF_RESKEY_group=string] [OCF_RESKEY_log=string]
[OCF_RESKEY_pid=string] [OCF_RESKEY_socket=string]
[OCF_RESKEY_test_table=string] [OCF_RESKEY_test_user=string]
[OCF_RESKEY_test_passwd=string]
[OCF_RESKEY_enable_creation=integer]
[OCF_RESKEY_additional_parameters=string]
[OCF_RESKEY_replication_user=string]
[OCF_RESKEY_replication_passwd=string] mysql [start | stop | status | monitor
| monitor | monitor | notify | promote | demote | validate-all | meta-data]
```

Description

Resource script for MySQL. It manages a MySQL Database instance as an HA resource.

Supported Parameters

OCF_RESKEY_binary=MySQL binary
Location of the MySQL binary

OCF_RESKEY_config=MySQL config
Configuration file

OCF_RESKEY_datadir=MySQL datadir
Directory containing databases

OCF_RESKEY_user=MySQL user
User running MySQL daemon

OCF_RESKEY_group=MySQL group
Group running MySQL daemon (for logfile and directory permissions)

OCF_RESKEY_log=MySQL log file
The logfile to be used for mysqld.

OCF_RESKEY_pid=MySQL pid file
The pidfile to be used for mysqld.

OCF_RESKEY_socket=MySQL socket
The socket to be used for mysqld.

OCF_RESKEY_test_table=MySQL test table
Table to be tested in monitor statement (in database.table notation)

OCF_RESKEY_test_user=MySQL test user
MySQL test user

OCF_RESKEY_test_passwd=MySQL test user password
MySQL test user password

OCF_RESKEY_enable_creation=Create the database if it does not exist
If the MySQL database does not exist, it will be created

OCF_RESKEY_additional_parameters=Additional parameters to pass to mysqld
Additional parameters which are passed to the mysqld on startup. (e.g. --skip-external-locking or --skip-grant-tables)

OCF_RESKEY_replication_user=MySQL replication user
MySQL replication user. Used for replication client and slave.

OCF_RESKEY_replication_passwd=MySQL replication user password
MySQL replication password. Used for replication client and slave.

ocf:nfsserver (7)

ocf:nfsserver — Manages an NFS server

Synopsis

```
[OCF_RESKEY_nfs_init_script=string]  
[OCF_RESKEY_nfs_notify_cmd=string]  
[OCF_RESKEY_nfs_shared_infodir=string] [OCF_RESKEY_nfs_ip=string]  
nfsserver [start | stop | monitor | meta-data | validate-all]
```

Description

Nfsserver helps to manage the Linux nfs server as a failover-able resource in Linux-HA. It depends on Linux specific NFS implementation details, so is considered not portable to other platforms yet.

Supported Parameters

OCF_RESKEY_nfs_init_script= Init script for nfsserver

The default init script shipped with the Linux distro. The nfsserver resource agent offloads the start/stop/monitor work to the init script because the procedure to start/stop/monitor nfsserver varies on different Linux distro.

OCF_RESKEY_nfs_notify_cmd= The tool to send out notification.

The tool to send out NSM reboot notification. Failover of nfsserver can be considered as rebooting to different machines. The nfsserver resource agent use this command to notify all clients about the happening of failover.

OCF_RESKEY_nfs_shared_infodir= Directory to store nfs server related information.

The nfsserver resource agent will save nfs related information in this specific directory. And this directory must be able to fail-over before nfsserver itself.

OCF_RESKEY_nfs_ip= IP address.

The floating IP address used to access the nfs service

ocf:oracle (7)

ocf:oracle — Manages an Oracle Database instance

Synopsis

```
OCF_RESKEY_sid=string [OCF_RESKEY_home=string]
[OCF_RESKEY_user=string] [OCF_RESKEY_ipcrm=string]
[OCF_RESKEY_clear_backupmode=boolean]
[OCF_RESKEY_shutdown_method=string] oracle [start | stop | status | monitor
| validate-all | methods | meta-data]
```

Description

Resource script for oracle. Manages an Oracle Database instance as an HA resource.

Supported Parameters

OCF_RESKEY_sid=sid
The Oracle SID (aka ORACLE_SID).

OCF_RESKEY_home=home
The Oracle home directory (aka ORACLE_HOME). If not specified, then the SID along with its home should be listed in /etc/oratab.

OCF_RESKEY_user=user
The Oracle owner (aka ORACLE_OWNER). If not specified, then it is set to the owner of file \$ORACLE_HOME/dbs/*\${ORACLE_SID}.ora. If this does not work for you, just set it explicitly.

OCF_RESKEY_ipcrm=ipcrm
Sometimes IPC objects (shared memory segments and semaphores) belonging to an Oracle instance might be left behind which prevents the instance from starting. It is not easy to figure out which shared segments belong to which instance, in particular when more instances are running as same user. What we use here is the

"oradebug" feature and its "ipc" trace utility. It is not optimal to parse the debugging information, but I am not aware of any other way to find out about the IPC information. In case the format or wording of the trace report changes, parsing might fail. There are some precautions, however, to prevent stepping on other peoples toes. There is also a dumpinstipc option which will make us print the IPC objects which belong to the instance. Use it to see if we parse the trace file correctly. Three settings are possible: - none: don't mess with IPC and hope for the best (beware: you'll probably be out of luck, sooner or later) - instance: try to figure out the IPC stuff which belongs to the instance and remove only those (default; should be safe) - orauser: remove all IPC belonging to the user which runs the instance (don't use this if you run more than one instance as same user or if other apps running as this user use IPC) The default setting "instance" should be safe to use, but in that case we cannot guarantee that the instance will start. In case IPC objects were already left around, because, for instance, someone mercilessly killing Oracle processes, there is no way any more to find out which IPC objects should be removed. In that case, human intervention is necessary, and probably all instances running as same user will have to be stopped. The third setting, "orauser", guarantees IPC objects removal, but it does that based only on IPC objects ownership, so you should use that only if every instance runs as separate user. Please report any problems. Suggestions/fixes welcome.

```
OCF_RESKEY_clear_backupmode=clear_backupmode
```

The clear of the backup mode of ORACLE.

```
OCF_RESKEY_shutdown_method=shutdown_method
```

How to stop Oracle is a matter of taste it seems. The default method ("checkpoint/abort") is: alter system checkpoint; shutdown abort; This should be the fastest safe way bring the instance down. If you find "shutdown abort" distasteful, set this attribute to "immediate" in which case we will shutdown immediate; If you still think that there's even better way to shutdown an Oracle instance we are willing to listen.

ocf:oralsnr (7)

ocf:oralsnr — Manages an Oracle TNS listener

Synopsis

```
OCF_RESKEY_sid=string [OCF_RESKEY_home=string]  
[OCF_RESKEY_user=string] OCF_RESKEY_listener=string oralsnr [start |  
stop | status | monitor | validate-all | meta-data | methods]
```

Description

Resource script for Oracle Listener. It manages an Oracle Listener instance as an HA resource.

Supported Parameters

OCF_RESKEY_sid=sid

The Oracle SID (aka ORACLE_SID). Necessary for the monitor op, i.e. to do tnsping SID.

OCF_RESKEY_home=home

The Oracle home directory (aka ORACLE_HOME). If not specified, then the SID should be listed in /etc/oratab.

OCF_RESKEY_user=user

Run the listener as this user.

OCF_RESKEY_listener=listener

Listener instance to be started (as defined in listener.ora). Defaults to LISTENER.

ocf:pgsql (7)

ocf:pgsql — Manages a PostgreSQL database instance

Synopsis

```
[OCF_RESKEY_pgctl=string] [OCF_RESKEY_start_opt=string]
[OCF_RESKEY_ctl_opt=string] [OCF_RESKEY_psql=string]
[OCF_RESKEY_pgdata=string] [OCF_RESKEY_pgdba=string]
[OCF_RESKEY_pghost=string] [OCF_RESKEY_pgport=string]
[OCF_RESKEY_pgdb=string] [OCF_RESKEY_logfile=string]
[OCF_RESKEY_stop_escalate=string] psql [start | stop | status | monitor |
meta-data | validate-all | methods]
```

Description

Resource script for PostgreSQL. It manages a PostgreSQL as an HA resource.

Supported Parameters

OCF_RESKEY_pgctl=pgctl
Path to pg_ctl command.

OCF_RESKEY_start_opt=start_opt
Start options (-o start_opt in pgi_ctl). "-i -p 5432" for example.

OCF_RESKEY_ctl_opt=ctl_opt
Additional pg_ctl options (-w, -W etc..). Default is ""

OCF_RESKEY_psql=psql
Path to psql command.

OCF_RESKEY_pgdata=pgdata
Path PostgreSQL data directory.

OCF_RESKEY_pgdba=pgdba
User that owns PostgreSQL.

OCF_RESKEY_pghost=pghost
Hostname/IP Address where PostgreSQL is listening

OCF_RESKEY_pgport=pgport
Port where PostgreSQL is listening

OCF_RESKEY_pgdb=pgdb
Database that will be used for monitoring.

OCF_RESKEY_logfile=logfile
Path to PostgreSQL server log output file.

OCF_RESKEY_stop_escalate=stop escalation
Number of retries (using -m fast) before resorting to -m immediate

ocf:pingd (7)

ocf:pingd — Monitors connectivity to specific hosts or IP addresses ("ping nodes")
(deprecated)

Synopsis

```
[OCF_RESKEY_pidfile=string] [OCF_RESKEY_user=string]  
[OCF_RESKEY_dampen=integer] [OCF_RESKEY_set=integer]  
[OCF_RESKEY_name=integer] [OCF_RESKEY_section=integer]  
[OCF_RESKEY_multiplier=integer] [OCF_RESKEY_host_list=integer]  
[OCF_RESKEY_ignore_deprecation=boolean] pingd [start | stop | monitor |  
meta-data | validate-all]
```

Description

Deprecation warning: This agent is deprecated and may be removed from a future release. See the ocf:pacemaker:pingd resource agent for a supported alternative. -- This is a pingd Resource Agent. It records (in the CIB) the current number of ping nodes a node can connect to.

Supported Parameters

OCF_RESKEY_pidfile=PID file
PID file

OCF_RESKEY_user=The user we want to run pingd as
The user we want to run pingd as

OCF_RESKEY_dampen=Dampening interval
The time to wait (dampening) further changes occur

OCF_RESKEY_set=Set name
The name of the instance_attributes set to place the value in. Rarely needs to be specified.

OCF_RESKEY_name=Attribute name

The name of the attributes to set. This is the name to be used in the constraints.

OCF_RESKEY_section=Section name

The section place the value in. Rarely needs to be specified.

OCF_RESKEY_multiplier=Value multiplier

The number by which to multiply the number of connected ping nodes by

OCF_RESKEY_host_list=Host list

The list of ping nodes to count. Defaults to all configured ping nodes. Rarely needs to be specified.

OCF_RESKEY_ignore_deprecation=Suppress deprecation warning

If set to true, suppresses the deprecation warning for this agent.

ocf:portblock (7)

ocf:portblock — Block and unblocks access to TCP and UDP ports

Synopsis

```
[OCF_RESKEY_protocol=string] [OCF_RESKEY_portno=integer]  
[OCF_RESKEY_action=string] [OCF_RESKEY_ip=string]  
[OCF_RESKEY_tickle_dir=string] [OCF_RESKEY_sync_script=string]  
portblock [start | stop | status | monitor | meta-data | validate-all]
```

Description

Resource script for portblock. It is used to temporarily block ports using iptables. In addition, it may allow for faster TCP reconnects for clients on failover. Use that if there are long lived TCP connections to an HA service. This feature is enabled by setting the tickle_dir parameter and only in concert with action set to unblock. Note that the tickle ACK function is new as of version 3.0.2 and hasn't yet seen widespread use.

Supported Parameters

OCF_RESKEY_protocol=protocol
The protocol used to be blocked/unblocked.

OCF_RESKEY_portno=portno
The port number used to be blocked/unblocked.

OCF_RESKEY_action=action
The action (block/unblock) to be done on the protocol::portno.

OCF_RESKEY_ip=ip
The IP address used to be blocked/unblocked.

OCF_RESKEY_tickle_dir=Tickle directory

The shared or local directory (must be absolute path) which stores the established TCP connections.

OCF_RESKEY_sync_script=Connection state file synchronization script

If the tickle_dir is a local directory, then the TCP connection state file has to be replicated to other nodes in the cluster. It can be csync2 (default), some wrapper of rsync, or whatever. It takes the file name as a single argument. For csync2, set it to "csync2 -xv".

ocf:proftpd (7)

ocf:proftpd — OCF Resource Agent compliant FTP script.

Synopsis

```
[OCF_RESKEY_binary=string] [OCF_RESKEY_confdir=string]
[OCF_RESKEY_pidfile=string] [OCF_RESKEY_curl_binary=string]
[OCF_RESKEY_curl_url=string] [OCF_RESKEY_test_user=string]
[OCF_RESKEY_test_pass=string] proftpd [start | stop | monitor | monitor |
validate-all | meta-data]
```

Description

This script manages Proftpd in an Active-Passive setup

Supported Parameters

OCF_RESKEY_binary=The Proftpd binary
The Proftpd binary

OCF_RESKEY_confdir=Configuration file name with full path
The Proftpd configuration file name with full path. For example, "/etc/proftpd.conf"

OCF_RESKEY_pidfile=PID file
The Proftpd PID file. The location of the PID file is configured in the Proftpd configuration file.

OCF_RESKEY_curl_binary=The absolut path to the curl binary
The absolut path to the curl binary for monitoring with OCF_CHECK_LEVEL greater zero.

OCF_RESKEY_curl_url=The URL which is checked by curl
The URL which is checked by curl with OCF_CHECK_LEVEL greater zero.

OCF_RESKEY_test_user=The name of the ftp user

The name of the ftp user for monitoring with OCF_CHECK_LEVEL greater zero.

OCF_RESKEY_test_pass=The password of the ftp user

The password of the ftp user for monitoring with OCF_CHECK_LEVEL greater zero.

ocf:Pure-FTPd (7)

ocf:Pure-FTPd — Manages a Pure-FTPd FTP server instance

Synopsis

```
OCF_RESKEY_script=string OCF_RESKEY_conffile=string  
OCF_RESKEY_daemon_type=string [OCF_RESKEY_pidfile=string]  
Pure-FTPd [start | stop | monitor | validate-all | meta-data]
```

Description

This script manages Pure-FTPd in an Active-Passive setup

Supported Parameters

OCF_RESKEY_script=Script name with full path
The full path to the Pure-FTPd startup script. For example, "/sbin/pure-config.pl"

OCF_RESKEY_conffile=Configuration file name with full path
The Pure-FTPd configuration file name with full path. For example, "/etc/pure-ftp/pure-ftp.conf"

OCF_RESKEY_daemon_type=Configuration file name with full path
The Pure-FTPd daemon to be called by pure-ftp-wrapper. Valid options are "" for pure-ftp, "mysql" for pure-ftp-mysql, "postgresql" for pure-ftp-postgresql and "ldap" for pure-ftp-ldap

OCF_RESKEY_pidfile=PID file
PID file

ocf:Raid1 (7)

ocf:Raid1 — Manages a software RAID1 device on shared storage

Synopsis

```
[OCF_RESKEY_raidconf=string] [OCF_RESKEY_raiddev=string]  
[OCF_RESKEY_homehost=string] Raid1 [start | stop | status | monitor | validate-  
all | meta-data]
```

Description

Resource script for RAID1. It manages a software Raid1 device on a shared storage medium.

Supported Parameters

OCF_RESKEY_raidconf=RAID config file
The RAID configuration file. e.g. /etc/raidtab or /etc/mdadm.conf.

OCF_RESKEY_raiddev=block device
The block device to use.

OCF_RESKEY_homehost=Homehost for mdadm
The value for the homehost directive; this is an mdadm feature to protect RAIDs against being activated by accident. It is recommended to create RAIDs managed by the cluster with "homehost" set to a special value, so they are not accidentally auto-assembled by nodes not supposed to own them.

ocf:Route (7)

ocf:Route — Manages network routes

Synopsis

```
OCF_RESKEY_destination=string OCF_RESKEY_device=string  
OCF_RESKEY_gateway=string OCF_RESKEY_source=string  
[OCF_RESKEY_table=string] Route [start | stop | monitor | reload | meta-data |  
validate-all]
```

Description

Enables and disables network routes. Supports host and net routes, routes via a gateway address, and routes using specific source addresses. This resource agent is useful if a node's routing table needs to be manipulated based on node role assignment. Consider the following example use case: - One cluster node serves as an IPsec tunnel endpoint. - All other nodes use the IPsec tunnel to reach hosts in a specific remote network. Then, here is how you would implement this scheme making use of the Route resource agent: - Configure an ipsec LSB resource. - Configure a cloned Route OCF resource. - Create an order constraint to ensure that ipsec is started before Route. - Create a colocation constraint between the ipsec and Route resources, to make sure no instance of your cloned Route resource is started on the tunnel endpoint itself.

Supported Parameters

`OCF_RESKEY_destination=Destination network`

The destination network (or host) to be configured for the route. Specify the netmask suffix in CIDR notation (e.g. "/24"). If no suffix is given, a host route will be created. Specify "0.0.0.0/0" or "default" if you want this resource to set the system default route.

`OCF_RESKEY_device=Outgoing network device`

The outgoing network device to use for this route.

OCF_RESKEY_gateway=Gateway IP address
The gateway IP address to use for this route.

OCF_RESKEY_source=Source IP address
The source IP address to be configured for the route.

OCF_RESKEY_table=Routing table
The routing table to be configured for the route.

ocf:rsyncd (7)

ocf:rsyncd — Manages an rsync daemon

Synopsis

```
[OCF_RESKEY_binpath=string] [OCF_RESKEY_conf file=string]  
[OCF_RESKEY_bwlimit=string] rsyncd [start | stop | monitor | validate-all | meta-  
data]
```

Description

This script manages rsync daemon

Supported Parameters

OCF_RESKEY_binpath=Full path to the rsync binary
The rsync binary path. For example, "/usr/bin/rsync"

OCF_RESKEY_conf file=Configuration file name with full path
The rsync daemon configuration file name with full path. For example,
"/etc/rsyncd.conf"

OCF_RESKEY_bwlimit=limit I/O bandwidth, KBytes per second
This option allows you to specify a maximum transfer rate in kilobytes per second.
This option is most effective when using rsync with large files (several megabytes
and up). Due to the nature of rsync transfers, blocks of data are sent, then if rsync
determines the transfer was too fast, it will wait before sending the next data block.
The result is an average transfer rate equaling the specified limit. A value of zero
specifies no limit.

ocf:SAPDatabase (7)

ocf:SAPDatabase — Manages any SAP database (based on Oracle, MaxDB, or DB2)

Synopsis

```
OCF_RESKEY_SID=string OCF_RESKEY_DIR_EXECUTABLE=string
OCF_RESKEY_DBTYPE=string OCF_RESKEY_NETSERVICENAME=string
OCF_RESKEY_DBJ2EE_ONLY=boolean OCF_RESKEY_JAVA_HOME=string
OCF_RESKEY_STRICT_MONITORING=boolean
OCF_RESKEY_AUTOMATIC_RECOVER=boolean
OCF_RESKEY_DIR_BOOTSTRAP=string OCF_RESKEY_DIR_SECSTORE=string
OCF_RESKEY_DB_JARS=string OCF_RESKEY_PRE_START_USEREXIT=string
OCF_RESKEY_POST_START_USEREXIT=string
OCF_RESKEY_PRE_STOP_USEREXIT=string
OCF_RESKEY_POST_STOP_USEREXIT=string SAPDatabase [start | stop | status
| monitor | validate-all | meta-data | methods]
```

Description

Resource script for SAP databases. It manages a SAP database of any type as an HA resource.

Supported Parameters

OCF_RESKEY_SID=SAP system ID

The unique SAP system identifier. e.g. P01

OCF_RESKEY_DIR_EXECUTABLE=path of sapstartsrv and sapcontrol

The full qualified path where to find sapstartsrv and sapcontrol.

OCF_RESKEY_DBTYPE=database vendor

The name of the database vendor you use. Set either: ORA,DB6,ADA

OCF_RESKEY_NETSERVICENAME=listener name

The Oracle TNS listener name.

OCF_RESKEY_DBJ2EE_ONLY=only JAVA stack installed

If you do not have a ABAP stack installed in the SAP database, set this to TRUE

OCF_RESKEY_JAVA_HOME=Path to Java SDK

This is only needed if the DBJ2EE_ONLY parameter is set to true. Enter the path to the Java SDK which is used by the SAP WebAS Java

OCF_RESKEY_STRICT_MONITORING=Activates application level monitoring

This controls how the resource agent monitors the database. If set to true, it will use SAP tools to test the connect to the database. Do not use with Oracle, because it will result in unwanted failovers in case of an archiver stuck

OCF_RESKEY_AUTOMATIC_RECOVER=Enable or disable automatic startup recovery

The SAPDatabase resource agent tries to recover a failed start attempt automatically one time. This is done by running a forced abort of the RDBMS and/or executing recovery commands.

OCF_RESKEY_DIR_BOOTSTRAP=path to j2ee bootstrap directory

The full qualified path where to find the J2EE instance bootstrap directory. e.g.
/usr/sap/P01/J00/j2ee/cluster/bootstrap

OCF_RESKEY_DIR_SECSTORE=path to j2ee secure store directory

The full qualified path where to find the J2EE security store directory. e.g.
/usr/sap/P01/SYS/global/security/lib/tools

OCF_RESKEY_DB_JARS=file name of the jdbc driver

The full qualified filename of the jdbc driver for the database connection test. It will be automatically read from the bootstrap.properties file in Java engine 6.40 and 7.00. For Java engine 7.10 the parameter is mandatory.

OCF_RESKEY_PRE_START_USEREXIT=path to a pre-start script

The full qualified path where to find a script or program which should be executed before this resource gets started.

OCF_RESKEY_POST_START_USEREXIT=path to a post-start script

The full qualified path where to find a script or program which should be executed after this resource got started.

OCF_RESKEY_PRE_STOP_USEREXIT=path to a pre-start script

The full qualified path where to find a script or program which should be executed before this resource gets stopped.

OCF_RESKEY_POST_STOP_USEREXIT=path to a post-start script

The full qualified path where to find a script or program which should be executed after this resource got stopped.

ocf:SAPInstance (7)

ocf:SAPInstance — Manages a SAP instance

Synopsis

```
OCF_RESKEY_InstanceName=string OCF_RESKEY_DIR_EXECUTABLE=string
OCF_RESKEY_DIR_PROFILE=string OCF_RESKEY_START_PROFILE=string
OCF_RESKEY_START_WAITTIME=string
OCF_RESKEY_AUTOMATIC_RECOVER=boolean
OCF_RESKEY_MONITOR_SERVICES=string
OCF_RESKEY_ERS_InstanceName=string
OCF_RESKEY_ERS_START_PROFILE=string
OCF_RESKEY_PRE_START_USEREXIT=string
OCF_RESKEY_POST_START_USEREXIT=string
OCF_RESKEY_PRE_STOP_USEREXIT=string
OCF_RESKEY_POST_STOP_USEREXIT=string SAPInstance [start | stop | status
| monitor | promote | demote | validate-all | meta-data | methods]
```

Description

Resource script for SAP. It manages a SAP Instance as an HA resource.

Supported Parameters

OCF_RESKEY_InstanceName=**instance name: SID_INSTANCE_VIR-HOSTNAME**
The full qualified SAP instance name. e.g. P01_DVEBMGS00_sapp01ci

OCF_RESKEY_DIR_EXECUTABLE=**path of sapstartsrv and sapcontrol**
The full qualified path where to find sapstartsrv and sapcontrol.

OCF_RESKEY_DIR_PROFILE=**path of start profile**
The full qualified path where to find the SAP START profile.

OCF_RESKEY_START_PROFILE=start profile name

The name of the SAP START profile.

OCF_RESKEY_START_WAITTIME=Check the successful start after that time (do not wait for J2EE-Addin)

After that time in seconds a monitor operation is executed by the resource agent.

Does the monitor return SUCCESS, the start is handled as SUCCESS. This is useful to resolve timing problems with e.g. the J2EE-Addin instance.

OCF_RESKEY_AUTOMATIC_RECOVER=Enable or disable automatic startup recovery

The SAPInstance resource agent tries to recover a failed start attempt automatically one time. This is done by killing running instance processes and executing cleanipc.

OCF_RESKEY_MONITOR_SERVICES=

OCF_RESKEY_ERS_InstanceName=

OCF_RESKEY_ERS_START_PROFILE=

OCF_RESKEY_PRE_START_USEREXIT=path to a pre-start script

The full qualified path where to find a script or program which should be executed before this resource gets started.

OCF_RESKEY_POST_START_USEREXIT=path to a post-start script

The full qualified path where to find a script or program which should be executed after this resource got started.

OCF_RESKEY_PRE_STOP_USEREXIT=path to a pre-stop script

The full qualified path where to find a script or program which should be executed before this resource gets stopped.

OCF_RESKEY_POST_STOP_USEREXIT=path to a post-stop script

The full qualified path where to find a script or program which should be executed after this resource got stopped.

ocf:scsi2reservation (7)

ocf:scsi2reservation — scsi-2 reservation

Synopsis

```
[OCF_RESKEY_scsi_reserve=string] [OCF_RESKEY_sharedisk=string]  
[OCF_RESKEY_start_loop=string] scsi2reservation [start | stop | monitor  
| meta-data | validate-all]
```

Description

The scsi-2-reserve resource agent is a place holder for SCSI-2 reservation. A healthy instance of scsi-2-reserve resource, indicates the own of the specified SCSI device. This resource agent depends on the scsi_reserve from scsires package, which is Linux specific.

Supported Parameters

OCF_RESKEY_scsi_reserve=Manages exclusive access to shared storage media through SCSI-2 reservations

The `scsi_reserve` is a command from scsires package. It helps to issue SCSI-2 reservation on SCSI devices.

OCF_RESKEY_sharedisk= Shared disk.

The shared disk that can be reserved.

OCF_RESKEY_start_loop= Times to re-try before giving up.

We are going to try several times before giving up. `Start_loop` indicates how many times we are going to re-try.

ocf:SendArp (7)

ocf:SendArp — Broadcasts unsolicited ARP announcements

Synopsis

```
[OCF_RESKEY_ip=string] [OCF_RESKEY_nic=string] SendArp [start | stop |  
monitor | meta-data | validate-all]
```

Description

This script send out gratuitous Arp for an IP address

Supported Parameters

OCF_RESKEY_ip=IP address

The IP address for sending arp package.

OCF_RESKEY_nic=NIC

The nic for sending arp package.

ocf:ServeRAID (7)

ocf:ServeRAID — Enables and disables shared ServeRAID merge groups

Synopsis

```
[OCF_RESKEY_serveraid=integer] [OCF_RESKEY_mergegroup=integer]  
ServeRAID [start | stop | status | monitor | validate-all | meta-data | methods]
```

Description

Resource script for ServeRAID. It enables/disables shared ServeRAID merge groups.

Supported Parameters

OCF_RESKEY_serveraid=serveraid
The adapter number of the ServeRAID adapter.

OCF_RESKEY_mergegroup=mergegroup
The logical drive under consideration.

ocf:sfex (7)

ocf:sfex — Manages exclusive access to shared storage using Shared Disk File EXclusiveness (SF-EX)

Synopsis

```
[OCF_RESKEY_device=string] [OCF_RESKEY_index=integer]
[OCF_RESKEY_collision_timeout=integer]
[OCF_RESKEY_monitor_interval=integer]
[OCF_RESKEY_lock_timeout=integer] sfex [start | stop | monitor | meta-data]
```

Description

Resource script for SF-EX. It manages a shared storage medium exclusively .

Supported Parameters

OCF_RESKEY_device=block device

Block device path that stores exclusive control data.

OCF_RESKEY_index=index

Location in block device where exclusive control data is stored. 1 or more is specified. Default is 1.

OCF_RESKEY_collision_timeout=waiting time for lock acquisition

Waiting time when a collision of lock acquisition is detected. Default is 1 second.

OCF_RESKEY_monitor_interval=monitor interval

Monitor interval(sec). Default is 10 seconds

OCF_RESKEY_lock_timeout=Valid term of lock

Valid term of lock(sec). Default is 20 seconds.

ocf:SphinxSearchDaemon (7)

ocf:SphinxSearchDaemon — Manages the Sphinx search daemon.

Synopsis

```
OCF_RESKEY_config=string [OCF_RESKEY_searchd=string]  
[OCF_RESKEY_search=string] [OCF_RESKEY_testQuery=string]  
SphinxSearchDaemon [start | stop | monitor | meta-data | validate-all]
```

Description

This is a searchd Resource Agent. It manages the Sphinx Search Daemon.

Supported Parameters

OCF_RESKEY_config=Configuration file
searchd configuration file

OCF_RESKEY_searchd=searchd binary
searchd binary

OCF_RESKEY_search=search binary
Search binary for functional testing in the monitor action.

OCF_RESKEY_testQuery=test query
Test query for functional testing in the monitor action. The query does not need to match any documents in the index. The purpose is merely to test whether the search daemon is able to query its indices and respond properly.

ocf:Squid (7)

ocf:Squid — Manages a Squid proxy server instance

Synopsis

```
[OCF_RESKEY_squid_exe=string] OCF_RESKEY_squid_conf=string  
OCF_RESKEY_squid_pidfile=string OCF_RESKEY_squid_port=integer  
[OCF_RESKEY_squid_stop_timeout=integer]  
[OCF_RESKEY_debug_mode=string] [OCF_RESKEY_debug_log=string] Squid  
[start | stop | status | monitor | meta-data | validate-all]
```

Description

The resource agent of Squid. This manages a Squid instance as an HA resource.

Supported Parameters

OCF_RESKEY_squid_exe=Executable file

This is a required parameter. This parameter specifies squid's executable file.

OCF_RESKEY_squid_conf=Configuration file

This is a required parameter. This parameter specifies a configuration file for a squid instance managed by this RA.

OCF_RESKEY_squid_pidfile=Pidfile

This is a required parameter. This parameter specifies a process id file for a squid instance managed by this RA.

OCF_RESKEY_squid_port=Port number

This is a required parameter. This parameter specifies a port number for a squid instance managed by this RA. If plural ports are used, you must specify the only one of them.

OCF_RESKEY_squid_stop_timeout=Number of seconds to await to confirm a normal stop method

This is an omittable parameter. On a stop action, a normal stop method is firstly used. and then the confirmation of its completion is awaited for the specified seconds by this parameter. The default value is 10.

OCF_RESKEY_debug_mode=Debug mode

This is an optional parameter. This RA runs in debug mode when this parameter includes 'x' or 'v'. If 'x' is included, both of STDOUT and STDERR redirect to the logfile specified by "debug_log", and then the builtin shell option 'x' is turned on. It is similar about 'v'.

OCF_RESKEY_debug_log=A destination of the debug log

This is an optional and omittable parameter. This parameter specifies a destination file for debug logs and works only if this RA run in debug mode. Refer to "debug_mode" about debug mode. If no value is given but it's required, it's made by the following rules: "/var/log/" as a directory part, the basename of the configuration file given by "syslog_ng_conf" as a basename part, ".log" as a suffix.

ocf:Stateful (7)

ocf:Stateful — Example stateful resource agent

Synopsis

```
OCF_RESKEY_state=string Stateful [start | stop | monitor | meta-data | validate-  
all]
```

Description

This is an example resource agent that impliments two states

Supported Parameters

```
OCF_RESKEY_state=State file  
    Location to store the resource state in
```

ocf:SysInfo (7)

ocf:SysInfo — Records various node attributes in the CIB

Synopsis

```
[OCF_RESKEY_pidfile=string] [OCF_RESKEY_delay=string] SysInfo [start  
| stop | monitor | meta-data | validate-all]
```

Description

This is a SysInfo Resource Agent. It records (in the CIB) various attributes of a node
Sample Linux output: arch: i686 os: Linux-2.4.26-gentoo-r14 free_swap: 1999 cpu_info:
Intel(R) Celeron(R) CPU 2.40GHz cpu_speed: 4771.02 cpu_cores: 1 cpu_load: 0.00
ram_total: 513 ram_free: 117 root_free: 2.4 Sample Darwin output: arch: i386 os:
Darwin-8.6.2 cpu_info: Intel Core Duo cpu_speed: 2.16 cpu_cores: 2 cpu_load: 0.18
ram_total: 2016 ram_free: 787 root_free: 13 Units: free_swap: Mb ram_*: Mb root_free:
Gb cpu_speed (Linux): bogomips cpu_speed (Darwin): Ghz

Supported Parameters

OCF_RESKEY_pidfile=PID file
PID file

OCF_RESKEY_delay=Dampening Delay
Interval to allow values to stabilize

ocf:syslog-ng (7)

ocf:syslog-ng — Syslog-ng resource agent

Synopsis

```
[OCF_RESKEY_configfile=string]
[OCF_RESKEY_syslog_ng_binary=string]
[OCF_RESKEY_start_opts=string]
[OCF_RESKEY_kill_term_timeout=integer] syslog-ng [start | stop | status
| monitor | meta-data | validate-all]
```

Description

This script manages a syslog-ng instance as an HA resource.

Supported Parameters

`OCF_RESKEY_configfile=`Configuration file

This parameter specifies a configuration file for a syslog-ng instance managed by this RA.

`OCF_RESKEY_syslog_ng_binary=`syslog-ng executable

This parameter specifies syslog-ng's executable file.

`OCF_RESKEY_start_opts=`Start options

This parameter specifies startup options for a syslog-ng instance managed by this RA. When no value is given, no startup options is used. Don't use option '-F'. It causes a stuck of a start action.

`OCF_RESKEY_kill_term_timeout=`Number of seconds to await to confirm a normal stop method

On a stop action, a normal stop method(`pkill -TERM`) is firstly used. And then the confirmation of its completion is waited for the specified seconds by this parameter. The default value is 10.

ocf:tomcat (7)

ocf:tomcat — Manages a Tomcat servlet environment instance

Synopsis

```
OCF_RESKEY_tomcat_name=string OCF_RESKEY_script_log=string  
[OCF_RESKEY_tomcat_stop_timeout=integer]  
[OCF_RESKEY_tomcat_suspend_trialcount=integer]  
[OCF_RESKEY_tomcat_user=string] [OCF_RESKEY_statusurl=string]  
[OCF_RESKEY_java_home=string] OCF_RESKEY_catalina_home=string  
OCF_RESKEY_catalina_pid=string  
[OCF_RESKEY_tomcat_start_opts=string]  
[OCF_RESKEY_catalina_opts=string]  
[OCF_RESKEY_catalina_rotate_log=string]  
[OCF_RESKEY_catalina_rotatetime=integer] tomcat [start | stop | status |  
monitor | meta-data | validate-all]
```

Description

Resource script for tomcat. It manages a Tomcat instance as an HA resource.

Supported Parameters

OCF_RESKEY_tomcat_name=The name of the resource
The name of the resource

OCF_RESKEY_script_log=A destination of the log of this script
A destination of the log of this script

OCF_RESKEY_tomcat_stop_timeout=Time-out at the time of the stop
Time-out at the time of the stop

OCF_RESKEY_tomcat_suspend_trialcount=The re-try number of times awaiting a stop

The re-try number of times awaiting a stop

OCF_RESKEY_tomcat_user=A user name to start a resource

A user name to start a resource

OCF_RESKEY_statusurl=URL for state confirmation

URL for state confirmation

OCF_RESKEY_java_home=Home directory of the Java

Home directory of the Java

OCF_RESKEY_catalina_home=Home directory of Tomcat

Home directory of Tomcat

OCF_RESKEY_catalina_pid=A PID file name of Tomcat

A PID file name of Tomcat

OCF_RESKEY_tomcat_start_opts=Tomcat start options

Tomcat start options

OCF_RESKEY_catalina_opts=Catalina options

Catalina options

OCF_RESKEY_catalina_rotate_log=Rotate catalina.out flag

Rotate catalina.out flag

OCF_RESKEY_catalina_rotatetime=Time span of the rotate catalina.out

Time span of the rotate catalina.out

ocf:VIPArip (7)

ocf:VIPArip — Manages a virtual IP address through RIP2

Synopsis

```
OCF_RESKEY_ip=string [OCF_RESKEY_nic=string]  
[OCF_RESKEY_zebra_binary=string] [OCF_RESKEY_ripd_binary=string]  
VIPArip [start | stop | monitor | validate-all | meta-data]
```

Description

Virtual IP Address by RIP2 protocol. This script manages IP alias in different subnet with quagga/ripd. It can add an IP alias, or remove one.

Supported Parameters

OCF_RESKEY_ip=The IP address in different subnet
The IPv4 address in different subnet, for example "192.168.1.1".

OCF_RESKEY_nic=The nic for broadcast the route information
The nic for broadcast the route information. The ripd uses this nic to broadcast the route informaton to others

OCF_RESKEY_zebra_binary=zebra binary
Absolute path to the zebra binary.

OCF_RESKEY_ripd_binary=ripd binary
Absolute path to the ripd binary.

ocf:VirtualDomain (7)

ocf:VirtualDomain — Manages virtual domains through the libvirt virtualization framework

Synopsis

```
OCF_RESKEY_config=string [OCF_RESKEY_hypervisor=string]
[OCF_RESKEY_force_stop=boolean]
[OCF_RESKEY_migration_transport=string]
[OCF_RESKEY_monitor_scripts=string] VirtualDomain [start | stop | status
| monitor | migrate_from | migrate_to | meta-data | validate-all]
```

Description

Resource agent for a virtual domain (a.k.a. domU, virtual machine, virtual environment etc., depending on context) managed by libvirt.

Supported Parameters

OCF_RESKEY_config=Virtual domain configuration file
Absolute path to the libvirt configuration file, for this virtual domain.

OCF_RESKEY_hypervisor=Hypervisor URI
Hypervisor URI to connect to. See the libvirt documentation for details on supported URI formats. The default is system dependent.

OCF_RESKEY_force_stop=Always force shutdown on stop
Always forcefully shut down ("destroy") the domain on stop. The default behavior is to resort to a forceful shutdown only after a graceful shutdown attempt has failed. You should only set this to true if your virtual domain (or your virtualization backend) does not support graceful shutdown.

`OCF_RESKEY_migration_transport=Remote` hypervisor transport

Transport used to connect to the remote hypervisor while migrating. Please refer to the libvirt documentation for details on transports available. If this parameter is omitted, the resource will use libvirt's default transport to connect to the remote hypervisor.

`OCF_RESKEY_monitor_scripts=`space-separated list of monitor scripts

To additionally monitor services within the virtual domain, add this parameter with a list of scripts to monitor. Note: when monitor scripts are used, the start and migrate_from operations will complete only when all monitor scripts have completed successfully. Be sure to set the timeout of these operations to accommodate this delay.

ocf:vmware (7)

ocf:vmware — Manages VMWare Server 2.0 virtual machines

Synopsis

```
[OCF_RESKEY_vmxpath=string] [OCF_RESKEY_vimshbin=string] vmware  
[start | stop | monitor | meta-data]
```

Description

OCF compliant script to control vmware server 2.0 virtual machines.

Supported Parameters

OCF_RESKEY_vmxpath=VMX file path
VMX configuration file path

OCF_RESKEY_vimshbin=vmware-vim-cmd path
vmware-vim-cmd executable path

ocf:WAS6 (7)

ocf:WAS6 — Manages a WebSphere Application Server 6 instance

Synopsis

```
[OCF_RESKEY_profile=string] WAS6 [start | stop | status | monitor | validate-all |  
meta-data | methods]
```

Description

Resource script for WAS6. It manages a Websphere Application Server (WAS6) as an HA resource.

Supported Parameters

OCF_RESKEY_profile=profile name
The WAS profile name.

ocf:WAS (7)

ocf:WAS — Manages a WebSphere Application Server instance

Synopsis

```
[OCF_RESKEY_config=string] [OCF_RESKEY_port=integer] WAS [start | stop |  
status | monitor | validate-all | meta-data | methods]
```

Description

Resource script for WAS. It manages a Websphere Application Server (WAS) as an HA resource.

Supported Parameters

OCF_RESKEY_config=configuration file
The WAS-configuration file.

OCF_RESKEY_port=port
The WAS-(snoop)-port-number.

ocf:WinPopup (7)

ocf:WinPopup — Sends an SMB notification message to selected hosts

Synopsis

[OCF_RESKEY_hostfile=string] WinPopup [start | stop | status | monitor | validate-all | meta-data]

Description

Resource script for WinPopup. It sends WinPopups message to a sysadmin's workstation whenever a takeover occurs.

Supported Parameters

OCF_RESKEY_hostfile=Host file

The file containing the hosts to send WinPopup messages to.

ocf:Xen (7)

ocf:Xen — Manages Xen unprivileged domains (DomUs)

Synopsis

```
[OCF_RESKEY_xmfile=string] [OCF_RESKEY_name=string]  
[OCF_RESKEY_shutdown_timeout=boolean]  
[OCF_RESKEY_allow_mem_management=boolean]  
[OCF_RESKEY_reserved_Dom0_memory=string]  
[OCF_RESKEY_monitor_scripts=string] Xen [start | stop | migrate_from |  
migrate_to | monitor | meta-data | validate-all]
```

Description

Resource Agent for the Xen Hypervisor. Manages Xen virtual machine instances by mapping cluster resource start and stop, to Xen create and shutdown, respectively. A note on names We will try to extract the name from the config file (the xmfile attribute). If you use a simple assignment statement, then you should be fine. Otherwise, if there's some python acrobacy involved such as dynamically assigning names depending on other variables, and we will try to detect this, then please set the name attribute. You should also do that if there is any chance of a pathological situation where a config file might be missing, for example if it resides on a shared storage. If all fails, we finally fall back to the instance id to preserve backward compatibility. Para-virtualized guests can also be migrated by enabling the meta_attribute allow-migrate.

Supported Parameters

OCF_RESKEY_xmfile=Xen control file

Absolute path to the Xen control file, for this virtual machine.

OCF_RESKEY_name=Xen DomU name

Name of the virtual machine.

OCF_RESKEY_shutdown_timeout=Shutdown escalation timeout

The Xen agent will first try an orderly shutdown using `xm shutdown`. Should this not succeed within this timeout, the agent will escalate to `xm destroy`, forcibly killing the node. If this is not set, it will default to two-third of the stop action timeout. Setting this value to 0 forces an immediate destroy.

OCF_RESKEY_allow_mem_management=Use dynamic memory management

This parameter enables dynamic adjustment of memory for start and stop actions used for Dom0 and the DomUs. The default is to not adjust memory dynamically.

OCF_RESKEY_reserved_Dom0_memory=Minimum Dom0 memory

In case memory management is used, this parameter defines the minimum amount of memory to be reserved for the dom0. The default minimum memory is 512MB.

OCF_RESKEY_monitor_scripts=list of space separated monitor scripts

To additionally monitor services within the unprivileged domain, add this parameter with a list of scripts to monitor. NB: In this case make sure to set the start-delay of the monitor operation to at least the time it takes for the DomU to start all services.

ocf:Xinetd (7)

ocf:Xinetd — Manages an Xinetd service

Synopsis

```
[OCF_RESKEY_service=string] Xinetd [start | stop | restart | status | monitor |  
validate-all | meta-data]
```

Description

Resource script for Xinetd. It starts/stops services managed by xinetd. Note that the xinetd daemon itself must be running: we are not going to start it or stop it ourselves. Important: in case the services managed by the cluster are the only ones enabled, you should specify the -stayalive option for xinetd or it will exit on Heartbeat stop. Alternatively, you may enable some internal service such as echo.

Supported Parameters

OCF_RESKEY_service=service name
The service name managed by xinetd.

パート V. 付録

単純なテストリソースのセットアップ例

A

この章では、単純なリソース(IPアドレス)を設定する基本的な例を示します。Pacemaker GUIまたはcrmコマンドラインツールをのいずれかを使用した、両方の方法を紹介します。

次の例では、第3章 *YaST*によるインストールと基本設定(21 ページ)で説明されているようにクラスタがセットアップされ、クラスタが2つ以上のノードで構成されていると想定します。Pacemaker GUIとcrmシェルでクラスタリソースを設定する方法の紹介と概要については、次の各章を参照してください。

- ・ クラスタリソースの設定と管理(GUI) (61 ページ)
- ・ クラスタリソースの設定と管理(コマンドライン) (95 ページ)

A.1 GUIによるリソースの構成

サンプルのクラスタリソースを作成して別のサーバにマイグレートすると、クラスタが正常に機能していることの確認に役立ちます。構成とマイグレート of シンプルなリソースは、IPアドレスです。

手順 A.1 IPアドレスクラスタリソースを作成する

- 1 5.1.1項「クラスタへの接続」(62 ページ)で説明したように、Pacemaker GUIを起動してクラスタにログインします。
- 2 左側のペインで [リソース] ビューに切り替え、右側のペインで、変更するグループを選択して [編集] をクリックします。次のウィンドウに

は、そのリソースに定義された基本的なグループパラメータとメタ属性とプリミティブが表示されます。

- 3 [プリミティブ] タブをクリックして、[追加] をクリックします。
- 4 次のダイアログで、次のパラメータを設定してIPアドレスをグループのサブリソースとして追加します。
 - 4a 一意のID(たとえば、myIP)を入力します。
 - 4b [クラス] リストで、リソースエージェントクラスとして [ocf] を選択します。
 - 4c OCFリソースエージェントの [プロバイダ] として、[heartbeat] を選択します。
 - 4d [タイプ] リストで、リソースエージェントとして [IPaddr] を選択します。
 - 4e [進む] をクリックします。
 - 4f [Instance Attribute(インスタンス属性)] タブで、[IP] エントリを選択して [編集] をクリックします(または [IP] エントリをダブルクリックします)。
 - 4g [値] として、目的のIPアドレスを入力します。たとえば、「10.10.0.1」と入力して、[OK] をクリックします。
 - 4h 新規インスタンス属性を [追加] して、nicを [名前] に、eth0を [値] に指定して [OK] をクリックします。

名前と値は、ハードウェア構成、およびHigh Availability Extensionソフトウェアのインストール中に選択したメディア構成とは独立しています。
- 5 すべてのパラメータを目的どおりに設定したら、[OK] をクリックして、そのリソースの設定を完了します。構成ダイアログが閉じて、メインウィンドウに変更されたリソースが表示されます。

リソースをPacemakerGUIで起動するには、左側のペインの〔管理〕を選択します。右側のペインで、リソースを右クリックして〔開始〕を選択します(またはツールバーから開始します)。

IPアドレスリソースを別のノード(satum)にマイグレートするには、次のようにします。

手順 A.2 リソースを他のノードへマイグレートする

- 1 左側のペインの〔管理〕ビューに切り替え、次に右側のペインのIPアドレスリソースを右クリックして〔*Migrate Resource*(リソースのマイグレート)〕を選択します。
- 2 新規ウィンドウで、〔*To Node*(マイグレート先ノード)〕ドロップダウンリストでsatumを選択し、選択したリソースをノードsatumに移動します。
- 3 リソースを一時的にマイグレートするには、〔*Duration*(期間)〕をアクティブにしてリソースが新規ノードにマイグレートされる時間を入力します。
- 4 〔OK〕をクリックして、マイグレーションを確認します。

A.2 リソースの手動設定

リソースは、コンピュータが提供するあらゆる種類のサービスです。リソースはRA(リソースエージェント)によって管理されている場合はHigh Availabilityに認識され、これにはLSBスクリプト、OCFスクリプト、従来のHeartbeat 1リソースがあります。すべてのリソースはcrmコマンドで、またはXMLとしてCIB(Cluster Information Base)のresourcesセクションで構成されます。使用できるリソースの概要は、第19章 *HA OCF Agents* (275 ページ)を参照してください。

IPアドレス10.10.0.1をリソースとして現在の構成に追加するには、crmコマンドを使用します。

手順 A.3 IPアドレスクラスタリソースを作成する

- 1 シェルを開いてrootになります。

2 「crm configure」と入力して、内部シェルを開きます。

3 IPアドレスリソースを作成します。

```
crm(live)configure# resource
primitive myIP ocf:heartbeat:IPaddr params ip=10.10.0.1
```

注記

リソースを**High Availability**で構成する場合、同じリソースをinitで初期化できません。高可用性はすべてのサービスの**start**または**stop**アクションを実施します。

構成が正常に終了した場合、新規リソースはクラスタのランダムノードで開始されたcrm_monに表示されます。

リソースを別のノードにマイグレートするには、次のようにします。

手順 A.4 リソースを他のノードへマイグレートする

1 シェルを起動してrootになります。

2 リソースmyipをノードsaturnにマイグレートします。

```
crm resource migrate myIP saturn
```


クラスタの最新製品バージョン へのアップグレード

B

SUSE® Linux Enterprise Server 10をベースとする既存クラスタがある場合は、そのクラスタを更新して、SUSE Linux Enterprise Server 11または11 SP1上のHigh Availability Extensionで実行することができます。

SUSE Linux Enterprise Server 10からSUSE Linux Enterprise Server 11または11 SP1へ移行する場合は、すべてのクラスタノードをオフラインにして、クラスタを全体として移行する必要があります。SUSE Linux Enterprise Server 10やSUSE Linux Enterprise Server 11上で実行している混合クラスタはサポートされていません。

B.1 SLES 10からSLEHA 11へのアップグレード

便宜のため、SUSE® Linux Enterprise High Availability Extensionには、`hb2openais.sh`スクリプトが含まれており、このスクリプトを使用すると、HeartbeatからOpenAISクラススタスタックへの移動時にデータを変換できます。スクリプトは、`/etc/ha.d/ha.cf`に保存されている環境設定を解析し、OpenAISクラススタスタック用の新しい環境設定ファイルを生成します。さらに、CIBを調整してOpenAIS表記規則と一致させ、OCFS2ファイルシステムを変換し、EVMSをcLVMで置き換えます。EVMS2コンテナは、すべて、cLVM2ボリュームに変換されます。CIB内の既存リソースで参照されるボリュームグループの場合は、新しいLVMリソースが作成されます。

クラスタをSUSE Linux Enterprise Server 10 SP3からSUSE Linux Enterprise Server 11に正常に移行するには、次の手順を実行する必要があります。

1. SUSE Linux Enterprise Server 10 SP3クラスタを準備する (374 ページ)
2. SUSE Linux Enterprise 11に更新する (375 ページ)
3. 変換をテストする (376 ページ)
4. データの変換 (377 ページ)

変換が正常に完了したら、更新したクラスタを再度オンラインにすることができます。

注記: 更新を元に戻すには

SUSE Linux Enterprise Server 11への更新後に、SUSE Linux Enterprise Server 10に戻す処理は、サポートされていません。

B.1.1 準備とバックアップ

クラスタを次の製品バージョンへ更新し、適宜、データを変換するには、その前に、現在のクラスタを準備する必要があります。

手順 B.1 *SUSE Linux Enterprise Server 10 SP3*クラスタを準備する

- 1 クラスタにログインします。
- 2 Heartbeat環境設定ファイル/etc/ha.d/ha.cfをレビューし、すべての通信メディアがマルチキャストをサポートしているかどうかチェックします。
- 3 次のファイルがすべてのノードで等しいことを確認します。/etc/ha.d/ha.cfおよび/var/lib/heartbeat/crm/cib.xml
- 4 各ノードで`rcheartbeat stop`を実行することで、すべてのノードをオフラインにします。
- 5 最新バージョンへの更新前の一般的なシステムバックアップ(推奨)に加えて、次のファイルをバックアップします。これらのファイルは、SUSE

Linux Enterprise Server 11への更新後の変換スクリプトの実行で必要になります。

- /var/lib/heartbeat/crm/cib.xml
- /var/lib/heartbeat/hostcache
- /etc/ha.d/ha.cf
- /etc/logd.cf

- 6** EVMS2リソースがある場合は、非LVM EVMS2ボリュームをSUSE Linux Enterprise Server 10上の互換ボリュームに変換します。これらは、変換処理中(B.1.3項「データの変換」(376 ページ)参照)に、LVM2ボリュームグループになります。変換後は、`vgchange -c y`を使用して、各ボリュームグループをHigh Availabilityクラスタのメンバとして必ずマークしてください。

B.1.2 更新/インストール

クラスタを準備し、ファイルをバックアップしたら、クライアントノードを次の製品バージョンへ更新できます。更新を実行する代わりに、クラスタノードにSUSE Linux Enterprise 11を新規インストールすることもできます。

手順 B.2 SUSE Linux Enterprise 11に更新する

- 1** すべてのクラスタノードで、SUSE Linux Enterprise Server 10 SP3からSUSE Linux Enterprise Server 11への更新を実行します。ご使用製品の更新方法については、『SUSE Linux Enterprise Server 11 導入ガイド』の「*SUSE Linux Enterprise* のアップデート」の章を参照してください。

または、すべてのクラスタノードで、SUSE Linux Enterprise Server 11の新規インストールを実行することもできます。

- 2** すべてクラスタノードで、SUSE Linux Enterprise ServerのアドオンとしてSUSE Linux Enterprise High Availability Extension 11をインストールします。詳細については、3.1項「High Availability Extensionのインストール」(21 ページ)を参照してください。

B.1.3 データの変換

SUSE Linux Enterprise Server 11とHigh Availability Extensionをインストールしたら、データ変換を開始できます。High Availability Extensionとともに出荷される変換スクリプトは、注意深く設定されていますが、完全な自動モードですべての設定を行うことはできません。このスクリプトでは、実効する変更について管理者に警告し、対話と管理者側での決定を必要とします。管理者は、クラスタの詳細を知っている必要があり、変更の妥当性を確認する責任があります。変換スクリプトは、`/usr/lib/heartbeat`(64ビットマシンの場合は、`/usr/lib64/heartbeat`)に格納されています。

注記: テストランの実行

変換プロセスをよく知るために、まず、変換をテストすること(変更なしで)を強くお勧めします。同じテストディレクトリを使用すると、ファイルを1回コピーするだけで、テストランを繰り返すことができます。

手順 B.3 変換をテストする

- 1 ノードの1つで、テストディレクトリを作成し、そのテストディレクトリにバックアップファイルをコピーします。

```
$ mkdir /tmp/hb2openais-testdir
$ cp /etc/ha.d/ha.cf /tmp/hb2openais-testdir
$ cp /var/lib/heartbeat/hostcache /tmp/hb2openais-testdir
$ cp /etc/logd.cf /tmp/hb2openais-testdir
$ sudo cp /var/lib/heartbeat/crm/cib.xml /tmp/hb2openais-testdir
```

- 2 次のコマンドで、テストランを開始します。

```
$ /usr/lib/heartbeat/hb2openais.sh -T /tmp/hb2openais-testdir -U
```

64ビットシステムを使用する場合は、次のコマンドを使用します。

```
$ /usr/lib64/heartbeat/hb2openais.sh -T /tmp/hb2openais-testdir -U
```

- 3 結果として生成された`openais.conf`ファイルと`cib-out.xml`ファイルを読んで検証します。

```
$ cd /tmp/hb2openais-testdir
$ less openais.conf
$ crm_verify -V -x cib-out.xml
```

変換段階の詳細については、インストールした**High Availability Extension**の `/usr/share/doc/packages/pacemaker/README.hb2openais` を参照してください。

手順 B.4 データの変換

テストランを実行し、出力をチェックしたら、データ変換を開始できます。変換は、1つのノードで実行するだけで済みます。メインクラスタ構成(CIB)が自動的にその他のノードにレプリケートされます。レプリケートの必要がある他のすべてのファイルは、変換スクリプトによって自動的にコピーされます。

- 1 変換スクリプトで他のクラスタノードにファイルを正常にコピーするため、`root`に許可されたアクセスで`sshd`がすべてのノードで実行されていることを確認します。
- 2 すべてのOCF2ファイルシステムがマウント解除されていることを確認します。
- 3 **High Availability Extension**は、デフォルトの**OpenAIS**環境設定ファイルとともに出荷されています。以降の手順で、デフォルトの環境設定を上書きしたくない場合は、`/etc/ais/openais.conf`環境設定ファイルのコピーを作成します。
- 4 変換スクリプトを`root`として起動します。`sudo`を使用する場合は、`-u`オプションで特権ユーザを指定します。

```
$ /usr/lib/heartbeat/hb2openais.sh -u root
```

`/etc/ha.d/ha.cf`に保存されている環境設定に基づいて、スクリプトは、**OpenAIS**クラスタスタック用の新しい環境設定ファイル`/etc/ais/openais.conf`を生成します。スクリプトは、**CIB**の設定を分析し、**Heartbeat**から**OpenAIS**への変更に伴いクラスタ設定の変更が必要かどうか通知してきます。すべてのファイル処理は、変換が実行されるノードで行われ、他のノードにレプリケートされます。

- 5 画面の指示に従います。

変換が正常に完了したら、新しいクラスタスタックを「3.3項「クラスタをオンラインにする」(30 ページ)」の説明に従って起動します。

アップグレードプロセスの後で、SUSE Linux Enterprise Server 10に戻すことはできません。

B.1.4 詳細情報

変換スクリプトおよび変換の各段階の詳細については、インストールしたHigh Availability Extensionの/usr/share/doc/packages/pacemaker/README.hb2openaisを参照してください。

B.2 SLEHA 11からSLEHA 11 SP1へのアップグレード

既存クラスタをSUSE Linux Enterprise High Availability Extension 11から11 SP1へ正常に移行するには、ノードを次々にアップグレードする「ローリングアップグレード」を実行できます。SUSE Linux Enterprise High Availability Extension 11 SP1で、主要なクラスタ設定ファイルが/etc/ais/openais.confから/etc/corosync/corosync.confへ変更されたので、スクリプトが必要な変換を実行します。それらは、openaisパッケージの更新時に自動的に実行されます。

手順 B.5 ローリングアップグレードを実行する

重要項目: ソフトウェアパッケージの更新

実行中のクラスタに属するノード上でソフトウェアパッケージを更新する場合は、そのノードでクラスタスタックを停止してから、ソフトウェアの更新を開始します。クラスタスタックを停止するには、rootとしてノードにログインし、「rcopenais stop」を入力します。

ソフトウェアの更新中にOpenAIS/Corosyncが実行されている場合は、アクティブノードのフェンシングなど、予期しない結果が生じる可能性があります。

-
- 1 アップグレードするノードでrootとしてログインし、OpenAISを停止します。

```
rcopenais stop
```

- 2 システムバックアップが最新で、復元可能かどうか確認します。
- 3 SUSE Linux Enterprise Server 11からSUSE Linux Enterprise Server 11 SP1へのアップグレードとSUSE Linux Enterprise High Availability Extension 11からSUSE Linux Enterprise High Availability Extension 11 SP1へのアップグレードを実行します。ご使用製品の更新方法については、『SUSE Linux Enterprise Server 11 SP1導入ガイド』の「*SUSE Linux Enterprise*のアップデート」の章を参照してください。
- 4 アップグレードしたノードでOpenAISまたはCorosyncを再起動して、ノードをクラスタに再加入させます。

```
rcopenais start
```

- 5 次のノードをオフラインにし、そのノードに関して手順を繰り返します。

新機能

以降のセクションでは、バージョンからバージョンへの変更内容の概要を示します。この概要では、たとえば、基本設定の完全な再設定、他の場所への設定ファイルの移動などの著しい変更が行われたかどうかを示しています。

C.1 バージョン10 SP3からバージョン11への変更点

SUSE Linux Enterprise Server 11では、クラスタスタックがHeartbeatからOpenAISに変更しました。OpenAISは業界標準のAPIとしてService Availability Forumが発行しているApplication Interface Specification (AIS)を実装しています。SUSE Linux Enterprise Server 10のクラスタリソースマネージャも残っていますが、大幅に機能が強化され、OpenAISに移植され、現在はPacemakerと呼ばれています。

SUSE® Linux Enterprise Server 10 SP3からSUSE Linux Enterprise Server 11へのHigh Availabilityコンポーネントの変更の詳細については、以降のセクションを参照してください。

C.1.1 新しく追加された機能

マイグレーションのしきい値と失敗タイムアウト

High Availability Extensionに、移行しきい値と失敗タイムアウトのコンセプトが含まれるようになりました。新しいノードへのマイグレートを行う

基準となるリソースの失敗回数を定義できます。デフォルトでは、管理者がリソースの失敗回数を手動でリセットするまで、ノードは失敗したリソースを実行できなくなります。ただし、リソースの`failure-timeout`オプションを設定することで、リソースの失敗回数を失効させることができます。

リソースと操作のデフォルト

リソースオプションと操作にグローバルなデフォルトを設定できるようになりました。

オフラインの設定変更のサポート

設定をアトミックに更新する前に、一連の変更の影響をプレビューすることが望ましい場合が多くあります。構成の「シャドー」コピーを作成して、実行前にコマンドラインインタフェースで編集し、アクティブなクラスタ構成を個別に変更できるようになりました。

ルール、オプション、操作セットの再利用

ルール、`instance_attributes`、`meta_attributes`、および操作セットは、1度定義しておけば、複数の箇所で参照できます。

CIB内の特定操作に対するXPath式の使用

CIBでXPathベースの`create`、`modify`、`delete`操作が使用できるようになりました。詳細は、`cibadmin`のヘルプテキストを参照してください。

多次元のコロケーションと順序の制約

一連のコロケーションリソースを作成する場合、これまではリソースグループを定義するか(設計を必ずしも正確に表現していない場合があった)、個別の制約として各関係を定義するかのいずれかが可能でした。その結果、リソースや組み合わせの数が増加するにつれて、制約も膨大なものになることもありました。今回、`resource_sets`の定義によって別な形式でコロケーションの制約を指定できるようになりました。

クラスタ化されていないマシンからのCIBへの接続

`Pacemaker`がマシンにインストールされていれば、マシン自体がクラスタに属していない場合でもクラスタに接続できます。

反復アクションを既知の回数トリガ

デフォルトでは、リソースの開始時刻に対して相対的に反復アクションがスケジュールされますが、これが適切ではない場合があります。操作を相対的に実施する日付/時刻を指定するため、操作の間隔開始時刻を設定

します。クラスタはこの時刻を使用して、開始時刻+(間隔*N)で操作を開始するように、適切な開始遅延を計算します。

C.1.2 変更された機能

リソースとクラスタオプションに関する命名規則

すべてのリソースとクラスタオプションには、アンダースコア(_)の代わりにダッシュ(-)を使用するようになりました。たとえばmaster_maxメタオプションは、master-maxという名前に変更されました。

master_slaveリソースの名前変更

master_slaveリソースは、masterという名前に変更されました。マスターリソースは、2つのモードのいずれかで実行可能な特殊なクローンタイプです。

属性のコンテナタグ

attributesコンテナタグは削除されました。

前提条件の操作フィールド

pre-req操作フィールドは、requiresという名前に変更されました。

操作間隔

すべての操作に間隔を指定する必要があります。開始および停止操作の場合、間隔は0(ゼロ)に設定する必要があります。

コロケーション属性と順序の制約

コロケーション属性および順序の制約の名前をわかりやすく変更しました。

障害によるマイグレーションのためのクラスタオプション

resource-failure-stickinessクラスタオプションは、migration-thresholdクラスタオプションに替わりました。マイグレーションのしきい値と失敗タイムアウト (381 ページ)も参照してください。

コマンドラインツールの引数

コマンドラインツールの引数が一定になりました。リソースとクラスタオプションに関する命名規則 (383 ページ)も参照してください。

XMLの検証と解析

クラスタ構成はXMLで作成されます。従来のDTD(文書型定義)の代わりに、より強力なRELAX NGスキーマを使用して、構造とコンテンツのパターンを定義するようになりました。libxml2はパーサとして使用します。

idフィールド

idフィールドは、次の制限を持つXML IDになりました。

- IDにコロンを含めることはできません。
- IDは数字から開始できません。
- IDはグローバルに固有なものでなければなりません(そのタグで固有なだけでなく)。

他のオブジェクトの参照

一部のフィールド(リソースへの参照が制約されるフィールドなど)はIDREFです。これは、設定を有効にするためには、それらのフィールドが既存のリソースまたはオブジェクトを参照する必要があることを意味します。そのため、別な場所で参照されているオブジェクトの削除は失敗します。

C.1.3 削除された機能

リソースメタオプションの設定

リソースメタオプションを最上位の属性として設定できなくなりました。代わりにメタ属性を使用してください。crm_resource(8)(248 ページ)も参照してください。

グローバルデフォルトの設定

リソースと操作デフォルトはcrm_configから読み込まれなくなりました。

C.2 バージョン11からバージョン11 SP1への変更点

クラスタ設定ファイル

主要なクラスタ設定ファイルが/etc/ais/openais.confから/etc/corosync/corosync.confへ変更されました。両方のファイルは非常によく似ています。SUSE Linux Enterprise High Availability Extension 11からSP1にアップグレードする際、それらのファイルのわずかな相違点は、スクリプトによって処理されます。OpenAISとCorosyncの関係の詳細については、[\[http://www.corosync.org/doku.php?id=faq:why\]](http://www.corosync.org/doku.php?id=faq:why)を参照してください。

ローリングアップグレード

最小のダウンタイムで既存のクラスタを移行するため、SUSE Linux Enterprise High Availability Extensionでは、SUSE Linux Enterprise High Availability Extension 11から11 SP1への「ローリングアップグレード」を実行できます。クラスタをオンラインにしたまま、次々とノードをアップグレードします。

自動クラスタ展開

クラスタの展開を容易にするため、AutoYaSTでは、既存ノードをクローンできます。AutoYaSTは、インストールデータと設定データを含むAutoYaSTプロファイルを使用して、ユーザの介入なしで、自動的に、1つ以上のSUSE Linux Enterpriseシステムをインストールするためのシステムです。プロファイルによって、インストールする対象と、インストールしたシステムが最終的に完全に使用準備が整ったシステムになるように設定する方法がAutoYaSTに指示されます。このプロファイルは、さまざまな方法で、大量展開に使用できます。

設定ファイルの転送

SUSE Linux Enterprise High Availability ExtensionにはCsync2が標準装備されています。Csync2は、クラスタ内のすべてのノードに設定ファイルを複製するツールです。このツールは、多数のホストを処理できます。また、一定のサブグループのホストだけでファイルを同期することも可能です。Csync2で同期する必要のあるファイルとホスト名の設定には、YaSTを使用します。

クラスタ管理用Webインターフェイス

High Availability Extensionには、管理タスク用のWebベースユーザインターフェイスとしてHA Web Konsoleも組み込まれました。このインターフェイスを使用すると、Linux以外のコンピュータからも、Linuxクラスタを監視および管理できます。このインターフェイスは、システムにグラフィックユーザインターフェイスがなかったり、使用できない場合も理想的なソリューションです。

リソース設定用テンプレート

コマンドラインインターフェイスを使用してリソースを作成および設定する際に、さまざまなリソーステンプレートから選択して、素早く容易に設定できるようになりました。

負荷ベースのリソース配置

特定のノードが提供する容量と特定のリソースが要求する容量を定義し、クラスタで配置ストラテジの1つを選択することによって、リソースをそれらの負荷インパクトに従って配置して、クラスタパフォーマンスの低下を防ぐことができます。

クラスタ対応型アクティブ/アクティブRAID 1

cmirrorrdの使用によって、2つの独立したSANから回復力の早いストレージ設定を作成できるようになりました。

読み取り専用GFS2のサポート

GFS2からOCFS2への移行を容易にするため、GFS2ファイルシステムを読み取り専用モードでマウントして、OCFS2ファイルシステムにデータをコピーできます。OCFS2は、SUSE Linux Enterprise High Availability Extensionによってフルサポートされています。

OCFS2用SCTPサポート

冗長リングが設定されている場合、OCFS2とDLMは、自動的に、ネットワークデバイスボンディングから独立したSCTPによる冗長通信パスを使用します。

ストレージ保護

データ破損からストレージを保護するためにセキュリティレイヤを追加するには、IOフェンシング(external/sbdフェンシングデバイスによる)とsfexリソースエージェントの組み合わせを使用して、排他的なストレージアクセスを確保できます。

Sambaクラスタリング

High Availability Extensionが、トリビアルデータベースのクラスタ実装であるCTDBをサポートするようになりました。これによって、クラスタ化したSambaサーバの設定が可能となり、異種混合環境にもHigh Availabilityソリューションが提供されます。

IP負荷分散用YaSTモジュール

新しいモジュールでは、グラフィックユーザインターフェイスによるカーネルベースの負荷分散の設定が可能です。これは、Linux Virtual Serverを管理し、実サーバを監視するユーザスペースデーモンldirectordのフロントエンドです。

GNU利用許諾契約書

D

この付録には、GNU一般公衆利用許諾契約書(GPL)とGNUフリー文書利用許諾契約書(GFDL)が含まれています。

GNU General Public License

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 「Program」, below, refers to any such program or work, and a 「work based on the Program」 means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 「modification」.) Each licensee is addressed as 「you」.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and 「any later version」, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the 「copyright」 line and a pointer to where the full notice is found.

one line to give the program's name and an idea of what it does. Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) year name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'. This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a 「copyright disclaimer」 for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

```
signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License [<http://www.fsf.org/licenses/lgpl.html>] instead of this License.

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of 「copyleft」, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The 「Document」, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as 「you」. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A 「Modified Version」 of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A 「Secondary Section」 is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The 「Invariant Sections」 are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The 「Cover Texts」 are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A 「Transparent」 copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not 「Transparent」 is called 「Opaque」.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The 「Title Page」 means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, 「Title Page」 means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section 「Entitled XYZ」 means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as 「Acknowledgements」, 「Dedications」, 「Endorsements」, or 「History」.) To 「Preserve the Title」 of such a section when you modify the Document means that it remains a section 「Entitled XYZ」 according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled 「History」, Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled 「History」 in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the 「History」 section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled 「Acknowledgements」 or 「Dedications」, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled 「Endorsements」. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled 「Endorsements」 or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled 「Endorsements」, provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled 「History」 in the various original documents, forming one section Entitled 「History」 ; likewise combine any sections Entitled 「Acknowledgements」 , and any sections Entitled 「Dedications」 . You must delete all sections Entitled 「Endorsements」 .

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled 「Acknowledgements」 , 「Dedications」 , or 「History」 , the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License 「or any later version」 applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
only as published by the Free Software Foundation;
with the Invariant Section being this copyright notice and license.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

用語集

アクティブ/アクティブ、アクティブ/パッシブ

サービスがノード上で実行される方法についてのコンセプト。アクティブ/パッシブシナリオでは、1つ以上のサービスがアクティブノード上で実行され、パッシブノードはアクティブノードの失敗を待機します。アクティブ/アクティブでは、各ノードがアクティブであると同時にパッシブです。

cluster

ハイパフォーマンスクラスタは、結果を早く出すためにアプリケーション負荷を共有するコンピュータ(実際または仮想のコンピュータ)のグループです。高可用性クラスタは、サービスの可用性を最大にすることを第一に設計されています。

クラスタ情報ベース(CIB)

クラスタ構成全体と状態の表現(ノードメンバーシップ、リソース、制約など)。XMLで書かれ、メモリに常駐しています。マスタCIBは、指定コーディネータ(DC) (398 ページ)で保持および保守され、他のノードに複製されます。

クラスタパーティション

1つ以上のノードとその他のクラスタ間で通信が失敗した場合は、常にクラスタパーティションが発生します。クラスタパーティションのノードはまだアクティブで、相互に通信できますが、どのノードと通信できないかを認識していません。その他のパーティションの損失を確認できないため、スプリットブレインシナリオが作成されました(スプリットブレイン (400 ページ)も参照)。

クラスタリソースマネージャ(CRM)

すべての非ローカルインタラクションの調整に責任を負う主要管理エンティティ。クラスタの各ノードにはノード独自のCRMがありますが、DC上で実行されるCRMは、決定を他の非ローカルCRMに中継し、それらからの入力进行处理するために選択されたCRMです。CRMは、多数のコンポーネント(CRM自身のノードとその他のノード両方のローカルリソースマネージャ、非ローカルCRM、管理コマンド、フェンシング機能、メンバーシップ層)と対話します。

コンセンサスクラスタメンバーシップ(CCM)

CCMは、どのノードがクラスタを構成するか決定し、この情報をクラスタで共有します。ノードまたはクォラムの新規追加および損失は、CCMによって通知されます。CCMモジュールはクラスタの各ノード上で実行されます。

指定コーディネータ(DC)

「マスタ」ノード。このノードには、CIBのマスタコピーが保持されます。その他すべてのノードは、現在のDCから構成とリソース割り当て情報を取得します。DCは、メンバーシップの変更後、クラスタ内のすべてのノードから選抜されます。

DLM(Distributed Lock Manager)

DLMは、クラスタファイルシステムのディスクアクセスを調整し、ファイルロッキングを管理して、パフォーマンスと可用性を向上します。

DRBD(Distributed Replicated Block Device)

DRBDは、高可用性クラスタを構築するためのブロックデバイスです。ブロックデバイス全体が専用ネットワーク経由でミラーリングされ、ネットワークRAID-1として認識されます。

failover (フェールオーバー)

リソースまたはノードが1台のマシンで失敗し、影響を受けるリソースが別のノードで起動されたときに発生します。

フェンシング

非クラスタメンバーによる共有リソースへのアクセスを防止するコンセプトを示します。「誤動作」しているノードを終了(シャットダウン)して問題の発生を防止する、状態が不明なノードからリソースをロックする、またはその他の方法で実現されます。フェンシングは、さらにノードフェンシングとリソースフェンシングに区別されます。

Heartbeat リソースエージェント

Heartbeat リソースエージェントはHeartbeatバージョン1で広く使用されてきました。あまり使用されなくなりましたが、バージョン2でサポートされています。Heartbeat リソースエージェントはstart、stop、status操作を実行でき、/etc/ha.d/resource.dまたは/etc/init.dの下にあります。Heartbeat リソースエージェントの詳細については、<http://www.linux-ha.org/HeartbeatResourceAgent>を参照してください(OCF リソースエージェント (399 ページ)も参照)。

ローカルリソースマネージャ(LRM)

ローカルリソースマネージャ(LRM)は、リソース上の操作の実行を担当します。リソースエージェントスクリプトを使用して処理を実行します。LRMはそれ自身ではポリシーを認識していないという点で、「ダム」です。何をすべきか認識させるにはDCが必要です。

LSBリソースエージェント

LSBリソースエージェントは標準LSB初期化スクリプトです。LSB初期化スクリプトは高可用性コンテキストでの使用に限定されません。あらゆるLSB準拠LinuxシステムはLSB初期化スクリプトを使用して、サービスを制御します。あらゆるLSBリソースエージェントはstart、stop、restart、status、force-reloadオプションをサポートし、オプションでtry-restartおよびreloadも使用できます。LSBリソースエージェントは/etc/init.dにあります。LSBリソースエージェントの詳細と実際の仕様は、<http://www.linux-ha.org/LSBResourceAgent>およびhttp://www.linux-foundation.org/spec/refspecs/LSB_3.0.0/LSB-Core-generic/LSB-Core-generic/iniscriptact.htmlを参照してください。(OCFリソースエージェント (399 ページ)とHeartbeatリソースエージェント (398 ページ)も参照してください)。

node (ノード)

クラスタのメンバで、ユーザには見えない(実際または仮想の)コンピュータ。

pingd

pingデーモン。ICMP pingを使用して、クラスタ外部の1台以上のサーバに常に接続します。

ポリシーエンジン(PE)

ポリシーエンジンはCIBでのポリシー変更を実装するために必要な処理を計算します。この情報はトランザクションエンジンに渡され、次にポリシー変更がクラスタセットアップで実装されます。PEは常にDC上で実行されます。

OCFリソースエージェント

OCFリソースエージェントはLSBリソースエージェント(初期化スクリプト)と同様です。任意のOCFリソースエージェントはstart、stop、status (monitorと呼ばれることもある)オプションをサポートする必要があります。また、metadataオプションをサポートし、リソースエー

ジェントタイプの説明をXMLで返します。追加オプションをサポートできますが、必須ではありません。OCFリソースエージェントは/usr/lib/ocf/resource.d/providerにあります。OCFリソースエージェントの詳細と仕様のドラフトについては、<http://www.linux-ha.org/OCFResourceAgent>および<http://www.opencf.org/cgi-bin/viewcvs.cgi/specs/ra/resource-agent-api.txt?rev=HEAD>を参照してください(「Heartbeatリソースエージェント (398 ページ)」も参照)。

クォーラム

クラスタでは、クラスタパーティションは、ノード(投票)の大多数を保有する場合、クォーラムを持つ(「定数に達している」と定義されます。クォーラムはただ1つのパーティションで識別されます。複数の切断されたパーティションまたはノードが処理を続行してデータおよびサービスが破損されないようにする、アルゴリズムの一部です(スプリットブレイン)。クォーラムはフェンシングの前提条件で、このためクォーラムは一意になります。

resource

Heartbeatに認識されている、任意のタイプのサービスまたはアプリケーション。IPアドレス、ファイルシステム、データベースなどです。

リソースエージェント(RA)

リソースエージェント(RA)は、プロキシとして動作してリソースを管理するスクリプトです。リソースエージェントには、OCF(Open Cluster Framework)リソースエージェント、LSBリソースエージェント(標準LSB初期化スクリプト)、Heartbeatリソースエージェント(Heartbeat v1 リソース)の3種類があります。

シングルポイント障害(SPOF)

シングルポイント障害(SPOF)は、失敗した場合、クラスタ全体の失敗につながるクラスタのコンポーネントです。

スプリットブレイン

クラスタノードが(ソフトウェアまたはハードウェア障害によって)互いに認識しない2つ以上のグループに分割される場合のシナリオです。STONITHによって、スプリットブレインがクラスタ全体に悪影響をおよぼさなくなります。「パーティションされたクラスタ」シナリオとも呼ばれます。

スプリットブレインという用語は、DRBDでも使用されますが、2つのノードに異なるデータが含まれることを意味します。

STONITH

「Shoot the other node in the head」の略で、基本的に誤動作しているノードを停止させ、クラスタでの問題発生を防止するものです。

遷移エンジン(TE)

遷移エンジン(TE)はPEからポリシーでいれくていぶを取得し、これを実行します。TEは常にDC上で実行されます。そこから、その他のノードのローカルリソースマネージャに、どのアクションを実行するか指示します。

