



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

Steven Levine

Red Hat Customer Content Services

slevine@redhat.com

法律上の通知

Copyright © 2018 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-sa/3.0/). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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	14

第1章 HIGH AVAILABILITY アドオンの概要

High Availability アドオンは、基幹実稼働サービスへ信頼性、スケーラビリティおよび可用性を提供するクラスター化されたシステムです。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（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サービスをあるクラスターノードから別のクラスターノードにフェイルオーバーすることにより、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内のサービスは、データの読み取りや書き込みを行います (read-write でマウントされたファイルシステム経由)。したがって、1つの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時点で、高可用性クラスターは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見えません (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 アドオンは、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。ハイパフォーマンスクラスターにより、アプリケーションは並行して稼働することができるようになるため、アプリケーションのパフォーマンスが向上します (ハイパフォーマンスクラスターは、演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 Red Hat Enterprise Linux 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、ハイパフォーマンスクラスターはサポートされていません。

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 アドオンは、以下の主要なコンポーネントから構成されています。

- 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 アドオンのセットアップ、設定、および管理を行うための設定および管理ツール。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います。

以下のコンポーネントで、High Availability アドオンを補足することができます。

- Red Hat GFS2 (Global File System 2): Resilient Storage アドオンの一部で、High Availability アドオンと共に使用するためのクラスターファイルシステムを提供します。GFS2を使用することにより、複数のノードがブロックレベルでストレージを共有でき、ストレージが各クラスターノードにローカルで接続されているようになります。GFS2のクラスターファイルシステムにはクラスターインフラストラクチャーが必要です。
- 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。
- Load Balancer アドオン: 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。Load Balancer アドオンは、負荷アルゴリズムを使用してクライアント要求を実サーバーに分散する冗長な仮想ルーターのクラスターで実行され、全体で仮想サーバーとして機能します。

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ーは、クラスター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から構成されます。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 Designated Coordinator (DC) (CIB を介してクラスター状態とアクションを格納および分散するために Pacemaker によって割り当てられたノード) から他のすべてのクラスターノードに対して現在の設定とステータス情報を分散し、同期する Pacemaker 情報デーモン。

Cluster Resource Management Daemon (CRMd)

Pacemaker クラスターリソースアクションは、このデーモンを介してルーティングされます。CRMd により管理されたリソースは、必要に応じてクライアントシステムによって問い合わせたり、移動したり、インスタンス化したり、変更したりできます。

各クラスターノードには、CRMd とリソース間のインターフェースとして動作する Local Resource Manager daemon (LRMd) 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 のクラスター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、クラスター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、通常のクラスター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、高可用性クラスター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です。これは、High Availability Add-On が機能するのに必要です。

corosync は、これらのメンバーシップとメッセージング機能のほかに、以下も実行します。

- クォーラムのルールおよび決定を管理します。
- クラスター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、クラスターのデプロイメント、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、以下のクラスター管理タスクを実行できます。

- Pacemaker/Corosync クラスターを作成および設定します。
- 実行中のクラスター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、およびクラスターのステータス情報の表示を行います。

pcsd Web UI

Pacemaker/Corosync クラスター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ース。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構成されます。

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である場合、クラスターでクォーラムが確立されます。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 (クラスターでクォーラムが確立されない場合)。

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他のクラスターとの通信を失った場合は、動作している過半数のノードがノードを分離するよう投票し、必要な場合は、修復のために接続を切断できます。

たとえば、6 ノードクラスターの場合、クォーラムは 4 つ以上のクラスターノードが動作している場合に確立されます。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ー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。監視の間隔はリソースエージェントによって決定されます。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 秒間隔の監視操作が作成されます。

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。設定できる制約のカテゴリーは以下のとおりです。

- 場所の制約: 場所の制約により、リソースを実行できるノードが決定されます。

- 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: コロケーションの制約により、他のリソースに対して相対的なリソースの配置先が決定されます。

一連のリソースを一緒に配置し、順番に起動し、逆順で停止する複数の制約を設定するために、Pacemaker ではリソースグループという概念がサポートされます。

3.5. リソースグループ

クラスターの最も一般的な構成要素の1つがリソースセットです。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、このコマンドによってグループが作成されます。グループが存在する場合は、このコマンドによってグループにリソースが追加されます。リソースはこのコマンドで指定した順序で起動し、起動とは逆の順序で停止します。

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。この場合は、以下のように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を行う必要がありました。

```
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** にまとめられます。

```
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** 設定インターフェースにアクセスでき、すべてのアクセスでノードの **ricci** パスワードが必要でした。

Red Hat Enterprise Linux 7 High Availability アドオンでは、**pcsd** Web UI で共通のシステムユーザーであるユーザー **hacluster** 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ー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** インターフェー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 7.1-1.1 翻訳ファイルを XML ソースバージョン 7.1-1 と同期	Mon Jan 4 2021	Red Hat
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 公開用バージョン	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](#)

クラスター

[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](#)

[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](#)

H

High Availability アドオン

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