



Red Hat Enterprise Linux 7

High Availability アドオンの概要

High Availability Add-On のコンポーネントの概要

Red Hat Enterprise Linux 7 High Availability アドオンの概要

High Availability Add-On のコンポーネントの概要

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2023 | You need to change the HOLDER entity in the en-US/High_Availability_Add-On_Overview.ent file |.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

Red Hat High Availability アドオンの概要では、Red Hat Enterprise Linux 7 向け High Availability アドオンについて簡単に説明します。Red Hat High Availability クラスターのデプロイに関する専門知識をさらに得るために、Red Hat High Availability Clustering (RH436) トレーニングコースが用意されています。

目次

第1章 HIGH AVAILABILITY アドオンの概要	3
1.1. クラスターの基礎	3
1.2. HIGH AVAILABILITY アドオンの紹介	4
1.3. PACEMAKER の概要	4
1.4. PACEMAKER アーキテクチャーコンポーネント	5
1.5. PACEMAKER 設定および管理ツール	5
第2章 クラスター操作	7
2.1. クォーラムの概要	7
2.2. フェンシングの概要	7
第3章 RED HAT HIGH AVAILABILITY アドオンリソース	9
3.1. RED HAT HIGH AVAILABILITY アドオンリソースの概要	9
3.2. RED HAT HIGH AVAILABILITY アドオンリソースクラス	9
3.3. リソースの監視	9
3.4. リソースの制約	9
3.5. リソースグループ	10
付録A RED HAT ENTERPRISE LINUX 6 HIGH AVAILABILITY アドオンからのアップグレード	11
A.1. リリース間での違いの概要	11
付録B 改訂履歴	13
索引	13

第1章 HIGH AVAILABILITY アドオンの概要

High Availability Add-On は、基幹実稼働サービスに、信頼性、スケーラビリティ、および可用性を提供するクラスターシステムです。以下のセクションでは、High Availability アドオンのコンポーネントおよび機能の概要を説明します。

- 「[クラスターの基礎](#)」
- 「[High Availability アドオンの紹介](#)」
- 「[Pacemaker アーキテクチャーコンポーネント](#)」

1.1. クラスターの基礎

クラスターとは連携してタスクを実行する 2 つ以上のコンピューター (ノードまたはメンバーと呼ばれます) です。クラスターには主に以下の 4 つのタイプがあります。

- ストレージ
- 高可用性
- 負荷分散
- ハイパフォーマンス

ストレージクラスターは、クラスター内のすべてのサーバーを対象に一貫性のあるファイルシステムイメージを提供し、サーバー群が同時に単一共有ファイルシステムに対する読み取りおよび書き込みを行えるようにします。ストレージクラスターは、アプリケーションのインストールやパッチ適用を 1 つのファイルシステムに制限することで、ストレージ管理を簡素化します。また、ストレージクラスターではクラスター全体のファイルシステムが使用されるため、アプリケーションデータの冗長コピーが必要なくなり、バックアップと障害回復が簡素化されます。High Availability アドオンは、Red Hat GFS2 (Resilient Storage アドオンの一部) との併用でストレージクラスターリングを提供します。

高可用性クラスターは、単一障害点を排除し、ノードが稼働しなくなった場合に、あるクラスターノードから別のクラスターノードにサービスをフェイルオーバーして、可用性が高いサービスを提供します。通常、高可用性クラスターのサービスは、(read-write でマウントされたファイルシステム経由で) データの読み取りや書き込みを行います。したがって、あるクラスターノードが別のクラスターノードからサービスの制御を引き継ぐ際に、高可能性クラスターでデータ整合性を維持する必要があります。高可用性クラスター内のノードの障害は、クラスター外にあるクライアントからは確認できません。また、高可用性クラスターはフェイルオーバークラスターと呼ばれることがあります。High Availability Add-On は、高可用性サービス管理コンポーネントの **Pacemaker** を介して、高可用性クラスターリングを提供します。

負荷分散クラスターは、ネットワークサービス要求を複数のクラスターノードに分散して、クラスターノード間の要求負荷のバランスを取ります。負荷分散は、負荷要件に応じてノード数を調節することができるため、コスト効率の良いスケーラビリティを提供します。負荷分散クラスター内の 1 つのノードが操作不能になった場合、負荷分散ソフトウェアは障害を検知して、要求を他のクラスターノードにリダイレクトします。負荷分散クラスター内のノードの障害は、クラスター外のクライアントからは見えません。負荷分散は Load Balancer アドオンで使用できます。

ハイパフォーマンスクラスターはクラスターノード群を使用して、同時演算を実行します。高パフォーマンスのクラスターでは、アプリケーションが並行して機能できるため、アプリケーションのパフォーマンスを強化できます。ハイパフォーマンスクラスターは、計算クラスターまたはグリッドコンピューティングとも呼ばれます。



注記

前述のクラスタータイプは基本的な設定を反映しています。ユーザーのニーズによっては、これらのクラスターの組み合わせが必要になることがあります。

さらに &RHEL; High Availability アドオンでは、高可用性サーバーの設定および管理のみのサポートが含まれます。高パフォーマンスクラスターはサポートしていません。

1.2. HIGH AVAILABILITY アドオンの紹介

High Availability アドオンは、パフォーマンス、高可用性、負荷分散、スケーラビリティ、ファイル共有、コスト効率などの各種ニーズに適した設定で導入できるソフトウェアコンポーネントの統合セットです。

High Availability Add-On は、以下の主要なコンポーネントで設定されています。

- クラスターインフラストラクチャー - クラスターとして連携するように、ノード群に基本的な機能 (設定ファイル管理、メンバーシップ管理、ロック管理、およびフェンシング) を提供します。
- 高可用性サービス管理: 1つのノードが操作不能になった場合にそのクラスターノードから別のクラスターノードへのサービスのフェイルオーバーを提供します。
- クラスター管理ツール - High Availability Add-On のセットアップ、設定、および管理を行うツール。これらのツールは、クラスターインフラストラクチャーのコンポーネント、高可用性とサービス管理のコンポーネント、およびストレージで使われます。

以下のコンポーネントで、High Availability Add-On を補完できます。

- Red Hat GFS2 (Global File System 2) - Resilient Storage Add-On に同梱され、High Availability Add-On で使用するクラスターファイルシステムを提供します。GFS2 により、ストレージがローカルで各クラスターノードに接続されているかのように、ブロックレベルにおいて、複数ノードでストレージを共有できるようになります。GFS2 クラスターファイルシステムを使用する場合は、クラスターインフラストラクチャーが必要になります。
- CLVM (Cluster Logical Volume Manager): Resilient Storage アドオンの一部で、クラスターストレージのボリューム管理を提供します。CLVM のサポートにもクラスターインフラストラクチャーが必要です。
- HAProxy - レイヤー 4 (TCP) およびレイヤー 7 (HTTP および HTTPS) サービスで高可用性負荷分散とフェイルオーバーを提供するルーティングソフトウェアです。

1.3. PACEMAKER の概要

High Availability アドオンのクラスターインフラストラクチャーは、コンピューター (ノードまたはメンバーと呼ばれる) の集合体が1つのクラスターとして連動して機能するための基本的な機能を提供します。クラスターインフラストラクチャーを使用してクラスターが形成された後は、ユーザーのクラスターリングニーズに合うように他のコンポーネントを使用できます (たとえば、GFS2 ファイルシステム上でファイルを共有するためのクラスターのセットアップまたはサービスフェイルオーバーのセットアップを行えます)。クラスターインフラストラクチャーでは、以下の機能が実行されます。

- クラスター管理
- ロック管理
- フェンシング

- クラスタ設定管理

1.4. PACEMAKER アーキテクチャーコンポーネント

Pacemaker で設定されたクラスタは、クラスタメンバーシップを監視する個別のコンポーネントデーモン、サービスを管理するスクリプト、および異なるリソースを監視するリソース管理サブシステムで設定されます。Pacemaker アーキテクチャーを形成するコンポーネントは、以下のとおりです。

Cluster Information Base (CIB)

XML を内部的に使用して、DC (Designated Coordinator) (CIB を介してクラスタのステータスと動作を格納および分散するために、Pacemaker により割り当てられたノード) から、他のすべてのクラスタノードに対して現在の設定とステータスの情報を分散し、同期する Pacemaker 情報デーモン。

Cluster Resource Management Daemon (CRMd)

Pacemaker クラスタリソースの動作は、このデーモンを介してルーティングされます。CRMd により管理されるリソースは、必要に応じてクライアントシステムが問い合わせることができます。また、リソースを移動したり、インスタンス化したり、変更したりできます。

各クラスタノードには、CRMd とリソースの間のインターフェイスとして動作する LRMd (Local Resource Manager daemon) も含まれます。LRMd は、起動、停止、ステータス情報のリレーなどのコマンドを、CRMd からエージェントに渡します。

Shoot the Other Node in the Head (STONITH)

多くの場合、電源スイッチとともにデプロイされる STONITH は、フェンス要求を処理する Pacemaker のクラスタリソースとして動作し、強制的にノードの電源をオフにし、クラスタからノードを削除してデータの整合性を確保します。STONITH は CIB で設定され、通常のクラスタリソースとして監視できます。

corosync

corosync は、コアメンバーシップと、高可用性クラスタのメンバー間の通信ニーズに対応するコンポーネントで、デーモンも同じ名前になります。これは、High Availability Add-On が機能するのに必要です。

corosync は、このようなメンバーシップとメッセージング機能のほかに、以下も提供します。

- クォーラムのルールおよび決定を管理します。
- クラスタの複数のメンバー間の調整機能、またはメンバー間で動作するメッセージング機能を提供します。そのため、インスタンス間で、ステートフルな情報またはその他の情報を通信できる必要があります。

1.5. PACEMAKER 設定および管理ツール

Pacemaker には、クラスタのデプロイメント、監視、および管理用の 2 つの設定ツールが含まれます。

pcs

pcs は、Pacemaker と Corosync ハートビートデーモンのすべての側面を制御できます。コマンドラインベースのプログラムである **pcs** は、以下のクラスタ管理タスクを実行できます。

- Pacemaker/Corosync クラスタの作成および設定

- 実行中のクラスタの設定を変更します。
- Pacemaker と Corosync の両方をリモートで設定し、起動、停止、およびクラスタのステータス情報の表示を行います。

pcsd Web UI

Pacemaker/Corosync クラスタを作成および設定するグラフィカルユーザーインターフェイス。コマンドラインベースの **pcs** ユーティリティと同じ機能を持ちます。

第2章 クラスター操作

本章では、さまざまなクラスターの機能と特徴について簡単に説明します。クラスタークォーラムの確立から分離のためのノードフェンシングまで、これらの異なる機能は、High Availability アドオンのコア機能から設定されます。

2.1. クォーラムの概要

クラスターの整合性と可用性を維持するために、クラスターシステムはクォーラムと呼ばれる概念を使用してデータの破損および損失を防ぎます。クラスターノードの過半数がオンラインになると、クラスターでクォーラムが確立されます。クラスターでクォーラムが確立されない場合は、障害によるデータ破損の可能性を小さくするために、Pacemaker はデフォルトですべてのリソースを停止します。

クォーラムは、投票システムを使用して確立されます。クラスターノードが通常どおり機能しない場合や、クラスターの他の部分との通信が失われた場合に、動作している過半数のノードが、問題のあるノードを分離するように投票し、必要に応じて、接続を切断して別のノードに切り替えてサービスを継続(フェンス)します。

たとえば、6 ノードクラスターで、4 つ以上のクラスターノードが動作している場合にクォーラムが確立されます。過半数のノードがオフラインまたは利用不可の状態になると、クラスターでクォーラムが確立されず、Pacemaker がクラスター化サービスを停止します。

Pacemaker のクォーラム機能により、スプリットブレイン(クラスターで通信が遮断され、各部分が別のクラスターとして動作し続ける現象。同じデータへの書き込みが行われ、破損または損失が発生することがあります)と呼ばれる現象が回避されます。

High Availability アドオンのクォーラムサポートは、**votequorum** と呼ばれる Corosync プラグインによって提供されます。このプラグインを使用すると、管理者がクラスター内の各システムに割り当てられた投票数でクラスターを設定でき、過半数の投票が存在する場合のみ、クラスターの操作の続行が許可されます。

過半数に満たない場合(内部通信ネットワークリンクが利用不可である 2 ノードクラスターであり、50% のクラスター分割になる場合など)は、**votequorum** がタイブレーカーポリシーを使用するよう設定できます(管理者は、ノード ID が最小の利用可能なクラスターノードと通信する残りのクラスターノードを使用してクォーラムを維持するために設定できます)。

2.2. フェンシングの概要

クラスターシステムでは、多くのノードが複数の重要な実稼働データを処理している場合があります。また、負荷が高いマルチノードクラスターのノードが不安定または利用不可になり管理者が対応を迫られることがあります。不安定なクラスターノードにより引き起こされた問題は、フェンシングポリシーを確立することにより緩和できます。

フェンシングとは、クラスターの共有ストレージからノードを切断することを指します。フェンシングによって共有ストレージからの I/O が遮断され、データの整合性が確保されます。クラスターのインフラストラクチャーでは、**STONITH** 機能を使用してフェンシングが実行されます。

Pacemaker はノードで障害が発生していることを検出した場合、そのことを他のクラスターインフラストラクチャーコンポーネントに通知します。障害の通知時に、**STONITH** によって、障害が発生しているノードが切断されます。また、他のクラスターインフラストラクチャーコンポーネントにより、行うべき操作(必要な復元の実行を含む)が決定されます。たとえば、ノード障害の通知時に、DLM または GFS2 では、**STONITH** が障害が発生したノードのフェンシングを完了するのを検出するまでアクティビティーが中断されます。障害が発生したノードのフェンシングが確認されると、DLM および GFS2 によって復元が実行されます。DLM の場合は障害が発生したノードのロックが解除され、GFS2 の場合は障害が発生したノードのジャーナルが復元されます。

STONITH を使用したノードレベルのフェンシングは、以下のものを含むさまざまなサポート対象フェンスデバイスで設定できます。

- 無停電電源装置 (UPS): 電源障害時にデバイスのフェンシングを行うために使用できるバッテリー内蔵デバイス
- Power Distribution Unit (PDU): クリーンな配電、フェンシング、および電源分離サービスのためにデータセンターで使用される、電源コンセントが複数あるデバイス
- ブレード電源管理デバイス: 障害発生時にクラスターノードのフェンシングを行うよう設定された、データセンターの専用システム
- Lights-Out デバイス: 管理者がローカルまたはリモートでクラスターノードの可用性を管理し、フェンシング、電源オン/オフ、および他のサービスを実行できる、ネットワークに接続されたデバイス

第3章 RED HAT HIGH AVAILABILITY アドオンリソース

本章では Red Hat High Availability のリソースとその操作の概要を提供します。

3.1. RED HAT HIGH AVAILABILITY アドオンリソースの概要

クラスターリソースは、クラスターサービスで管理するプログラム、データ、またはアプリケーションのインスタンスです。これらのリソースは、クラスター環境でリソースを管理する標準的なインターフェイスを提供するエージェントによって抽象化されます。この標準化は、業界で承認されたフレームワークとクラスに基づき、さまざまなクラスターリソースの可用性の管理がクラスターサービス自体に対して透過的になります。

3.2. RED HAT HIGH AVAILABILITY アドオンリソースクラス

Red Hat High Availability アドオンでは、以下の複数のリソースエージェントのクラスがサポートされています。

- LSB: Linux Standards Base エージェントは、LSB によりサポートされた準拠サービス (`/etc/init.d` のサービスとサービスの成功および失敗状態 (起動済み、停止中、実行中ステータス) を表す関連リターンコード) を抽象化します。
- OCF: Open Cluster Framework は、サーバー初期化スクリプトの作成および実行の標準や環境変数を使用したスクリプトの入力パラメーターなどを設定する LSB (Linux Standards Base) のスーパーセットです。
- systemd: Linux ベースシステムの最新のシステムサービスマネージャーである Systemd は、LSB と OCF のように初期化スクリプトではなくユニットファイルセットを使用します。これらのユニットは、管理者が手動で作成したり、サービス自体が作成および管理したりできます。Pacemaker は、OCF または LSB 初期化スクリプトを管理するのと同じ方法でこれらのユニットを管理します。
- Upstart: systemd と同様に、Upstart は、Linux 向けのシステム初期化マネージャーの1つです。Upstart は、systemd や初期化スクリプトでのユニットではなくジョブを使用します。
- STONITH: STONITH を使用してサービスとエージェントのフェンシングを行う専用リソースエージェント。
- Nagios: Nagios システムとインフラストラクチャー監視ツール向けのプラグインを抽象化するエージェント。

3.3. リソースの監視

リソースの健全性を維持するために、リソースの定義に監視操作を追加することができます。リソースの監視操作を指定しない場合は、`pcs` コマンドによってデフォルトで監視操作が作成されます。リソースエージェントでデフォルトの監視間隔が提供されない場合は、`pcs` コマンドにより 60 秒間隔の監視動作が作成されます。

3.4. リソースの制約

クラスター内のリソースの動作は、リソースの制約を設定することにより決定できます。以下の制約のカテゴリーを設定できます。

- 場所の制約 - リソースを実行できるノードを設定する

- 順序の制約: 順序の制約により、リソースが実行される順序が決定されます。
- コロケーションの制約 - 他のリソースに対して相対的なリソースの配置先を設定する

複数リソースをまとめて配置して、順番に起動するまたは逆順で停止する一連の制約を簡単に設定する方法として、Pacemaker ではリソースグループという概念に対応しています。

3.5. リソースグループ

クラスターの最も一般的な設定要素の1つがリソースセットです。リソースセットはまとめて配置し、順番に起動し、その逆順で停止する必要があります。リソースセットは一緒に配置し、順番に起動し、その逆順で停止する必要があります。この設定を簡略化するために、Pacemaker ではグループという概念がサポートされます。

pcs resource コマンドでリソースグループを作成し、グループに含めるリソースを指定します。グループが存在しない場合は、このコマンドによりグループが作成されます。グループが存在する場合は、このコマンドにより別のリソースがグループに追加されます。リソースは、このコマンドで指定された順序で起動し、その逆順で停止します。

付録A RED HAT ENTERPRISE LINUX 6 HIGH AVAILABILITY アドオンからのアップグレード

この付録では、Red Hat Enterprise Linux High Availability アドオンのリリース 6 からリリース 7 へのアップグレードについて簡単に説明します。

A.1. リリース間での違いの概要

Red Hat Enterprise Linux 7 High Availability アドオンには、高可用性システムのベースとなる新しいテクノロジー群が使用されています。これらのテクノロジーは Pacemaker および Corosync に基づいており、以前のリリースの High Availability アドオンに使用されていた CMAN および RGManger テクノロジーを完全に置き換えています。以下に、2つのリリース間の違いの一部を示します。リリース間の違いを包括的に確認する場合は、『Red Hat Enterprise Linux High Availability Add-On リファレンス』の付録 A.1 クラスター作成 - rgmanager と Pacemaker を参照してください。

- 設定ファイル: 以前は、クラスター設定が `/etc/cluster/cluster.conf` ファイルにありましたが、リリース 7 のクラスター設定は `/etc/corosync/corosync.conf` (メンバーシップおよびクォーラム設定の場合) と `/var/lib/pacemaker/cib/cib.xml` (クラスターノードおよびリソース設定の場合) にあります。
- 実行可能ファイル: 以前は、クラスターコマンドが **ccs** (コマンドラインの場合) と **luci** (グラフィカル設定の場合) にありました。Red Hat Enterprise Linux 7 High Availability アドオンでは、設定は **pcs** (コマンドラインの場合) と **pcsd** (デスクトップでの Web UI 設定の場合) で行われます。
- サービスの起動: 以前は、High Availability アドオンのサービスを含むすべてのサービスが、**service** コマンド (サービスを起動する) と **chkconfig** コマンド (システムブート時にサービスを起動するよう設定する) を使用して実行されていました。これは、すべてのクラスターサービス (**rgmanager**、**cman**、および **ricci**) に対して個別に設定する必要があります。以下に例を示します。

```
service rgmanager start
chkconfig rgmanager on
```

Red Hat Enterprise Linux 7 High Availability アドオンでは、以下のように **systemctl** によって手動による起動とブート時の自動的な起動の両方が制御され、すべてのクラスターサービスが **pcsd.service** でグループ化されます。以下に例を示します。

```
systemctl start pcsd.service
systemctl enable pcsd.service
pcs cluster start -all
```

- ユーザーアクセス: 以前は、root ユーザーまたは適切なパーミッションを持つユーザーが **luci** 設定インターフェイスにアクセスできます。すべてのアクセスには、ノードの **ricci** パスワードが必要です。

Red Hat Enterprise Linux 7 High Availability アドオンでは、**pcsd** Web UI が、共通のシステムユーザーである **hacluster** ユーザーとして認証する必要があります。root ユーザーは **hacluster** のパスワードを設定できます。

- クラスター、ノード、およびリソースの作成: 以前は、ノードの作成は、**ccs** (コマンドラインの場合) または **luci** グラフィカルインターフェイスを使用して行われていました。クラスターの作成とノードの追加は異なるプロセスです。たとえば、コマンドラインでクラスターを作成し、ノードを追加するには、以下のコマンドを実行します。

```
ccs -h node1.example.com --createcluster examplecluster  
ccs -h node1.example.com --addnode node2.example.com
```

Red Hat Enterprise Linux 7 High Availability アドオンでは、クラスター、ノード、およびリソースの追加は、**pcs** (コマンドラインの場合) または **pcs**d Web UI を使用して行われます。たとえば、コマンドラインでクラスターを作成するには、以下のコマンドを実行します。

```
pcs cluster setup examplecluster node1 node2 ...
```

- クラスターの削除: 以前は、管理者が **luci** インターフェイスから手動でノードを削除するか、各ノードの **cluster.conf** ファイルを削除することによりクラスターを削除していました。

Red Hat Enterprise Linux 7 High Availability アドオンでは、管理者が **pcs cluster destroy** コマンドを発行してクラスターを削除できます。

付録B 改訂履歴

改訂 7.1-1 7.7 GA 公開用ドキュメントの準備	Wed Aug 7 2019	Steven Levine
改訂 6.1-2 7.6 GA 公開用ドキュメントの準備	Thu Oct 4 2018	Steven Levine
改訂 5.1-2 7.5 GA 公開用ドキュメントの準備	Wed Mar 14 2018	Steven Levine
改訂 5.1-1 7.5 ベータ版公開用ドキュメントの準備	Wed Dec 13 2017	Steven Levine
改訂 4.1-3 7.4 GA 公開用ドキュメントバージョン	Tue Aug 1 2017	Steven Levine
改訂 4.1-1 7.4 ベータ版公開用ドキュメントの準備	Wed May 10 2017	Steven Levine
改訂 3.1-2 7.3 GA リリースのバージョン	Mon Oct 17 2016	Steven Levine
改訂 3.1-1 7.3 Beta 公開用ドキュメントの準備	Wed Aug 17 2016	Steven Levine
改訂 2.1-5 7.2 GA 公開用ドキュメントの準備。	Mon Nov 9 2015	Steven Levine
改訂 2.1-1 7.2 ベータ公開用ドキュメントの準備	Tue Aug 18 2015	Steven Levine
改訂 1.1-3 7.1 GA 向けバージョン	Tue Feb 17 2015	Steven Levine
改訂 1.1-1 7.1 ベータリリース向けバージョン	Thu Dec 04 2014	Steven Levine
改訂 0.1-9 7.0 GA リリース向けバージョン	Tue Jun 03 2014	John Ha
改訂 0.1-4 Red Hat Enterprise Linux 7 のベータ向けビルド	Wed Nov 27 2013	John Ha
改訂 0.1-1 Red Hat Enterprise Linux 7 の初版	Wed Jan 16 2013	Steven Levine

索引

シンボル

クォーラム, [クォーラムの概要](#)

フェンシング, [フェンシングの概要](#)

, [Red Hat Enterprise Linux 6 High Availability アドオンからのアップグレード](#)

, [Pacemaker の概要](#), [Pacemaker アーキテクチャーコンポーネント](#), [Pacemaker 設定および管理ツール](#), [Red Hat Enterprise Linux 6 High Availability アドオンからのアップグレード](#)

C

cluster

[クォーラム](#), [クォーラムの概要](#)

[フェンシング](#), [フェンシングの概要](#)

H

High Availability Add-On

[リリース 6 と 7 の相違点](#), [リリース間での違いの概要](#)