Red Hat Enterprise Linux 6

ロードバランサーの管理

Red Hat Enterprise Linux 向け Load Balancer Add-On

エディッション 6

Last Updated: 2017-10-12

Red Hat Enterprise Linux 6 ロードバランサーの管理

Red Hat Enterprise Linux 向け Load Balancer Add-On
エディッション 6

法律上の通知

概要

Load Balancer Add-On のシステムを構築することでルーティングや負荷分散機能向けの特殊サーバー LVS (Linux Virtual Server) を使用する実か動揺サービスに対して可用性および拡張性の高いソリューションを実現します。本ガイドでは、Red Hat Enterprise Linux と Red Hat Enterprise Linux 6 向け Load Balancer Add-On を使用したパフォーマンス性の高いシステムとサービスの構成について説明しています。

目次

はじめに

本書では、ロードバランサーのアドオンコンポーネントをインストール、設定、管理する情報を提供します。Load Balancer Add-Onは、トラフィックをサーバープールに送り出す特別のルーティング技術によって、負荷を分散します。

本書は、Red Hat Enterprise Linux について高度な運用知識があり、クラスター、ストレージ、サーバーコンピューティングの概念を理解している方を対象としています。

本書は以下のような構成になっています。

- 1章*Load Balancer Add-On の概要*
- 2章*Load Balancer Add-On の初期設定*
- 3章*Load Balancer Add-On の設定*
- 4章***Piranha Configuration Tool** を使った Load Balancer Add-Onの設定*
- 付録A *High Availability アドオンを使った Load Balancer Add-Onの使用*

Red Hat Enterprise Linux 6 の詳細については、以下の資料を参照してください。

- 『Red Hat Enterprise Linux インストールガイド』 – Red Hat Enterprise Linux 6 のインストールに関する情報を提供しています。
- 『Red Hat Enterprise Linux 導入ガイド』 – Red Hat Enterprise Linux 6 の導入、設定、管理に関する情報を提供しています。

Red Hat Enterprise Linux 6 の Load Balancer Add-Onと関連製品についての詳細は、以下の資料を参照してください。

- 『Red Hat Cluster Suite の概要 』 – High Availability アドオン、 Resilient Storage アドオン、Load Balancer Add-Onに関する高レベルでの概要です。
- 『High Availability アドオンの設定と管理』 は、Red Hat Enterprise Linux 6 向けの High Availability アドオン (別名: Red Hat Cluster) の設定と管理について説明しています。
- 『論理ボリュームマネージャの管理』 – 論理ボリュームマネージャ (LVM) について説明しており、クラスター化された環境における LVM の実行に関する情報が含まれます。
- 『Global File System 2: 設定と管理 』 – Red Hat Resilient Storage アドオン (別名: Red Hat Global File System 2) のインストール、設定、および保守に関する情報を提供しています。
- 『DM Multipath』 – Red Hat Enterprise Linux 6 のデバイスマッパーマルチパスの機能の使用法に関する情報を提供します。
- 『リリースノート』 – Red Hat 製品の現在のリリースに関する情報を提供します。

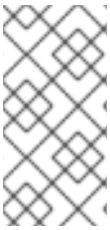
本ガイドを含め Red Hat のドキュメントについては HTML 版、PDF 版、EPUB 版がオンラインの http://access.redhat.com/documentation/docs でご覧いただけます。

1. フィードバック

本書内で誤字・脱字を発見された場合や、本書改善のためのご意見がございましたら、弊社にご連絡ください。その場合は、製品 **Red Hat Enterprise Linux 6** 、コンポーネント**doc-Load_Balancer_Administration**、およびバージョン番号 6.1 で Bugzilla (http://bugzilla.redhat.com/bugzilla/) 内でご報告ください。

本書改善のご提案がある場合は、できるだけ詳しい説明をお願いします。エラーを発見された場合は、該当セクションの番号と前後の文の一部を含めていただくと弊社でより迅速に発見することができます。

第1章 LOAD BALANCER ADD-ON の概要



注記

Red Hat Enterprise Linux 6.6 からは Piranha 負荷分散機能ソフトウェアに加え HAProxy および keepalived についても対応するようになります。 HAProxy および keepalived で Red Hat Enterprise Linux を設定する方法については Red Hat Enterprise Linux 7 のロードバランサーの管理に関するドキュメントを参照してください。

複数の実サーバー全体の IP 負荷を分散させる目的で Linux Virtual Server (LVS) を提供するソフトウェアコンポーネントをひとつにまとめたセットが Load Balancer Add-On です。*アクティブ LVS ルーター*および *バックアップ LVS ルーター*で稼働します。アクティブ LVS ルーターには以下の 2 つの役割があります。

- 複数の実サーバー全体の負荷分散
- それぞれの実サーバー上にあるサービスの整合性チェック

バックアップ LVS ルーターはアクティブ LVS ルーターを監視し、アクティブ LVS ルーターに障害が発生した場合に引き継ぎを行います。

本章では以下のセクションに沿って Load Balancer Add-On の各コンポーネントおよび機能の概要を説明していきます。

- 「Load Balancer Add-On の基本的な設定」
- 「Load Balancer Add-Onの三層構成」
- 「Load Balancer Add-On スケジューリング機能の概要」
- 「ルーティングメソッド」
- 「永続性とファイアウォールマーク」
- 「Load Balancer Add-On – ブロックダイアグラム」

1.1. LOAD BALANCER ADD-ON の基本的な設定

二層から成るシンプルな設定を 図1.1「Load Balancer Add-On の基本的な設定」 に示します。第一層はアクティブ LVS ルーターが 1 つ、バックアップ LVS ルーターが 1 つで構成されています。各 LVS ルーターには、インターネット用インターフェースとプライベートネットワーク用インターフェースの 2 種類のネットワークインターフェースがあり、2 つのネットワーク間のトラフィックを規制します。この例では、アクティブルーターが *Network Address Translation* (*NAT*) を使ってインターネットからのトラフィックを第二層にある複数の実サーバー (サーバー数は状況に応じて可変) にダイレクトすると、実サーバーが必要とされるサービスを提供します。つまり、この例の実サーバーは専用のプライベートネットワークセグメントに接続されているため、パブリックのトラフィックはすべてアクティブ LVS ルーターを経由することになります。外部からはサーバーは一つのエンティティに見えます。

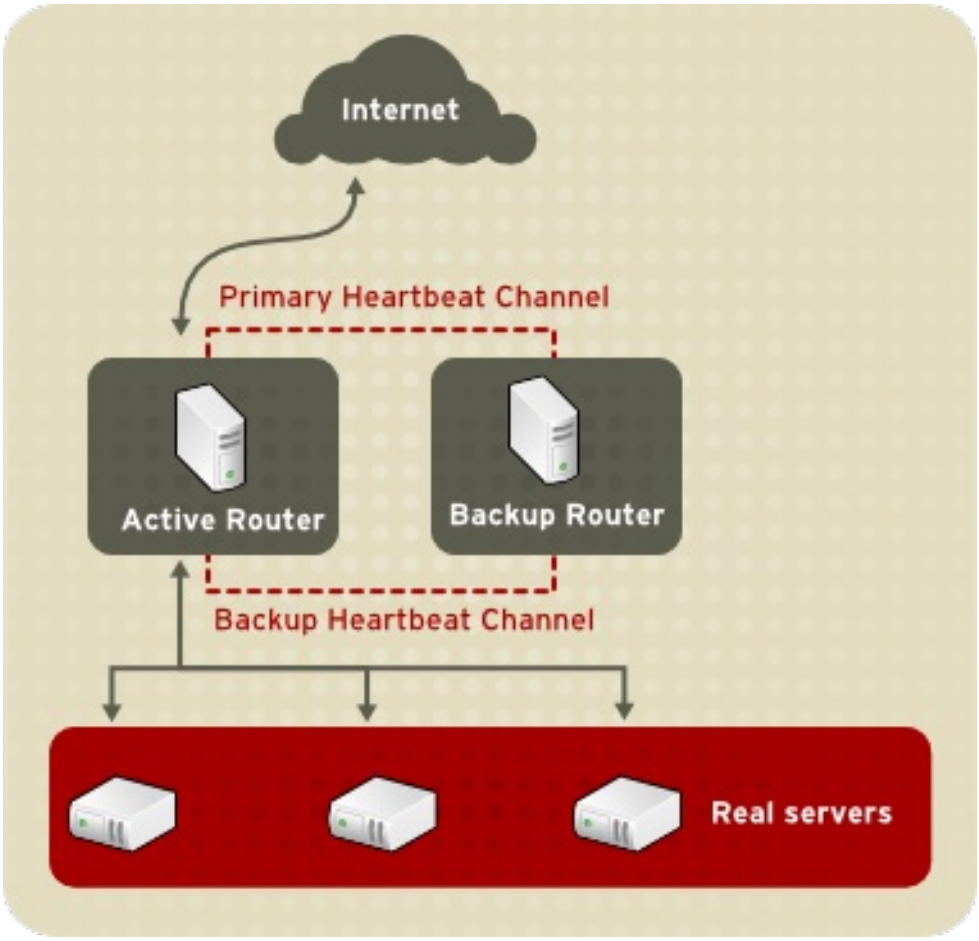


図1.1 Load Balancer Add-On の基本的な設定

LVS ルーターに届いたサービス要求は *仮想 IP* アドレス (*VIP*) に送信されます。VIP とはサイト管理者が www.example.com など完全修飾ドメイン名を関連付けている公的にルーティングが可能なアドレスのため、1 *仮想サーバー*または複数の *仮想サーバー*に割り当てられます。仮想サーバーとは特定の仮想 IP でリッスンするよう設定されたサービスになります。**Piranha Configuration Tool** を使って仮想サーバーを設定する方法については 「**VIRTUAL SERVERS**」 を参照してください。フェイルオーバーの際には VIP アドレスは次の LVS ルーターに移行されるため、引き続き利用することができます (別名: **フローティング IP アドレス**)。

LVS ルーターをインターネットに接続させるデバイスに複数の VIP アドレスをエイリアスさせることができます。例えば、インターネットに eth0 を接続する場合、複数の仮想サーバーを **eth0:1** にエイリアスすることができます。または、各仮想サーバーをサービスごと異なるデバイスに関連付けることもできます。例えば、HTTP トラフィックは **eth0:1** で FTP トラフィックは **eth0:2** でそれぞれ処理することができます。

アクティブにできるのは一度にひとつの LVS ルーターのみです。アクティブルーターの役割は、仮想 IP アドレスからのサービス要求を実サーバーにリダイレクトすることです。リダイレクトは対応している 8 種類の負荷分散アルゴリズムのいずれかをベースとします。これについては 「Load Balancer Add-On スケジューリング機能の概要」 で説明しています。

アクティブルーターはシンプルな *send/expect のスクリプト*を使って実サーバー上の特定サービスの全体的な健全性を動的に監視します。HTTPS や SSL といった動的データを必要とするサービスの健全性確認を補助する目的で、外部の実行可能ファイルを管理者側で呼び出すこともできます。実サーバー上のサービスが正常に機能していない場合、アクティブルーターは正常な動作に戻るまでそのサーバーへのジョブの送信を停止します。

予備システムの役割を果たすのがバックアップルーターです。LVS ルーターはプライマリーの外部パブリックインターフェースを使って定期的にハートビートメッセージを交換、またフェイルオーバーの

際はプライベートインターフェースを使用します。バックアップノード側が想定している間隔でハートビートメッセージを受信できなかった場合、フェイルオーバーを開始してアクティブルーターの役割を引き継ぎます。フェイルオーバー時、障害が発生したルーターで提供していた VIP アドレスを引き継ぐため *ARP スプーフィング*と呼ばれる技術を使用します。バックアップ LVS ルーターにより障害の発生したノード宛てに送信される IP パケットの宛先はバックアップ LVS ルーターであると通知されます。障害が発生したノードがアクティブに戻ると、バックアップノードは再びホットバックアップの役割を引き継ぐことになります。

静的な Web ページのように頻繁には変更が行われないデータを提供する場合は、実サーバー同士がノード間でデータの自動同期を行わないため、 図1.1「Load Balancer Add-On の基本的な設定」 のシンプルな二層設定が最適な設定になります。

1.1.1. 実サーバー間でのデータの複製とデータの共有

Load Balancer Add-On には実サーバー間で同一データを共有するためのビルトインコンポーネントがないため、管理側で行えるのは以下の 2 つのオプションになります。

- 実サーバープール全体でデータを同期する
- 共有データにアクセスするるための第三層をトポロジーに追加する

実サーバー上へのデータのアップロードやデータの変更が限られたユーザーにしか許可されないようなサーバーの場合は最初のオプションが適しています。電子商取引サイト、インターネットによる通信販売など、データの変更を多くのユーザーに許可するような設定の場合には第三層を追加するオプションの方がよいでしょう。

1.1.1.1. データを同期するよう実サーバーを設定する

実サーバーのプール全体でデータを同期させる方法はいろいろあります。例えば、Web エンジニアが任意のページに更新を加えた場合、そのページがすべてのサーバーに同時に送られるようにするシェルスクリプトを導入する、また **rsync** などのプログラムを使って全ノードを対象として変更が加えられたデータの複製を一定の間隔で作成することもできます。

しかし、ファイルがユーザーによって絶えずアップロードされていたり、データベースのトランザクションが発行されたりしているような過剰負荷の構成の場合、こうしたデータの同期方法では最適な動作は期待できません。負荷が高い構成の場合には *三層のトポロジー*が理想的なソリューションです。

1.2. LOAD BALANCER ADD-ONの三層構成

Load Balancer Add-On の一般的な三層トポロジーを 図1.2「Load Balancer Add-Onの三層構成」 に示します。インターネットからの要求がアクティブ LVS ルーターにより実サーバーのプールにルーティングされます。各実サーバーはネットワークを経由して共有データのソースにアクセスします。

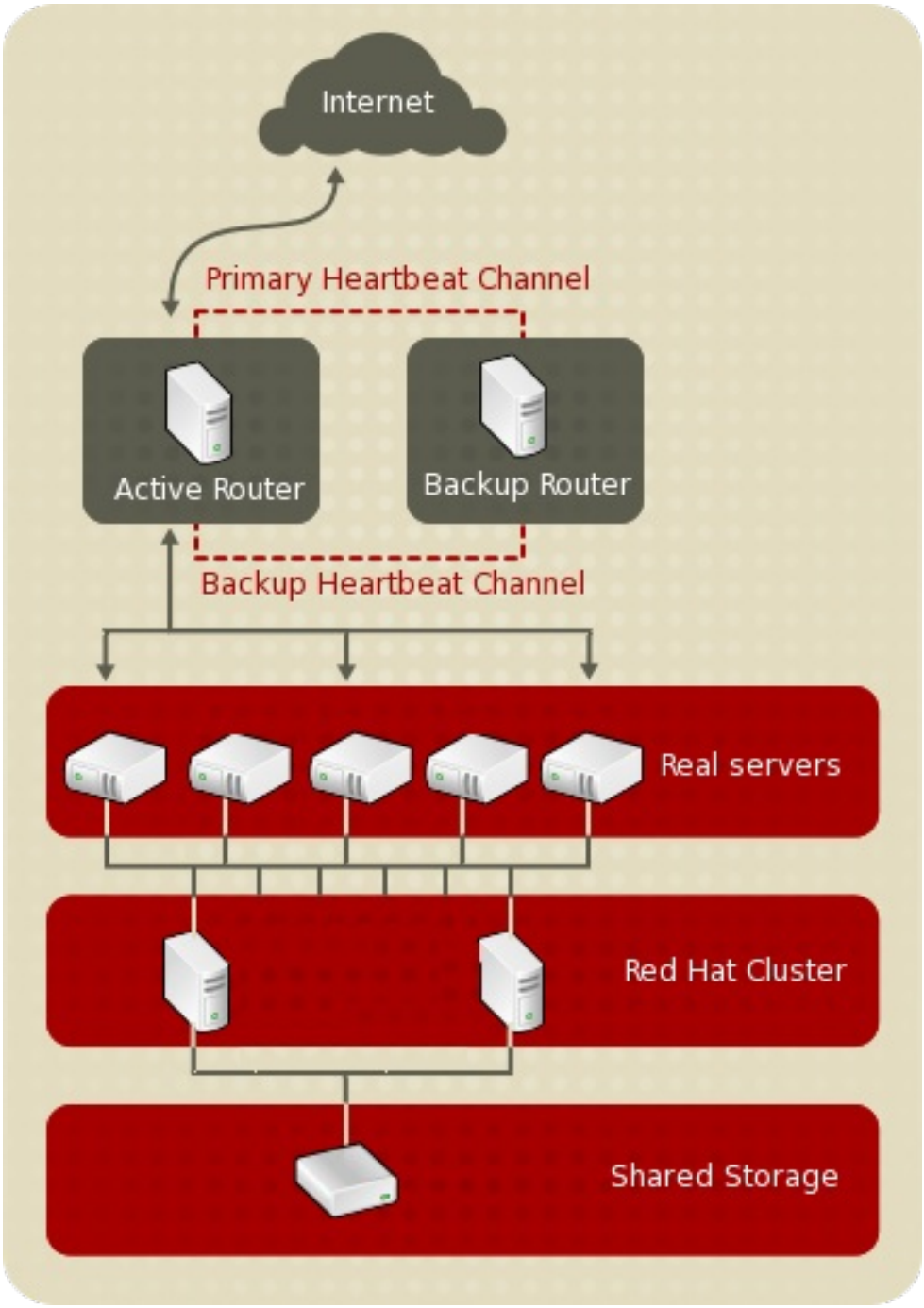


図1.2 Load Balancer Add-Onの三層構成

アクセス可能なデータが中央となる高可用性サーバーに格納されていて、実サーバーはエクスポートした NFS ディレクトリーまたは Samba 共有を使ってそのデータにアクセスするような構成がトラフィック量の多い FTP サーバーには理想的な構成と言えます。また、トランザクションのため中央となる高可用性データベースにアクセスを行うような Web サイトにもこのトポロジーをお勧めします。また、Load Balancer Add-Onでアクティブ-アクティブ設定を使用すると、これら両方の役割を同時に果たす高可用性クラスターを設定することもできます。

上記の例では、第三層で Load Balancer Add-Onを使う必要はありませんが、高可用性のソリューションを使用しないと重大な単一点障害をもたらすことになります。

1.3. LOAD BALANCER ADD-ON スケジューリング機能の概要

Load Balancer Add-On を使う利点の一つは、柔軟性のある IP レベルの負荷分散を実サーバーのプールで実行できることです。この柔軟な負荷分散は Load Balancer Add-Onの設定時に選択できるスケ

ジューリングアルゴリズムの多様性により実現しています。DNS の階層性質やクライアントマシンによるキャッシングが不均衡な負荷につながる *ラウンドロビン DNS*などの柔軟性に乏しい方法に比べ、Load Balancer Add-On の負荷分散は非常に優れています。また、ネットワークパケットレベルでの負荷分散により計算オーバーヘッドが最小限に抑えられ拡張性が高まるため、LVS ルーターが使用する低レベルのフィルタリングの方がアプリケーションレベルの要求転送より優れていると言えます。

スケジューリング機能を使用すると、サービス要求をルーティングする際にアクティブルーター側で実サーバーのアクティビティや、オプションで管理者が割り当てた *重み* 要素などを考慮させることができます。重み割り当てを使うことで各マシンに任意の優先順位が与えられます。この形式のスケジューリング機能を使うと、各種ハードウェアとソフトウェアの組み合わせを使用した複数の実サーバーから成るグループを作成することができ、アクティブルーターが負荷を各実サーバーに均等に分散することができます。

Load Balancer Add-Onのスケジューリングのメカニズムは *IP 仮想サーバー*または *IPVS* モジュールと呼ばれるカーネルパッチの集合で提供されます。このモジュールにより 、ひとつの IP アドレスで複数のサーバーが正しく動作するよう設計されている *layer 4* (*L4*) トランスポート層の切り替えが可能になります。

実サーバーへのパケットを効率的に追跡、ルーティングするため、IPVS はカーネル内に *IPVS テーブル*を構築します。アクティブ LVS ルーターにより仮想サーバーアドレスとプール内の実サーバー間での要求のリダイレクトに使用されます。IPVS テーブルはクラスターメンバーの可用性に応じてそのメンバーの追加や削除を行う *ipvsadm* というユーティリティで継続的に更新されます。

1.3.1. スケジューリングのアルゴリズム

IPVS テーブルの構造は管理者が仮想サーバーに選択するスケジューリングアルゴリズムによって異なります。クラスター化できるサービスタイプとこれらのサービスのスケジューリングでの柔軟性を最大限に利用するため、Red Hat Enterprise Linux では以下のようなスケジューリングアルゴリズムを提供しています。スケジューリングアルゴリズムの割り当て方については 「**VIRTUAL SERVER** サブセクション」 を参照してください。

Round-Robin Scheduling

要求を順番に実サーバーのプールに振り分けます。このアルゴリズムを使うと、処理能力や負荷に関係なくすべての実サーバーが平等に扱われます。このモデルはラウンドロビン DNS に似ていますが、ホストベースではなくネットワーク接続をベースにするためより細かな調整が可能です。また、Load Balancer Add-On のラウンドロビンスケジューリングはキャッシュされた DNS クエリーが原因で負荷分散が偏ることもありません。

Weighted Round-Robin Scheduling

要求を順番に実サーバーのプールに振り分けますが、より処理能力の高いサーバーに対して多くのジョブを振り分けます。処理能力はユーザーが割り当てた重み要素で示され、動的な負荷情報で上方修正または下方修正されます。実サーバーに重みを付ける方法については 「サーバーの重み付けとスケジューリング」 を参照してください。

プール内の実サーバー間で処理能力に大幅な違いがある場合は、重み付きラウンドロビンスケジューリングが適しています。ただし、要求負荷が大きく変化する場合は、重みの大きいサーバーが割り当て以上の要求に応じる可能性があります。

Least-Connection

実際の接続が少ない実サーバーにより多くの要求を振り分けます。IPVS テーブルで実サーバーへのライブ接続を継続的に追跡するため、Least-Connection は動的なスケジューリングアルゴリズムになります。要求負荷の変化が大きい場合に適しています。このアルゴリズムは各メンバーノードの処理能力がほぼ同じで大差がないような実サーバープールに最適です。サーバーグループの処理能力が異なる場合は weighted least-connection スケジューリングの方が適しています。

Weighted Least-Connections (デフォルト)

処理能力に対して相対的に実際の接続が少ないサーバーにより多くの要求を振り分けます。処理能力はユーザーが割り当てた重み要素で示され、動的な負荷情報で上方修正または下方修正されます。実サーバーのプールに異なる処理能力のハードウェアがある場合には、重みを付けることで理想的なアルゴリズムになります。実サーバーへの重みの付け方については 「サーバーの重み付けとスケジューリング」 を参照してください。

Locality-Based Least-Connection Scheduling

接続先 IP に対して相対的に実際の接続が少ないサーバーにより多くの要求を振り分けます。プロキシキャッシュサーバーのクラスターでの使用を目的として設計されています。任意の IP アドレスのパケットをその IP アドレスのサーバーに送信します。ただし、そのサーバーの負荷がその処理能力を超えている一方、別のサーバーの負荷は処理能力の半分に留まっている場合は、IP アドレスを負荷の一番少ない実サーバーに割り当てます。

Locality-Based Least-Connection Scheduling with Replication Scheduling

接続先 IP に対して相対的に実際の接続が少ないサーバーにより多くの要求を振り分けます。このアルゴリズムもプロキシキャッシュサーバーのクラスターでの使用を目的として設計されています。Locality-Based Least-Connection スケジューリングとの違いは、目的 IP アドレスを実サーバーノードのサブセットにマッピングする点です。要求はマッピングされたサブセット内で接続数が最少となるサーバーにルーティングされます。接続先 IP のノードがすべて処理能力を越えてしまっている場合は、実サーバーのプール全体で接続が一番少ないサーバーをその接続先 IP の実サーバーサブセットに追加して、その接続先 IP アドレスの新しいサーバーを複製します。一方、過剰な複製を防ぐため、負荷の最も高いノードが実サーバーのサブセットから外されます。

Destination Hash Scheduling

静的なハッシュテーブル内の接続先 IP を検索して、実サーバーのプールに要求を振り分けます。プロキシキャッシュサーバーのクラスターでの使用を目的として設計されています。

Source Hash Scheduling

静的なハッシュテーブル内のソース IP を検索して、実サーバーのプールに要求を振り分けます。複数のファイアウォールが設定される LVS ルーター向けに設計されています。

1.3.2. サーバーの重み付けとスケジューリング

Load Balancer Add-On の管理者は実サーバープール内の各ノードに対して *重み* を割り当てることができます。整数値で設定する重みは **重み付けを認識する** スケジューリングアルゴリズムならいずれのアルゴリズムにも組み入れられます (weighted least-connection など)。また、LVS ルーターで異なる処理能力のハードウェアにより均等に負荷を分散する場合に役立ちます。

重みは互いに相対的な割合として機能します。例えば、ある実サーバーに 1 の重みを付け、別のサーバーには 5 の重みを付けた場合、1 の重みを付けたサーバーが 1 回接続される度、5 の重みを付けたサーバーは 5回接続されます。実サーバーのデフォルトの重み値は 1 です。

実サーバープール内でそれぞれ異なるハードウェア構成のノードに重みを付けるとクラスターでの負荷分散をより効率的に行う場合に役立ちますが、 weighted least-connection スケジューリングで仮想サーバーが設定されている際、任意の実サーバーがその実サーバープールに挿入されると、一時的に不均衡が生じる場合があります。例えば、実サーバープールに 3 台のサーバーがあったとします。サーバー A と B に 1 の重みが付けられ、サーバー C には 2 の重みが付けられていたとします。サーバー C が何らかの理由でダウンした場合、放棄された負荷がサーバー A と B に均等に分散されます。しかし、サーバー C がオンラインに復帰すると、LVS ルーター側でサーバー C の接続がまったくないと判断され、サーバー A および B と同等になるまで着信要求をすべてサーバー C に集中的に振り分けることになります。

この現象を避けるため、管理側で仮想サーバーを *休止* サーバーにすることができます。これを利用すると、上述のような実サーバー C は仮想サーバーテーブルから削除される代わりに重みが 0 に設定されます。これにより実質このサーバーは無効になります。実サーバー C が利用できる状態になると、オリジナルの重み値に戻され再び有効になります。

1.4. ルーティングメソッド

Red Hat Enterprise Linux では Load Balancer Add-On に *ネットワークアドレス変換* (*NAT ルーティング*) を使用します。使用できるハードウェアを活用し、既存ネットワークに Load Balancer Add-On を統合する際に優れた柔軟性を得ることができます。

1.4.1. NAT ルーティング

インターネットとプライベートネットワーク間での要求の移動に NAT ルーティングを利用している Load Balancer Add-On を 図1.3「NAT ルーティングを実装した Load Balancer Add-On」 に示します。

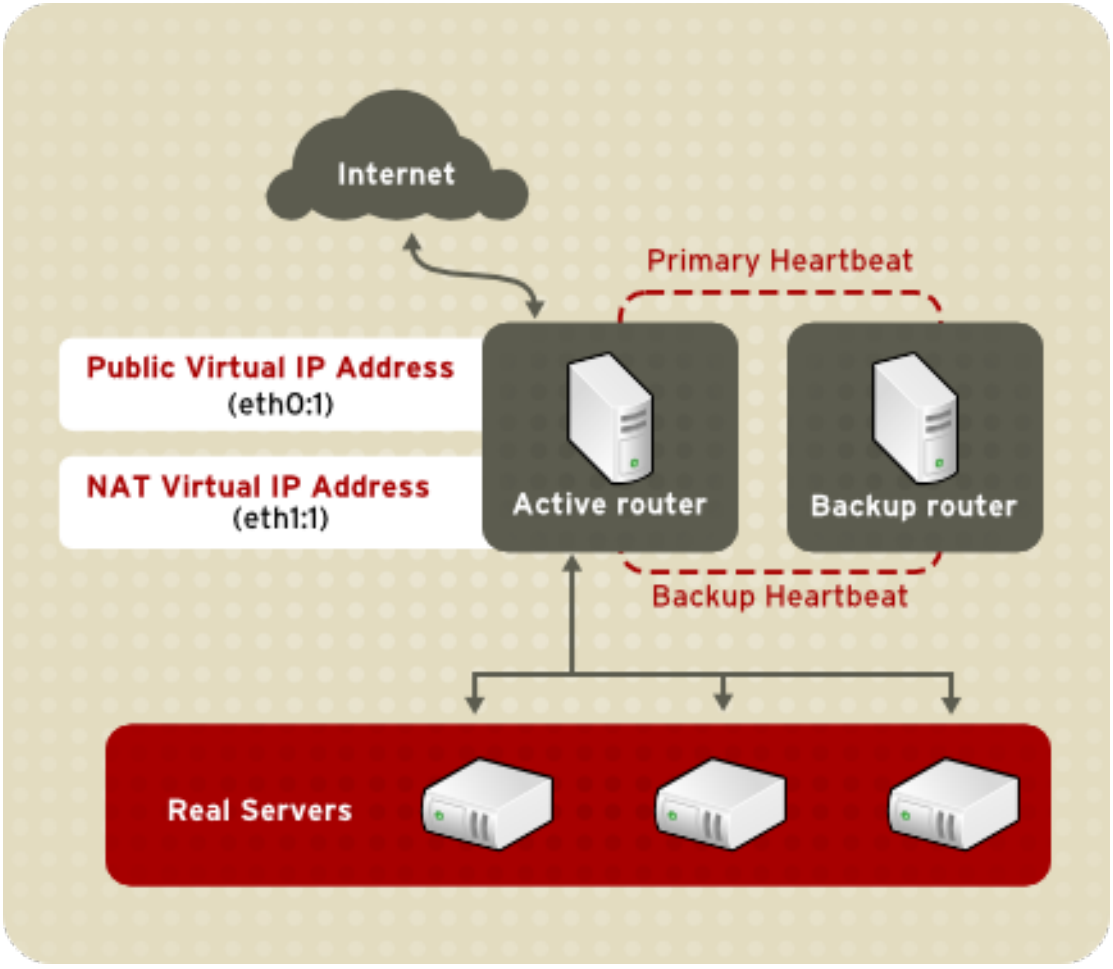


図1.3 NAT ルーティングを実装した Load Balancer Add-On

この例の場合、アクティブ LVS ルーターには 2 種類の NIC があります。インターネット用の NIC には eth0 に *実際の IP アドレス* を持たせフローティング IP アドレスを eth0:1 にエイリアスしています。プライベートネットワーク用の NIC には eth1 に実際の IP アドレスを持たせフローティング IP アドレスを eth1:1 にエイリアスしています。フェイルオーバーが発生すると、インターネット側仮想インターフェースとプライベートネットワーク側仮想インターフェースがバックアップ LVS ルーターに同時に引き継がれます。プライベートネットワークに配置している実サーバーはすべて NAT ルーターのフローティング IP をデフォルトのルートとして使用しアクティブ LVS ルーターと通信を行うため、インターネットからの要求への応答に支障をきたすことはありません。

また、LVS ルーターのパブリックのフローティング IP アドレスとプライベートの NAT フローティング

IP アドレスは物理的な NIC にそれぞれエイリアスされています。それぞれのフローティング IP アドレスを LVS ルーターノード上の各物理デバイスに関連付けることはできますが、NIC を 3 つ以上持たせる必要はありません。

このトポロジーを使うと、アクティブ LVS ルーターは要求を受信して適切なサーバーにルーティングします。実サーバーはその要求を処理してパケットを LVS ルーターに返します。LVS ルーターはネットワークアドレス変換を使ってパケット内の実サーバーのアドレスを LVS ルーターのパブリック VIP アドレスに置き換えます。実サーバーの本当の IP アドレスは要求を行っているクライアントからは見えないよう隠しているため、*IP マスカレード*と呼ばれます。

NAT ルーティングを使用する場合は、実サーバーにするマシンの種類や稼働させるオペレーティングシステムの種類に制限はありません。ただし、発信要求および着信要求のいずれも LVS ルーターで処理しなければならないため、大規模なクラスター導入の場合には LVS ルーターがボトルネックとなる場合があります。

1.4.2. ダイレクトルーティング

ダイレクトルーティングを使用する Load Balancer Add-On 設定を構築すると、他の Load Balancer Add-On のネットワークトポロジーよりもパフォーマンス性が高くなります。発信パケットを LVS ルーター経由で渡すのではなく、実サーバーでパケットを処理、要求元のユーザーに直接ルーティングすることができます。ダイレクトルーティングは、LVS ルーターのジョブを着信パケットの処理だけに特化させることで、ネットワークパフォーマンスの問題が発生する可能性を低減します。

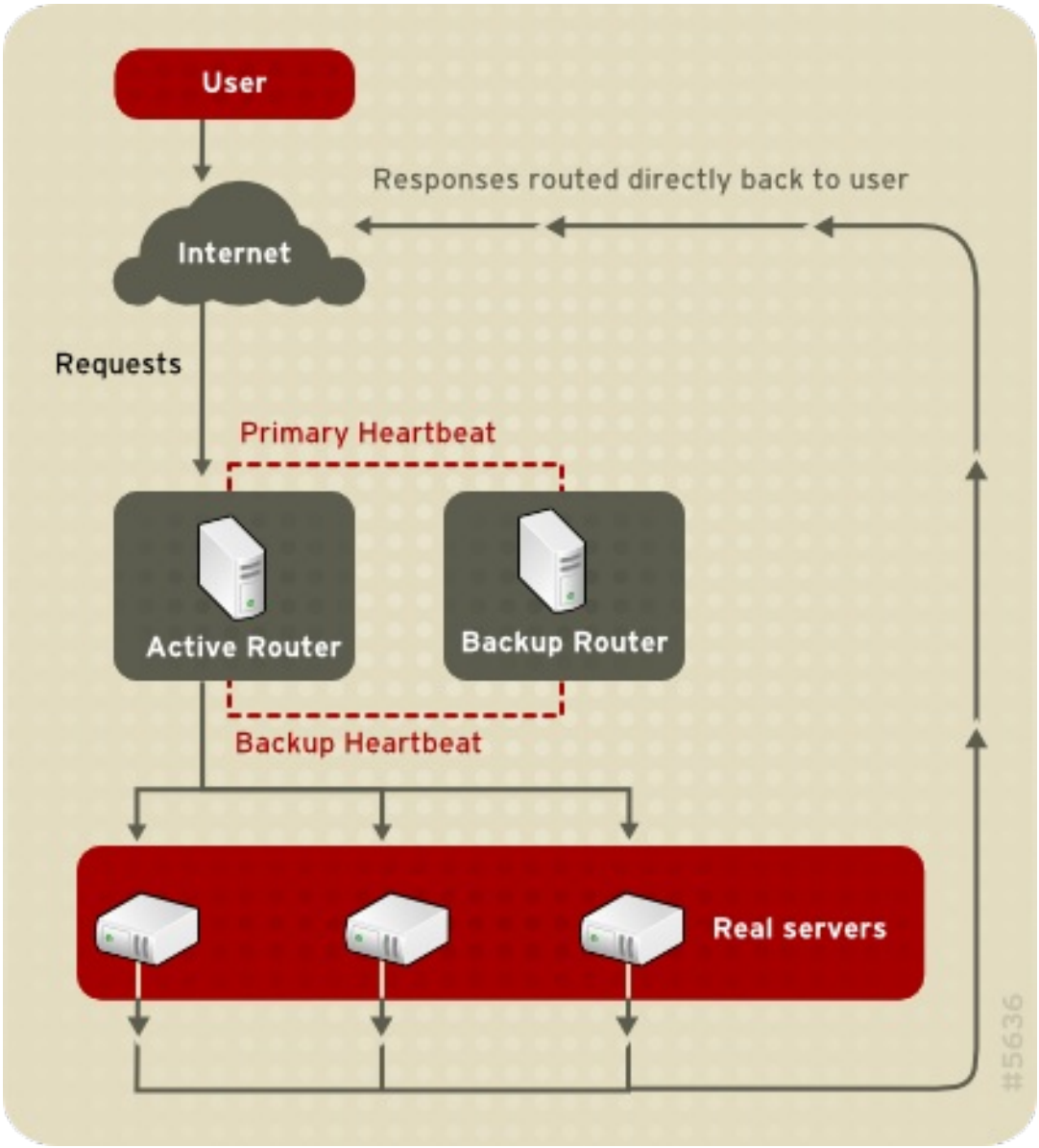


図1.4 ダイレクトルーティングで実装した Load Balancer Add-On

一般的なダイレクトルーティングによる Load Balancer Add-On の設定では、LVS ルーターは仮想 IP (VIP) を使って着信サーバー要求を受け取り、スケジューリングアルゴリズムを使用してこの要求を実サーバーにルーティングします。実サーバーはこの要求を処理すると LVS ルーターを迂回して直接クライアントに応答を送信します。実サーバーからクライアントへの発信パケットのルーティングを LVS ルーターに行わせると、ネットワーク負荷が高い環境ではボトルネックとなる可能性があります。ダイレクトルーティングメソッドの場合には、この負担を LVS ルーターにかけることなく実サーバーを追加できるという拡張性を備えています。

1.4.2.1. ダイレクトルーティングと ARP 制限

Load Balancer Add-On でダイレクトルーティングを使用すると役に立つ利点が多くありますが、一方で制限もあります。ダイレクトルーティングでの Load Balancer Add-On に関する最も一般的な問題として *アドレス解決プロトコル*(ARP) に関する問題があります。

インターフェース上のクライアントは要求を IP アドレスに送信します。ネットワークルーターは ARP を使って IP アドレスをマシンの MAC アドレスに関連付けることで要求を宛先に送信します。ARP 要求がネットワークに接続されているすべてのマシンにブロードキャストされ、IP アドレスと MAC アドレスの正しい組み合わせを持つマシンがパケットを受け取ることになります。IP と MAC の関連性は ARP キャッシュに保存され、定期的に消去と再保存が行われます (通常 15 分ごと)。

任意の IP アドレスに送信されるクライアント要求はそれを処理する MAC アドレスに関連付けられなければなりません。Load Balancer Add-Onシステムの仮想 IP アドレスも MAC アドレスに関連付けなければなりません。これがダイレクトルーティングによる Load Balancer Add-On 設定での ARP 要求で問題になります。LVS ルーターと実サーバーはいずれも同じ VIP が与えられているため、ARP 要求はこの VIP に関連付けられているマシンすべてに対してブロードキャストされます。このため、VIP が実サーバーのいずれか 1 台に直接関連付けられてしまい要求を直接処理してしまったり、LVS ルーターを完全に迂回してしまい Load Balancer Add-Onを設定している意味がなくなってしまうなどの問題の原因となることがあります。

この問題を解決するには、着信要求が実サーバーではなく、必ず LVS ルーターに送信されるようにすることです。**arptables_jf** または **iptables** パケットフィルタリングツールを使用すると以下の理由でこれを行うことができます。

- **arptables_jf** は ARP が VIP と実サーバーを関連付けないようにします。
- **iptables** メソッドの場合、実サーバー上での VIP 設定はまったく行わないため ARP に関する問題を完全回避することができます。

ダイレクトルーティングによる Load Balancer Add-On 環境で **arptables** や **iptables** を使用する方法については 「ダイレクトルーティングおよび**arptables_jf**」 または 「ダイレクトルーティングと **iptables**」 を参照してください。

1.5. 永続性とファイアウォールマーク

状況によっては、Load Balancer Add-Onの負荷分散アルゴリズムを使って要求を最適なサーバーに送信するのではなく、クライアント側から同じ実サーバーに繰り返し再接続を行わせた方がよい場合もあります。例えば、マルチスクリーンのウェブ申し込みやクッキー、SSL、FTP 接続などが挙げられます。このような場合、コンテキストを保持するため同じサーバーがトランザクションを処理しないと、クライアントが適切に機能しないことがあります。Load Balancer Add-On ではこの状況に対処するため、*永続性* と *ファイアウォールマーク* という 2 つの機能を提供します。

1.5.1. 永続性

永続性は有効にするとタイマーのように動作します。クライアントがサービスに接続すると、Load Balancer Add-On では指定期間の最終接続を記憶します。同じ期間内に同じクライアント IP アドレスが接続を行うと、前回接続したサーバーと同じサーバーに送信されます (負荷分散メカニズムを無視)。指定期間を過ぎてから接続が発生した場合は設定されているスケジューリングルールにしたがって処理されます。

また、永続性を使うと、管理側でサブネットマスクを指定して、どのアドレスがより高い永続性を持つかを管理するツールとして、クライアント IP アドレステストに適用することができます。こうすることで、接続をそのサブネットにグループ化できます。

通信に複数のポートを使用する FTP などのプロトコルの場合、宛先が別々のポートの接続をグループ化することがとても重要な場合があります。ただし、宛先が別々のポートの接続をグループ化する上で発生する問題に対処する場合、永続性は最も効率的な方法とは言えません。このような場合には *ファイアウォールマーク* を使用するのが最適です。

1.5.2. ファイアウォールマーク

プロトコルに使用されている複数のポートのグループ化が行える簡単で効率的な方法がファイアウォールマークです。例えば、インターネット通販の運営に Load Balancer Add-On を導入した場合、ファイアウォールマークを使ってポート 80 上の HTTP 接続とポート 443 上の安全な HTTPS 接続をまとめる

ことができます。各プロトコルの仮想サーバーに同じファイアウォールマークを割り当てると、LVS ルーターは接続開始後すべての要求を同じ実サーバーに転送するため、トランザクションの状態情報を保持することができるようになります。

ファイアウォールマークは効率的なだけでなく使い勝手もよいため、Load Balancer Add-On の管理の際に接続をグループ化する場合はできる限り永続性ではなくファイアウォールマークを使用してください。ただし、クライアントを再接続する場合に一定期間は必ず同じサーバーに接続されるよう設定する場合はファイアウォールと併用して永続性も仮想サーバーに追加する必要があります。

1.6. LOAD BALANCER ADD-ON – ブロックダイアグラム

LVS ルーターは複数プログラムの集合を使用してクラスターメンバーやクラスターサービスを監視します。図1.5「Load Balancer Add-On のコンポーネント」では、各種プログラムがアクティブ LVS ルーターとバックアップ LVS ルーターの両方でクラスターを管理するためどのように動作しているかを示します。

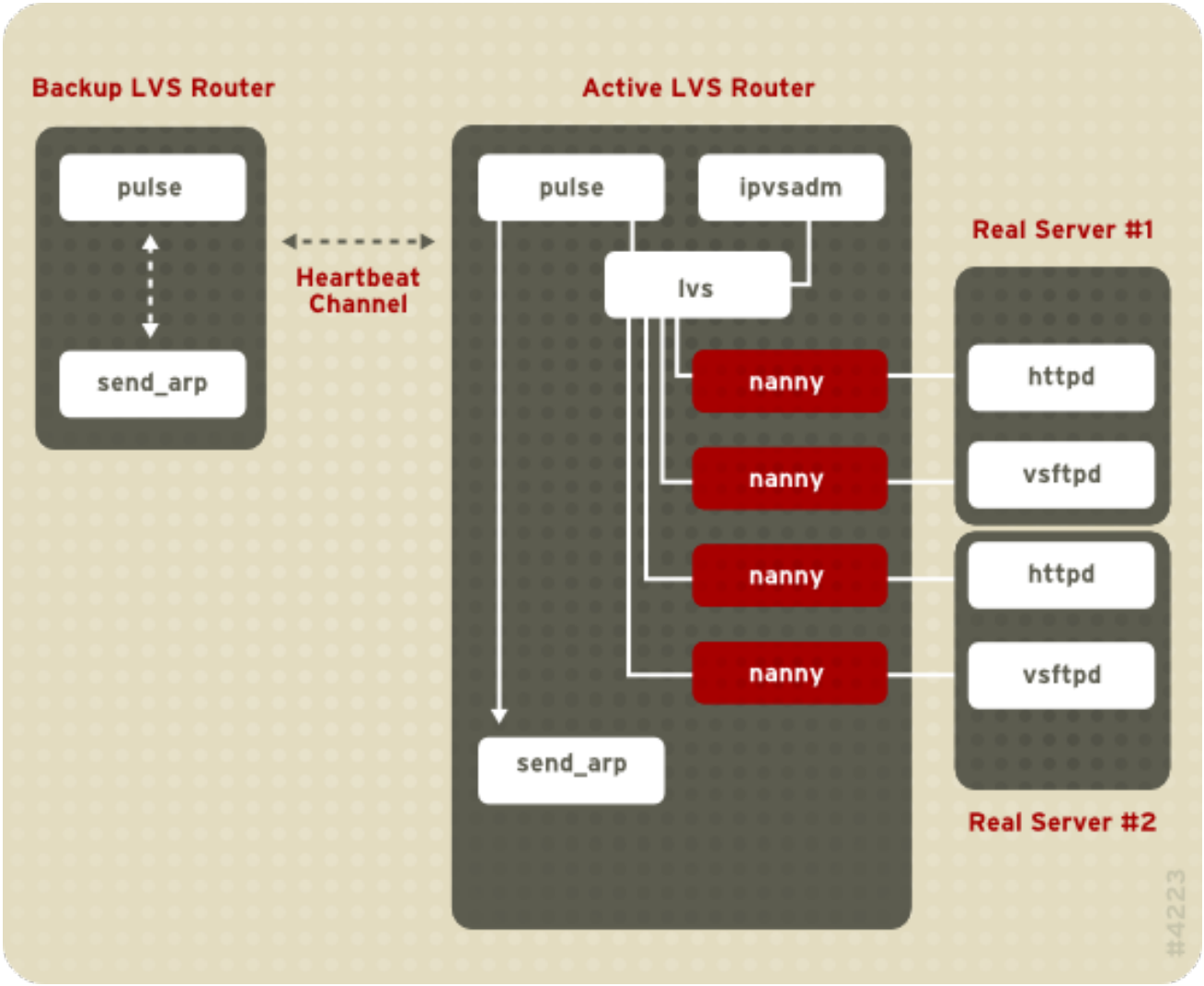


図1.5 Load Balancer Add-On のコンポーネント

pulse デーモンはアクティブ LVS ルーターとバックアップ LVS ルーターの両方で実行させます。バックアップルーター側の **pulse** からアクティブルーターのパブリックインターフェースに対して *ハートビート* が送信され、アクティブルーターが正しく動作しているか確認が行われます。アクティブルーター側の **pulse** は **lvs** デーモンを開始してバックアップルーターからの *ハートビート* クエリーに応答します。

lvs デーモンが開始されると、**ipvsadm** ユーティリティが呼び出されカーネル内で IPVS ルーティン

グテーブルを設定し管理します。また、各実サーバーで設定されている各仮想サーバーに **nanny** プロセスが開始されます。**nanny** の各プロセスは任意の実サーバーで設定されている任意のサービスの状態をチェックし、そのサービスが誤動作していないか **lvs** デーモンに伝えます。誤動作が検出されると、**lvs** デーモンは IPVS ルーティングテーブルからその実サーバーを取り除くよう**ipvsadm** に指示を出します。

バックアップルーター側でアクティブルーターから応答が受信できないと、バックアップルーターは **send_arp** を呼び出してすべての仮想 IP アドレスをバックアップノードの NIC ハードウェアアドレス (*MAC* アドレス) に再割り当てしてフェイルオーバーを開始します。アクティブルーターの**lvs** デーモンをシャットダウンするコマンドがパブリックネットワークインターフェースとプライベートネットワークインターフェースの両方を経由してアクティブルーターに送信され、バックアップノードの **lvs** デーモンが起動、設定されている仮想サーバーの要求の受け取りを開始します。

1.6.1. Load Balancer Add-On のコンポーネント

LVS ルーター内の各ソフトウェアコンポーネントの詳細を 「**pulse**」 に示します。

1.6.1.1. pulse

LVS ルーターに関連する他のすべてのデーモンを開始する制御プロセスです。このデーモンは起動時に **/etc/rc.d/init.d/pulse** スクリプトで起動され、設定ファイル **/etc/sysconfig/ha/lvs.cf** を読み込みます。アクティブルーター側の **pulse** により LVS デーモンが起動されます。バックアップルーター側の **pulse** からユーザー設定が可能な間隔でシンプルなハートビートが実行されアクティブルーター側の健全性が確認されます。ユーザー設定された間隔を過ぎてもアクティブルーター側が応答できない場合、フェイルオーバーが開始されます。フェイルオーバー中、バックアップルーター側の **pulse** からアクティブルーター側の **pulse** デーモンに対してすべての LVS サービスをシャットダウンするよう指示が出され、フローティング IP アドレスをバックアップルーターの MAC アドレスに再割り当てするため **send_arp** プログラムが開始、**lvs** デーモンが起動されます。

1.6.1.2. lvs

lvs デーモンは **pulse** に呼び出されるとアクティブ LVS ルーターで稼働します。設定ファイル **/etc/sysconfig/ha/lvs.cf** を読み込み、**ipvsadm** ユーティリティを呼び出して IPVS ルーティングテーブルを構築し管理を行います。また、設定されている Load Balancer Add-Onの各サービスにそれぞれ **nanny** プロセスを割り当てます。**nanny** により実サーバーがダウンしていることが報告されると、**lvs** より **ipvsadm** ユーティリティに IPVS ルーティングテーブルからその実サーバーを削除するよう指示が出されます。

1.6.1.3. ipvsadm

カーネル内の IPVS ルーティングテーブルの更新を行います。**ipvsadm** を呼び出し IPVS ルーティングテーブル内のエントリーの追加、変更、削除を行うことで **lvs** デーモンは Load Balancer Add-On の設定および管理を行っています。

1.6.1.4. nanny

nanny 監視デーモンはアクティブ LVS ルーター上で稼働します。このデーモンを使ってアクティブルーターは各実サーバーの健全性を確認、オプションでその負荷も監視します。各実サーバーで定義されているサービスごとに個別のプロセスが実行されます。

1.6.1.5. /etc/sysconfig/ha/lvs.cf

Load Balancer Add-On の設定ファイルです。直接または間接的にすべてのデーモンが設定情報をこのファイルから取得することになります。

1.6.1.6. Piranha Configuration Tool

Load Balancer Add-On の監視、設定、管理を行う Web ベースのツールです。Load Balancer Add-On の設定ファイル **/etc/sysconfig/ha/lvs.cf** 管理用のデフォルトツールになります。

1.6.1.7. send_arp

フェイルオーバーの際、フローティング IP アドレスがあるノードから別のノードに変更されるとき、このプログラムにより ARP ブロードキャストが送信されます。

Red Hat Enterprise Linux を LVS ルーターに設定する前に行うべきインストール後の重要な設定手順については *2章Load Balancer Add-On の初期設定*で説明します。

第2章 LOAD BALANCER ADD-ON の初期設定

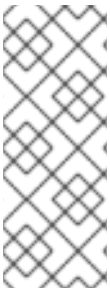
Red Hat Enterprise Linux をインストールしたら、LVS ルーターと実サーバーをセットアップするために基本的なステップを実行する必要があります。本章では、これらのステップを詳述します。

注記

Load Balancer Add-On 開始後にアクティブノードになる LVS ルーターノードは、*プライマリーノード*とも呼ばれます。Load Balancer Add-On の設定時には、プライマリーノードで **Piranha Configuration Tool** を使います。

2.1. LVS ルーターでのサービス設定

Load Balancer Add-On の設定に必要なコンポーネントはすべて Red Hat Enterprise Linux インストールプログラムによってインストールされますが、Load Balancer Add-On を設定する前に適切なサービスをアクティブにしておく必要があります。LVS ルーターに対して起動時に適切なサービスが開始するよう設定します。起動時にサービスをアクティブにするツールが Red Hat Enterprise Linux には 3 種類あります。コマンドラインプログラムの **chkconfig**、ncurses ベースのプログラム **ntsysv**、グラフィカルな **Services Configuration Tool** です。いずれのツールを使用する場合にも root でのアクセスが必要になります。



注記

root アクセスを取得するには、シェルプロンプトを開いて **su -** コマンドを入力した後 root パスワードを入力します。以下に例を示します。

```
$ su -
Password:root password
```

LVS ルーターで以下の 3 つのサービスを起動時にアクティブにする必要があります。

- **piranha-gui** サービス (プライマリーノードのみ)
- **pulse** サービス
- **sshd** サービス

マルチポートサービスをクラスター化しているまたはファイアウォールマークを使用している場合は、**iptables** サービスも有効にする必要があります。

これらのサービスはランレベル 3 およびランレベル 5 の両レベルでアクティブにしておくのが最適です。**chkconfig** を使ってアクティベートするには、以下のコマンドを各サービスに対して実行します。

/sbin/chkconfig --level 35 *daemon* on

上記のコマンドの *daemon* にはアクティブにするサービスの名前を入れてください。システム上のサービス一覧、そのサービスがアクティブにされるランレベルなどを表示する場合は次のコマンドを実行します。

/sbin/chkconfig --list

警告

chkconfig を使用して上記のサービスをオンにしても直ちにデーモンを開始するわけではありません。デーモンを直ちに開始する場合は **/sbin/service** コマンドを使用します。**/sbin/service** コマンドの使用例については「**Piranha Configuration Tool** サービスの開始」を参照してください。

ランレベルおよび **ntsysv** や **Services Configuration Tool** を使ったサービスの設定方法については『Red Hat Enterprise Linux System Administration Guide』の『Controlling Access to Services』の章を参照してください。

2.2. PIRANHA CONFIGURATION TOOL のパスワード設定

Piranha Configuration Tool をプライマリー LVS ルーターで最初に使用する前に、パスワードを作成してこのツールへのアクセスを制限する必要があります。これを行うには、**root** でログインして以下のコマンドを実行します。

```
/usr/sbin/piranha-passwd
```

このコマンドの入力後にプロンプトが表示されたら、管理用のパスワードを作成します。

警告

パスワードの安全性を確保するため、固有名詞や一般的に使用されている略語、言語にかかわらず辞書に載っている単語などは含めないようにしてください。システム上でパスワードを暗号化しないまま放置しないようにしてください。

アクティブな **Piranha Configuration Tool** セッション中にパスワードが変更された場合は、管理者は新たなパスワードを提供するようにプロンプト表示されます。

2.3. PIRANHA CONFIGURATION TOOL サービスの開始

Piranha Configuration Tool のパスワード設定後に、**/etc/rc.d/init.d/piranha-gui** にある **piranha-gui** サービスを開始、または再起動します。これには、以下のコマンドを **root** で入力します。

```
/sbin/service piranha-gui start
```

または

```
/sbin/service piranha-gui restart
```

このコマンドを実行すると、**/usr/sbin/piranha_gui -> /usr/sbin/httpd** のシンボリックリンクが呼び出され Apache HTTP Server のプライベートセッションが開始されます。安全上、**httpd**

の **piranha-gui** バージョンは piranha ユーザーとして別プロセスで実行されます。実際には**httpd** サービスは **piranha-gui** によって開始されるため以下の点に注意してください。

1. Apache HTTP Server をシステムにインストールしておく必要があります。
2. Apache HTTP Server を **service** コマンドを使って停止、または再開すると**piranha-gui** サービスが停止します。

警告

LVS ルーターで **/sbin/service httpd stop** もしくは **/sbin/service httpd restart** が実行した場合は、**piranha-gui** サービスを以下のコマンドで開始する必要があります。

```
/sbin/service piranha-gui start
```

Load Balancer Add-On の設定を開始するのに必要なサービスは **piranha-gui** サービスのみです。ただし、Load Balancer Add-On を遠隔から設定する場合は、**sshd** サービスも必要になります。**Piranha Configuration Tool** を使った設定が完了するまでは、**pulse** サービスを開始する必要は**ありません**。**pulse** サービスの開始については「Load Balancer Add-Onを開始する」 を参照してください。

2.3.1. Piranha Configuration Tool Web サーバーポートの設定

Piranha Configuration Tool はデフォルトではポート **3636** で稼働します。このポート番号を変更する場合は、**piranha-gui** Web サーバー設定ファイル **/etc/sysconfig/ha/conf/httpd.conf** のセクション 2 内にある **Listen 3636** の行を変更します。

Piranha Configuration Tool を使用するには、最低でもテキストベースの Web ブラウザが必要になります。プライマリー LVS ルーター上で Web ブラウザを開始する場合は、ロケーション **http://*localhost*:3636** を開きます。*localhost* の部分をプライマリー LVS ルーターのホスト名または IP アドレスに置き換えると、Web ブラウザでどこからでも **Piranha Configuration Tool** にアクセスできます。

ブラウザで **Piranha Configuration Tool** に接続したら、設定サービスにアクセスするためログインしなければなりません。**ユーザー名 (Username)** フィールドに **piranha**、**パスワード (Password)** フィールドに **piranha-passwd** で設定したパスワードをそれぞれ入力します。

これで **Piranha Configuration Tool** が稼働し始めます。ネットワーク経由でこのツールにアクセスできるメンバーを制限したい場合があります。次のセクションではアクセスの制限を実施する方法を見ていきます。

2.4. PIRANHA CONFIGURATION TOOL へのアクセス制限

Piranha Configuration Tool では、有効なユーザー名とパスワードの組み合わせが要求されます。しかし、**Piranha Configuration Tool** に送られるデータはすべてプレーンテキストなので、アクセスを信頼できるネットワークまたはローカルマシンに限定することが推奨されます。

アクセスを制限する最も簡単な方法は **/etc/sysconfig/ha/web/secure/.htaccess** を編集して Apache HTTP Server に組み込まれているアクセス制御メカニズムを利用する方法です。このファイル

を変更した後は、サーバーがディレクトリにアクセスする度 **.htaccess** ファイルをチェックするため、**piranha-gui** サービスを再起動する必要はありません。

デフォルトでは、このディレクトリのアクセス制御は誰にでも制限なくディレクトリコンテンツの読み取りを許可しています。デフォルトのアクセスを以下に示します。

```
Order deny,allow
Allow from all
```

Piranha Configuration Tool へのアクセスをローカルホストのみに制限するには、**.htaccess** ファイルを変更してループバックデバイス (127.0.0.1) からのアクセスのみに制限します。ループバックデバイスについての詳細は『Red Hat Enterprise Linux Reference Guide』 の 『Network Scripts』 の章を参照してください。

```
Order deny,allow
Deny from all
Allow from 127.0.0.1
```

以下の例のように、特定のホストやサブネットを許可することもできます。

```
Order deny,allow
Deny from all
Allow from 192.168.1.100
Allow from 172.16.57
```

この例では、IP アドレス 192.168.1.100 のマシンと、172.16.57/24 ネットワーク上の マシンの Web ブラウザのみが **Piranha Configuration Tool** にアクセスできます。

警告

/etc/sysconfig/ha/web/secure/ ディレクトリ内の設定ページへのアクセスについては **Piranha Configuration Tool** の **.htaccess** ファイルを編集すると行うことができますが、**/etc/sysconfig/ha/web/** ディレクトリ配下のログインおよびヘルプページへのアクセスは制限できません。このディレクトリへのアクセスを制限する場合は、**/etc/sysconfig/ha/web/secure/.htaccess** と同じ **order**、**allow**、**deny** の行を持たせた **.htaccess** ファイルを **/etc/sysconfig/ha/web/** ディレクトリ内に作成してください。

2.5. パケット転送をオンにする

LVS ルーターが実サーバーに正確にネットワークパケットを転送するためには、カーネル内で各 LVS ルーターノードの IP 転送をオンにしておく必要があります。root でログインして **/etc/sysctl.conf** 内の **net.ipv4.ip_forward = 0** の行を以下のように変更します。

```
net.ipv4.ip_forward = 1
```

システムを再起動すると変更が反映されます。

IP 転送がオンになっているか確認するため root で次のコマンドを実行します。

```
/sbin/sysctl net.ipv4.ip_forward
```

上記のコマンドで **1** が返される場合は IP 転送が有効になっています。**0** が返される場合は以下のコマンドを使って手作業でオンにすることができます。

```
/sbin/sysctl -w net.ipv4.ip_forward=1
```

2.6. 実サーバーでサービスを設定する

実サーバーが Red Hat Enterprise Linux システムの場合は、起動時に適切なサーバーデーモンがアクティブになるよう設定しておきます。Web サービスの **httpd** や FTP サービスまた Telnet サービスの **xinetd** などのデーモンがこれに該当します。

また、実サーバーに遠隔からもアクセスできると便利なため **sshd** デーモンもインストールして実行しておいてください。

第3章 LOAD BALANCER ADD-ON の設定

Load Balancer Add-On は LVS ルーターの集合と実サーバーの集合という二つの基本グループで構成されています。単一点障害を防止するため、それぞれのグループには少なくとも二つのメンバーシステムを含める必要があります。

LVS ルーターグループは、Red Hat Enterprise Linux を実行している同一または非常に似ている二つのシステムで構成する必要があります。そのうちの 1 つはアクティブ LVS ルーターとして機能し、もう 1 つはホットスタンバイモードで待機するので、この 2 つができるだけ同じキャパシティを備えている必要があります。

実サーバーグループのハードウェアの選択や設定を行う前に、まず3 種類の Load Balancer Add-On トポロジーのうちどれを使用するかを決定します。

3.1. NAT を使った LOAD BALANCER ADD-ON ネットワーク

NAT トポロジーを使用すると、既存ハードウェアの活用の自由度が高まりますが、大量の負荷を処理するには限界があります。これは、プールを出入りするパケットがすべて Load Balancer Add-On ルーターを通過するためです。

ネットワークレイアウト

NAT ルーティングを使った Load Balancer Add-On のトポロジーはパブリックネットワークへのアクセスポイントが 1 つあれば構成できるため、ネットワークレイアウトの観点からは最も設定が簡単なトポロジーになります。実サーバーはすべての要求を LVS ルーター経由で返すため、すべての実サーバーがそれ専用のプライベートネットワーク上に配置されることになります。

ハードウェア

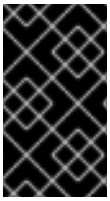
正常に機能させるため実サーバーを Linux マシンにする必要はないため、ハードウェアに関しては NAT トポロジーが最も柔軟なトポロジーになります。各実サーバーが応答するのは LVS ルーターのみのため、実サーバー側に必要な NIC は 1 つのみになります。一方、LVS ルーターでは 2 種類のネットワークのトラフィックを別々にルーティングさせるため NIC が 2 つ必要になります。このトポロジーの場合、LVS ルーターの部分がネットワークのボトルネックになるため、各 LVS ルーターにギガビットイーサネットの NIC を使用し LVS ルーターで処理できる帯域幅を増大させることが可能です。LVS ルーターにギガビットイーサネットを使用する場合は、負荷を効率的に処理するため実サーバーを LVS ルーターに接続しているスイッチについてもギガビットイーサネットポートが少なくとも 2 つ搭載されているスイッチが必要になります。

ソフトウェア

NAT トポロジーには一部の設定で **`iptables`** を使用する必要があるため、**Piranha Configuration Tool** 以外にも設定を必要とするソフトウェアがあります。特に FTP サービスとファイアウォールマークの場合、LVS ルーターで要求を正しくルーティングできるよう手作業による設定を必要とします。

3.1.1. NAT を使って Load Balancer Add-On のネットワークインターフェースを設定する

NAT を使って Load Balancer Add-On を設定する場合、まず LVS ルーター上にパブリックネットワーク用とプライベートネットワーク用のネットワークインターフェースを設定しなければなりません。以下の例では、LVS ルーターのパブリックインターフェース (**`eth0`**) は 192.168.26/24 ネットワーク (ルーティング可能な IP ではないが LVS ルーターの前にファイアウォールがあると仮定)、実サーバーにつながっているプライベートインターフェース (**`eth1`**) は 10.11.12/24 ネットワークになります。



重要

記載されている手順で編集を行ったファイルは **network** サービスでは使用されますが、**NetworkManager** サービスでは使用されません。Load Balancer Add-on には **NetworkManager** サービス との互換性はありません。

アクティブ (*プライマリー*) LVS ルーターノードのパブリックインターフェースのネットワークスクリプト **/etc/sysconfig/network-scripts/ifcfg-eth0** の例を以下に示します。

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.26.9
NETMASK=255.255.255.0
GATEWAY=192.168.26.254
```

LVS ルーターのプライベート NAT インターフェースのスクリプト **/etc/sysconfig/network-scripts/ifcfg-eth1** の例を以下に示します。

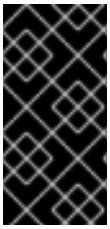
```
DEVICE=eth1
BOOTPROTO=static
ONBOOT=yes
IPADDR=10.11.12.9
NETMASK=255.255.255.0
```

この例では、LVS ルーターのパブリックインターフェースの VIP は 192.168.26.10、NAT (プライベート) インターフェースの VIP は 10.11.12.10 になります。したがって、実サーバーが要求を返すのは NAT インターフェースの VIP になると言う点に注意してください。

重要

本セクションのイーサネットインターフェースの設定例は、LVS ルーターの実 IP アドレス用であり、フローティング IP アドレス用では**ありません**。パブリックおよびプライベートのフローティング IP アドレスを設定する場合は、「**グローバル設定 (GLOBAL SETTINGS)**」 および 「**VIRTUAL SERVER** サブセクション」 で説明しているように **Piranha Configuration Tool** を使用してください。

アクティブ LVS ルーターノードのネットワークインターフェースを設定したら、バックアップ LVS ルーターの実ネットワークインターフェースの設定を行います。IP アドレスがネットワーク上の他の IP アドレスと競合しないよう注意してください。



重要

バックアップノード上の各インターフェースがアクティブノード上のインターフェースと同じネットワークに接続するようにしてください。例えば、アクティブノードで eth0 がパブリックネットワークに接続されている場合には、バックアップノードでも eth0 をパブリックネットワークに接続してください。

3.1.2. 実サーバー上でのルーティング

NAT トポロジーで実サーバーのネットワークインターフェースを設定する場合、忘れてはいけないもっとも重要な作業が LVS ルーターの NAT フローティング IP アドレス用にゲートウェイを設定することです。この例では、ゲートウェイは 10.11.12.10 になります。

注記

実サーバー上でのネットワークインターフェース設定が完了すると、マシンは他の方法でパブリックネットワークに ping したり接続したりすることができなくなります。これは正常なことです。しかし、LVS ルーターのプライベートインターフェースの実 IP、この場合は 10.11.12.9、に ping することはできます。

実サーバーの **/etc/sysconfig/network-scripts/ifcfg-eth0** ファイルは以下のようになります。

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=10.11.12.1
NETMASK=255.255.255.0
GATEWAY=10.11.12.10
```

警告

実サーバーに **GATEWAY=** の行で設定された複数のネットワークインターフェースがある場合、最初に記載されている設定がゲートウェイ用となります。このため、**eth0** と **eth1** いずれも設定されていて **eth1** が Load Balancer Add-On 用に使用されている場合、実サーバーは要求を正しくルーティングできない可能性があります。

関係ないネットワークインターフェースはオフにするのが最適です。**/etc/sysconfig/network-scripts/** ディレクトリ内の各ネットワークスクリプトで **ONBOOT=no** とセットするか、または 1 番目に記載されているインターフェースでゲートウェイが正しく設定されているか確認します。

3.1.3. LVS ルーターで NAT ルーティングを有効にする

クラスター化したサービスが 1 つのポートしか使用しない、たとえば HTTP に使用させるのはポート 80、などのシンプルな NAT を使った Load Balancer Add-On 構成の場合は、 LVS ルーターでパケット転送を有効にするだけで外の世界と実サーバー間の要求を正しくルーティングすることができます。パケット転送の調整方法については「パケット転送をオンにする」の説明をご覧ください。ただし、ユーザーセッション中は同じ実サーバーに戻れるようクラスター化したサービスに複数のポートを必要とする場合はさらに設定が必要となります。ファイアウォールマークを使って複数ポートのサービスを作成する方法については「マルチポートサービスと Load Balancer Add-On」を参照してください。

LVS ルーターでパケット転送を有効にし、実サーバーをセットアップしてクラスタ化したサービスが稼働するようになったら、*4章Piranha Configuration Tool を使った Load Balancer Add-Onの設定*で説明しているように **Piranha Configuration Tool** を使って Load Balancer Add-On を設定します。

警告

eth0:1 または**eth1:1** のフローティング IP の設定を行う場合は必ず **Piranha Configuration Tool** を使用してください。使い方については「**グローバル設定 (GLOBAL SETTINGS)**」および「**VIRTUAL SERVER** サブセクション」の説明をご覧ください。ネットワークスクリプトを手作業で編集する、ネットワーク設定ツールを使うなどの方法は使用しないでください。

設定後は、「Load Balancer Add-Onを開始する」の説明にしたがい **pulse** サービスを開始します。**pulse** が稼働し始めると、アクティブ LVS ルーターにより実サーバープールへの要求のルーティングが開始されます。

3.2. ダイレクトルーティングを使った LOAD BALANCER ADD-ON

「ダイレクトルーティング」で記載しているように、ダイレクトルーティングでは実サーバーがパケットを処理した後、発信パケットを LVS ルーター経由でルーティングするのではなく、実サーバーが直接、要求元のユーザーにルーティングすることができます。ダイレクトルーティングの場合、実サーバーが LVS ルーターのあるネットワークセグメントに物理的に接続され、発信パケットの処理とルーティングができなければなりません。

ネットワークレイアウト

ダイレクトルーティングを使った Load Balancer Add-On では、LVS ルーターが着信要求を受け取り、それらを処理のために適切な実サーバーにルーティングする必要があります。次に実サーバーがその応答を **直接**、クライアントにルーティングする必要があります。例えば、クライアントがインターネット上に存在し、パケットを LVS ルーター経由で実サーバーに送信した場合、実サーバーはインターネット経由で直接クライアントに到達できる必要があります。実サーバーにゲートウェイを設定しパケットをインターネットに渡すことでこれを実現します。サーバープール内の各実サーバーには別々のゲートウェイ (また各ゲートウェイにはインターネットへの接続がある) を持たせることができるため、最大限のスループットとスケーラビリティを実現できます。ただし、標準的な Load Balancer Add-On 設定の場合、複数の実サーバーは 1 つの ゲートウェイで通信することができます (したがってネットワーク接続も 1 つ)。

重要

LVS ルーターを実サーバーのゲートウェイとして使用することは **推奨されません**。そのような使用方法では、LVS ルーターに不要な設定の複雑性とネットワーク負荷を追加することになり、NAT ルーティングにみられるネットワークのボトルネックを戻すことになってしまいます。

ハードウェア

ダイレクトルーティング使用の Load Balancer Add-On システムでのハードウェア要件は、他の Load Balancer Add-On トポロジーと同様のものです。LVS ルーターが着信要求を処理して実サーバー用にロードバランシングを実行するには Red Hat Enterprise Linux の稼働を必要としますが、実サーバーが正常に機能するには Linux マシンである必要はありません。LVS ルーターはそれぞれ 1 つまたは 2 つの NIC を必要とします (バックアップルーターの有無による)。設定を容易にしてトラフィックを明確に分けるために、2 つの NIC を使用することもできます。こうすると、1 つの NIC で着信要求を処理し、実サーバーへのパケット回送はもうひとつの NIC に任せることができます。

実サーバーは LVS ルーターを迂回して送信パケットを直接クライアントに送信するため、インターネットへのゲートウェイが必要となります。パフォーマンスと可用性を最大化するには、クライアントが接続しているキャリアネットワーク (インターネットやイントラネットなど) に専用接続がある独自のゲートウェイに実サーバーを接続します。

ソフトウェア

ダイレクトルーティングで Load Balancer Add-On を使用していて ARP 問題に直面している管理者には特に、 **Piranha Configuration Tool** の範囲外で必要な設定があります。 詳細情報は、「ダイレクトルーティングおよび **arptables_jf**」 または 「ダイレクトルーティングと **iptables**」 を参照してください。

3.2.1. ダイレクトルーティングおよび**arptables_jf**

arptables_jf を使用してダイレクトルーティングを設定するには、実サーバーでそれらの仮想 IP アドレスが設定されており、パケットが直接送信で可能となっている必要があります。VIP 用の ARP 要求は実サーバーでは完全に無視されます。そ例外の、VIP を含んでいて送信される ARP パケットは、mangle 化されて VIP ではなく実サーバーの IP が含まれるようになります。

arptables_jf メソッドを使用すると、アプリケーションは実サーバーが接続している個別の VIP またはポートにバインドします。例えば、**arptables_jf** メソッドの使用により、Apache HTTP Server の複数のインスタンスはシステム上の異なる VIP に明示的にバインドして実行することが可能になります。また、**arptables_jf** の使用は、**iptables** オプションよりもパフォーマンスで大きな利点があります。

しかし、**arptables_jf** メソッドを使うと、標準の Red Hat Enterprise Linux システム設定ツールを使用して起動時に VIP を開始する設定ができません。

それぞれの仮想 IP アドレスの ARP 要求を無視するように実サーバーを設定するには、以下の手順を実行します。

1. 実サーバー上で仮想 IP アドレス用に ARP テーブルのエントリを作成します (real_ip とは実サーバーとの通信にディレクタが使用する IP のこと。多くの場合、**eth0** にバインドされた IP)。

   ```
   arptables -A IN -d <virtual_ip> -j DROP
   arptables -A OUT -s <virtual_ip> -j mangle --mangle-ip-s <real_ip>
   ```

   これにより、仮想 IP アドレス向けのすべての ARP 要求を実サーバーが無視するようになります。また、他の方法では仮想 IP を含むことになる送信 ARP 反応を変更させて、それらがサーバーの実 IP を含むようになります。VIP の ARP 要求に反応する唯一のノードは、現在アクティブな LVS ノードです。

2. これが実サーバー上で完了したら、実サーバー上で以下のコマンドを入力して ARP テーブルのエントリを保存します。

   ```
   service arptables_jf save

   chkconfig --level 2345 arptables_jf on
   ```

   chkconfig コマンドは、ネットワーク開始前にシステムが起動時に arptables 設定をリロードするようにします。

3. **ifconfig** を使用して実サーバー上で仮想 IP アドレスを設定し、 IP エイリアスを作成します。例えば

```
# ifconfig eth0:1 192.168.76.24 netmask 255.255.252.0 broadcast
192.168.79.255 up
```

または **iproute2** のユーティリティ **ip** を使用します。 例えば

```
# ip addr add 192.168.76.24 dev eth0
```

ここまでの記述にあるように、仮想 IP アドレスは Red Hat システム設定ツールを使用して起動時に開始するように設定することはできません。この問題を回避する方法の 1 つは **/etc/rc.d/rc.local** 内にこれらのコマンドを配置することです。

4. ダイレクトルーティング用に Piranha を設定します。詳細情報は *4章Piranha Configuration Tool を使った Load Balancer Add-Onの設定*を参照してください。

3.2.2. ダイレクトルーティングと iptables

iptables ファイアウォールルールを作成することで、 ダイレクトルーティングメソッドを使用した場合の ARP 問題を回避することもできます。**iptables** を使用してダイレクトルーティングを設定するには、VIP アドレスがシステム上に存在しなくても VIP アドレスに送信されたパケットを実サーバーが扱うように透過プロキシを作成するルールを追加する必要があります。

iptables メソッドは **arptables_jf** メソッドよりも 設定が簡単です。仮想 IP アドレスがアクティブ LVS ディレクタ上にのみ存在するため、このメソッドでは LVS ARP 問題も完全に回避できます。

しかし、パケットの転送/マスカレードでのオーバーヘッドがあるため、**arptables_jf** と比較すると、**iptables** メソッドの使用にはパフォーマンスの問題があります。

また、**iptables** メソッドを使用してポートを再利用することはできません。例えば、二つの別々の Apache HTTP Server サービスは 両方とも仮想 IP アドレスではなく ***INADDR_ANY*** にバインドする必要があるため、ポート 80 にバインドされた二つの別々の Apache HTTP Server サービスを実行することはできません。

iptables メソッドを使用してダイレクトルーティングを設定するには、以下の手順を実行します。

1. 実サーバーでの実行が意図された VIP、ポート、プロトコル (TCP または UDP)の組み合わせすべてに、以下のコマンドを実行します。

   iptables -t nat -A PREROUTING -p <tcp|udp> -d <vip> --dport <port> -j REDIRECT

   このコマンドで、実サーバーは与えられた VIP とポートが宛先となっているパケットを処理します。

2. 実サーバー上で設定を保存します。

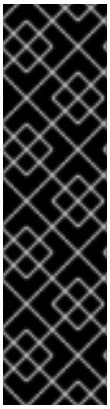
   ```
   # service iptables save
   # chkconfig --level 2345 iptables on
   ```

   上記のコマンドで、システムはネットワーク開始前に、起動時に **iptables** 設定をリロードします。

3.3. 設定を組み合わせる

上記のルーティングメソッドでどれを使用するか決定した後、ハードウェアをネットワーク上でリン

クさせます。



重要

LVS ルーター上のアダプタデバイスは、同じネットワークにアクセスするように設定する必要があります。例えば、**eth0** がパブリックネットワークに接続し、**eth1** がプライベートネットワークに接続する場合、バックアップ LVS ルーター上の同じデバイスは同じネットワークに接続する必要があります。

また、起動時に最初に出現するインターフェースにリストされているゲートウェイは、ルーティングテーブルに追加され、他のインターフェースにリストされているそれ以降のゲート ウェイは無視されます。これは、実サーバーを設定する場合に特に考慮すべき重要点です。

ハードウェアを物理的に接続したら、プライマリー LVS ルーターとバックアップ LVS ルーター上でネットワークインターフェースを設定します。これは **system-config-network** などのグラフィカルアプリケーションを使用するか、手動でネットワークスクリプトを編集することで可能です。**system-config-network** を使用したデバイス追加の詳細情報は、『Red Hat Enterprise Linux 導入ガイド』 内にある 『ネットワーク設定』 を参照してください。この章の残りの部分では、手動もしくは**Piranha Configuration Tool** によるネットワークインターフェースへの変更例を説明します。

3.3.1. Load Balancer Add-Onネットワーキングの一般的なヒント

Piranha Configuration Tool を使用して Load Balancer Add-On の設定する前に、LVS ルーター上でパブリックネットワークとプライベートネットワークの両方用に実 IP アドレスを設定します。各トポロジーのセクションではサンプルのネットワークアドレスが使われていますが、実際には本物のネットワークアドレスが必要になります。以下にネットワークインターフェースの起動とそれらのステータスチェックに役に立つコマンドを挙げます。

実ネットワーキングインターフェースの起動

実ネットワークインターフェースを起動するには、root で以下のコマンドを実行して、 *N* の部分をインターフェースに相当する番号で置き換えます (**eth0** および **eth1**)。

```
/sbin/ifup ethN
```

警告

Piranha Configuration Tool を使用して設定するフローティング IP アドレス (**eth0:1** や **eth1:1**) を起動する際に、**ifup** スクリプトを使用**しないで下さい**。代わりに **service** コマンドを 使用して **pulse** を開始してください (詳細は 「Load Balancer Add-Onを開始する」 参照してください)。

実ネットワークインターフェースの停止

実ネットワークインターフェースを停止するには、root で以下のコマンドを実行して、 *N* の部分をインターフェースに相当する番号で置き換えます (**eth0** および **eth1**)。

```
/sbin/ifdown ethN
```

ネットワークインターフェースのステータスチェック

ある時点でどのネットワークインターフェースが起動しているかをチェックするには、以下を入力します。

```
/sbin/ifconfig
```

マシンのルーティングテーブルを表示するには、以下のコマンドを実行します。

```
/sbin/route
```

3.3.1.1. 仮想 IP アドレス問題のトラブルシューティング

アクティブ LVS ホストからスタンバイホストへの自動フェイルオーバー中に、管理者が問題に直面する場合があります。フェイルオーバーの際にスタンバイホストではすべての仮想 IP アドレスがアクティベートしない場合があります。この問題は、スタンバイホストが停止されプライマリホストがアクティベートされる際にも発生する可能性があります。仮想 IP アドレスがすべてアクティベートするのは、**pulse** サービスが手動で再起動される時のみです。

この問題を一時的に軽減するには、以下のコマンドを root のシェルプロンプトで実行します。

```
echo 1 > /proc/sys/net/ipv4/conf/all/promote_secondaries
```

このコマンドは問題を **一時的に** 軽減するのみで、システムを再起動するとコマンドが維持されないことに注意してください。

この問題を永続的に軽減するには、**/etc/sysctl.conf** ファイルを開いて以下の行を追加します。

```
net.ipv4.conf.all.promote_secondaries = 1
```

3.4. マルチポートサービスと LOAD BALANCER ADD-ON

LVS ルーターはいかなるトポロジーでも、マルチポートの Load Balancer Add-On サービスを作成する場合は、追加の設定が必要になります。HTTP (ポート 80) および HTTPS (ポート 443) などのように異なるものの、関連するプロトコルをバンドル化するファイアウォールマークを使用することで、または Load Balancer Add-Onが FTP などの真のマルチポートプロトコルで使用される際に、マルチポートサービスは人為的に作成することができます。いずれのケースでも、送信先が異なるポートでも同じファイアウォール マークを付けていて同様に処理されるべきパケットを認識するために、LVS ルーターはファイアウォールマークを使用します。また、永続性と合わせると、ファイアウォールマークは接続が永続性パラメーターで指定されている時間内に発生する限り、クライアントマシンからの接続が同じホストに送信されることを確実にします。永続性を仮想サーバーに割り当てる方法の詳細情報については 「**VIRTUAL SERVER** サブセクション」 をご覧下さい。

残念ながら、実サーバー上で負荷バランスを取るために使用するメカニズムである IPVS は、パケットに割り当てられたファイアウォールマークは認識できますが、ファイアウォールマークを割り当てることはできません。ファイアウォールマークを **割り当てる** 作業はネットワークパケットフィルタである **iptables** で **Piranha Configuration Tool** の外部で実行される必要があります。

3.4.1. ファイアウォールマークの割り当て

送信先が特定ポートとなっているパケットにファイアウォールマークを割り当てるには、管理者は **iptables** を使用する必要があります。

このセクションでは、例として HTTP と HTTPS のバンドル方法を説明します。ただし、FTP も一般的に使用されるクラスタ化されたマルチポートプロトコルです。Load Balancer Add-On が FTP サービスに使用される場合には、設定方法の詳細について「FTP の設定」 を参照してください。

ファイアウォールマークを使用する際の基本的ルールは、**Piranha Configuration Tool** でファイアウォールマークを使用しているプロトコルすべてに、ネットワークパケットにマークを割り当てるため同数の **iptables** ルールがなくてはならない、という点です。

ネットワークパケットのフィルタルールを作成する前に、既に他のルールが存在しないか確認します。これを行うには、シェルプロンプトを開いて、root でログインして以下を入力します。

```
/sbin/service iptables status
```

iptables が実行されていない場合、すぐにプロンプトが再出現します。

iptables がアクティブな場合、ルールセットが表示されます。ルールが存在する場合、以下のコマンドを入力します。

```
/sbin/service iptables stop
```

既存のルールが重要な場合、**/etc/sysconfig/iptables** の内容を確認して、保存する価値のあるルールを安全な場所にコピーしてから継続します。

以下のルールでは、同一ファイアウォールマーク 80 を、送信先がポート 80 と 443 上のフローティング IP アドレス *n.n.n.n* になっている着信トラフィックに割り当てます。

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 -m multiport -
-dports 80,443 -j MARK --set-mark 80
```

VIP をパブリックネットワークインターフェースに割り当てる際の指示については「**VIRTUAL SERVER** サブセクション」 を参照してください。また、初めてルールを発行する前には、root でログインしてから **iptables** 用のモジュールを読み込む必要があることに注意してください。

上述の **iptables** コマンドの *n.n.n.n* は、使用中の HTTP および HTTPS 仮想サーバーのフローティング IP で置き換える必要があります。これらのコマンドは、該当するポート上の VIP が送信先となっている全トラフィックをファイアウォールマーク 80 に割り当てることと同様の効果があります。これが IPVS に認識され、適切に転送されます。

警告

上述のコマンドはすぐに効果を発揮しますが、システムを再起動すると維持されません。ネットワークパケットのフィルタ設定が再起動後に復元するようにするには「ネットワークパケットフィルター設定の保存」 を参照してください。

3.5. FTP の設定

ファイル転送プロトコル (FTP) は旧式で複雑なマルチポートプロトコルで、Load Balancer Add-On 環境では明らかな課題をもたらします。これら課題の性質を理解するには、まず FTP の動作について基本事項を理解する必要があります。

3.5.1. FTP の動作

ほとんどのサーバー/クライアント関係では、クライアントマシンが特定のポート上でサーバーへ接続を開いて、サーバーがそのポートのクライアントに応答します。FTP クライアントが FTP サーバーに接続する場合、FTP 制御ポート 21 への接続を開きます。そして、その**クライアント**が FTP **サーバー**に*アクティブ*か *パッシブ*のどちらの接続を開くかを指示します。クライアントが選択した接続タイプにより、サーバーの対応方法とトランザクションが発生するポートを決定します。

データ接続は以下の 2 種類です。

アクティブ接続

アクティブ接続が確立されると、**サーバー**はポート 20 からクライアントマシン上の高い範囲のポートにクライアントへデータ接続を開きます。サーバーからのすべてのデータは、この接続を通じて送信されます。

パッシブ接続

パッシブ接続が確立されると、**クライアント**は FTP サーバーに対してパッシブ接続ポートを確立するように依頼します。これは 10,000 より高いポートになります。するとサーバーは、この特定のセッション用に高い数値のポートをバインドして、このポート番号をクライアントに中継します。クライアントは、データ接続のために新規にバインドされたポートを開きます。クライアントが作成するデータ要求それぞれ、別個のデータ接続となります。最近の FTP クライアントのほとんどは、 サーバーからデータを要求する場合、パッシブ接続を試みます。

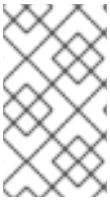
注記

接続タイプを決定するのは、サーバーではなく **クライアント** です。つまり、効果的に FTP をクラスタ化するには、アクティブ接続とパッシブ接続の両方を処理するように LVS ルーターを設定する必要があることになります。

FTP のクライアント/サーバーの組み合わせは、**Piranha Configuration Tool** と IPVS が認識していない多くのポートを開く可能性があります。

3.5.2. Load Balancer Add-On への影響

IPVS パケット転送は、それをベースにしたクラスタのへの接続とそのクラスタからの接続のみを許可し、そのポート番号やファイアウォールマークを認識します。クラスタ外のクライアントが IPVS で処理するように設定されていないポートを開こうとした場合、接続は切断されます。同様に、実サーバーが IPVS が認識できないポート上でインターネット接続を開こうとした場合も、接続は切断されます。つまり、インターネット上の FTP クライアントからの**すべての**接続は、それらに割り当てられているファイアウォールマークと同じである**必要があり**、FTP サーバーからの全接続は、ネットワークパケットのフィルタリングルールを使用して正常にインターネットに転送される**必要がある**ことを意味します。



注記

パッシブ FTP 接続を有効にするには、**`ip_vs_ftp`** カーネルモジュールがロードされていることを確認します。これは、シェルプロンプトで管理ユーザーとして **`modprobe ip_vs_ftp`** コマンドを実行することで可能です。

3.5.3. ネットワークパケットフィルタルールの作成

FTP サービスの **iptables** ルールを割り当てる前に、マルチポートサービスおよび既存ネットワークパケットフィルタリングルールをチェックする技術に関して「ファイアウォールマークの割り当て」内の情報を再確認してください。

以下に示すのは、FTP トラフィックに同一ファイアウォールマークの 21 を割り当てるルールです。これらのルールが正しく機能するには、**Piranha Configuration Tool** の **仮想サーバー** サブセクションを使用して **ファイアーマーク** フィールド内に値 **21** を記入してポート 21 の仮想サーバーを設定する必要があります。詳細は「**VIRTUAL SERVER** サブセクション」を参照してください。

3.5.3.1. アクティブ接続のルール

アクティブ接続のルールは、カーネルに FTP データポートであるポート 20 上にある **内部** のフローティング IP アドレスへ届く接続を受け付けて転送するように指示します。

以下の **iptables** コマンドにより、LVS ルーターは IPVS が認識していない実サーバーからの外向けの接続を受け付けることが可能になります。

```
/sbin/iptables -t nat -A POSTROUTING -p tcp -s n.n.n.0/24 --sport 20 -j
MASQUERADE
```

この **iptables** コマンドでは、*n.n.n* は **Piranha Configuration Tool** の **グローバル設定** 内で定義されている NAT インターフェースの内部ネットワークインターフェース用のフローティング IP の最初の三つの値で置き換える必要があります。

3.5.3.2. パッシブ接続のルール

パッシブ接続のルールでは、10,000 から 20,000 という広い範囲のポートにあるサービスのフローティング IP へのインターネットからの接続に適切なファイアウォールマークを割り当てます。

警告

パッシブ接続でのポート範囲を制限している場合、VSFTP サーバーを設定して一致するポート範囲を使用するように設定する必要があります。これは以下の行を **/etc/vsftpd.conf** に追加することで可能です。

```
pasv_min_port=10000

pasv_max_port=20000
```

実際の FTP サーバーアドレスを上書きする **pasv_address** の設定は使用しないでください。LVS により仮想 IP アドレスに更新されるためです。

他の FTP サーバーの設定については、個別のドキュメンテーションを参照してください。

この範囲はほとんどの状況では十分なものです。しかし、以下のコマンド内の **10000:20000** を **1024:65535** に変更することで、利用可能な非保護ポートすべてを含めることができます。

以下の **iptables** コマンドは、適切なポート上のフローティング IP が送信先であるトラフィックをファイアウォールマーク 21 に割り当てることと同様の効果があります。これは、IPVS で 認識されて適切に転送されます。

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport 21 -j
MARK --set-mark 21
```

```
/sbin/iptables -t mangle -A PREROUTING -p tcp -d n.n.n.n/32 --dport
10000:20000 -j MARK --set-mark 21
```

iptables コマンドでは、*n.n.n.n* は **Piranha Configuration Tool** の **仮想サーバー** サブセクション内で定義されている FTP 仮想サーバーのフローティング IP で置き換える必要があります。

警告

上述のコマンドはすぐに効果を発揮しますが、システムを再起動すると維持されません。ネットワークパケットのフィルタ設定が再起動後に復元するようにするには「ネットワークパケットフィルター設定の保存」を参照してください。

最後に、適切なサービスが正しいランレベルでアクティベートするように確認してください。これに関する詳細情報は、「LVS ルーターでのサービス設定」を参照してください。

3.6. ネットワークパケットフィルター設定の保存

ユーザーの状況に応じた適切なネットワークパケットフィルタを設定した後は、 その設定を保存して再起動後に復元するようにします。**iptables** には以下のコマンドを実行します。

```
/sbin/service iptables save
```

これで **/etc/sysconfig/iptables** 内の設定が保存され、起動時に再度呼び出すことができます。

このファイルが書き込まれると、**/sbin/service** コマンドを使用して **iptables** の開始、停止、ステータスの確認（ステータススイッチを使用）ができるようになります。**/sbin/service** は自動的に適切な モジュールを読み込みます。**/sbin/service** コマンドの使用法の例は、「**Piranha Configuration Tool** サービスの開始」 を参照してください。

最後に、適切なサービスが正しいランレベルでアクティベートするように確認してください。これに関する詳細情報は、「LVS ルーターでのサービス設定」を参照してください。

次の章では、**Piranha Configuration Tool** を使用して LVS ルーター を設定する方法と Load Balancer Add-On をアクティベートする手順を説明しています。

第4章 PIRANHA CONFIGURATION TOOL を使った LOAD BALANCER ADD-ONの設定

Piranha Configuration Tool は、 Load Balancer Add-Onに必要な設定ファイルである **/etc/sysconfig/ha/lvs.cf** の作成に関して体系的なアプローチを提供します。本章では、**Piranha Configuration Tool** の基本的な操作と設定後に Load Balancer Add-Onをアクティベートする方法を説明します。

重要

Load Balancer Add-Onの設定ファイルは、厳格なフォーマットルールにしたがいます。**lvs.cf** での構文エラーとソフトウェア障害を防ぐには、**Piranha Configuration Tool** が最善の方法となります。

4.1. 必要なソフトウェア

Piranha Configuration Tool を使用するには **piranha-gui** サービスがプライマリー LVS ルーターで稼働している必要があります。Load Balancer Add-Onの設定には、最低でも **links** のようなテキストベースの Webブラウザが必要です。別のマシンから LVS ルーターにアクセスしている場合は、root ユーザーでのプライマリー LVS ルーターへの **ssh** 接続も必要になります。

プライマリー LVS ルーターの設定中は、端末ウィンドウで同時 **ssh** 接続を開いておくとよいでしょう。この接続は **pulse** や他のサービスを再起動し、ネットワークパケットフィルターを設定、 トラブル解決中に **/var/log/messages** を監視するセキュアな方法を提供します。

以下に続く 4 セクションでは、**Piranha Configuration Tool** の設定方法を説明し、それを使用した Load Balancer Add-Onの設定方法の指示をご紹介します。

4.2. PIRANHA CONFIGURATION TOOLへのログイン

Load Balancer Add-Onを設定する際は、常に **Piranha Configuration Tool** を使ったプライマリルーターの設定から始めてください。これを行うには、**piranha-gui** サービスが実行中で、「**Piranha Configuration Tool** のパスワード設定」の説明にあるように管理パスワードが設定されていることを確認します。

ローカルでマシンにアクセスしている場合は、Web ブラウザで **http://localhost:3636** を開いて **Piranha Configuration Tool** にアクセスすることができます。そうでない場合は、ホスト名もしくは実 IP アドレスに続けて **:3636** を入力してください。ブラウザーが接続されると、図4.1「開始画面」の画面が表示されます。

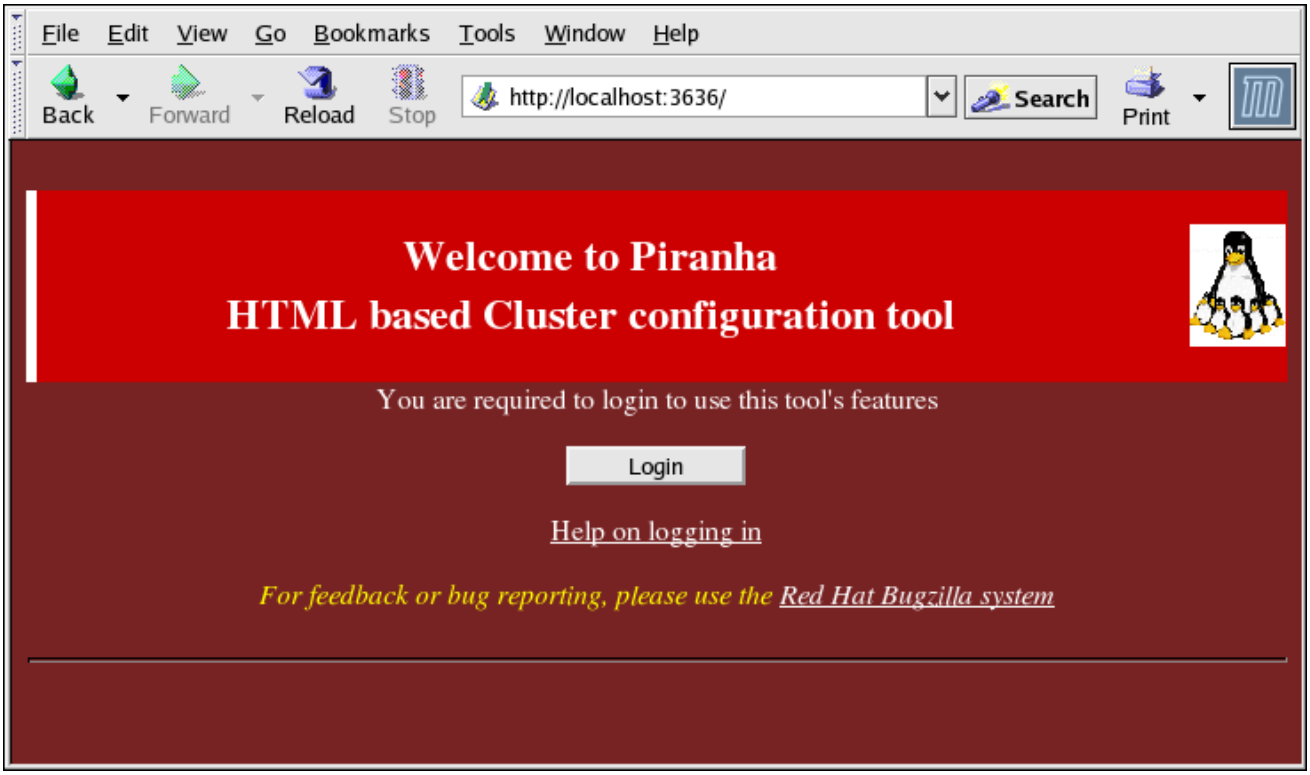


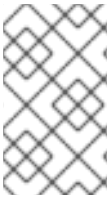
図4.1 開始画面

ログイン (Login) ボタンをクリックし、**ユーザー名 (Username)** フィールドには **piranha** を、**パスワード (Password)** フィールドには作成した管理パスワードを入力します。

Piranha Configuration Tool は*パネル*と呼ばれる 4 つの画面で構成されています。さらに、**仮想サーバー (Virtual Servers)** パネルには 4 つの *サブセクション*があります。**制御/監視 (CONTROL/MONITORING)** パネルがログイン画面の後に最初に現れるパネルです。

4.3. 制御/監視 (CONTROL/MONITORING)

制御/監視 (CONTROL/MONITORING) パネルは、Load Balancer Add-Onの制限ランタイムステータスを示します。表示されるのは、**pulse** デーモン、LVS ルーティングテーブル、LVS 派生の **nanny** プロセスのステータスです。



注記

現行 LVS ルーティングテーブル (CURRENT LVS ROUTING TABLE) および **現行 LVS プロセス (CURRENT LVS PROCESSES)** のフィールドは、「Load Balancer Add-Onを開始する」 にあるように Load Balancer Add-Onが開始されるまで空白のままです。

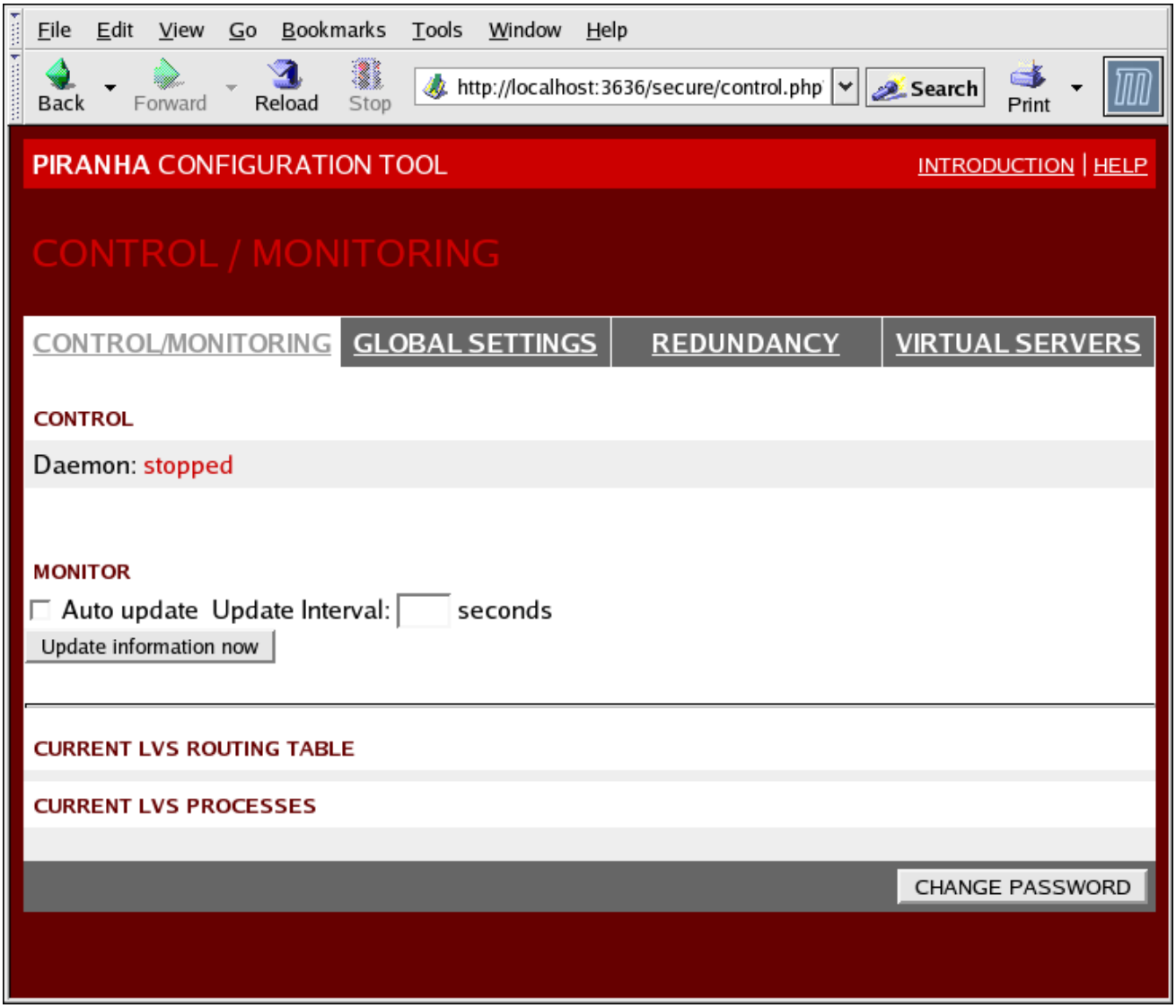


図4.2 制御/監視 (CONTROL/MONITORING) パネル

自動更新 (Auto update)

このページのステータス表示は、ユーザーの設定した間隔で自動的に更新ができます。この機能を有効にするには、**自動更新 (Auto update)** チェックボックスをクリックして **秒単位の更新頻度 (Update frequency in seconds)** テキストボックス（デフォルトは 10 秒） 内の更新頻度をセットします。

この自動更新機能を 10 秒以下の間隔にセットすることは推奨されません。そうした場合、ページが頻繁に更新し過ぎて **自動更新 (Auto update)** の間隔の再設定が困難になります。この問題に遭遇した場合は、別のパネル上でクリックしてから、**自動更新 (Auto update)** に戻ります。

自動更新 (Auto update) 機能は **Mozilla** などのすべてのブラウザで動作するわけではありません。

今すぐ情報を更新する (Update information now)

このボタンをクリックすると、ステータス情報を手動で更新することができます。

パスワードを変更 (CHANGE PASSWORD)

このボタンをクリックすると、**Piranha Configuration Tool** の管理者 パスワードの変更に関する情報を含むヘルプ画面に移動します。

4.4. グローバル設定 (GLOBAL SETTINGS)

グローバル設定 (GLOBAL SETTINGS) パネルでは、プライマリー LVS ルーターのパブリックおよびプライベートネットワークインターフェースのネットワーキングの詳細を設定します。

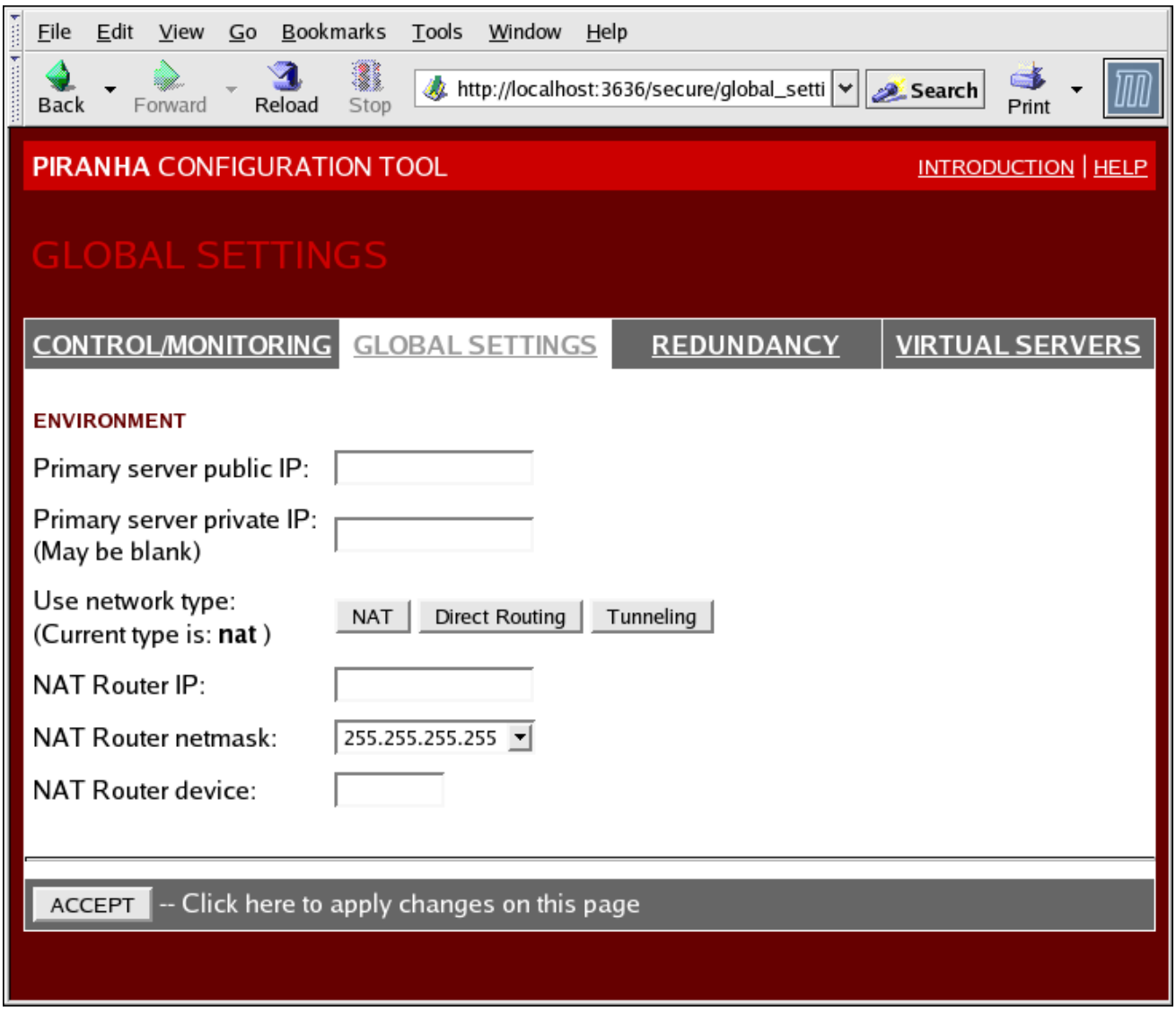


図4.3 グローバル設定 (GLOBAL SETTINGS) パネル

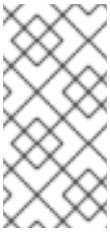
このパネルの上半分では、プライマリー LVS ルーターのパブリックおよびプライベートネットワークインターフェースを設定します。これらは「NAT を使って Load Balancer Add-On のネットワークインターフェースを設定する」で既に設定されているインターフェースです。

Primary server public IP

このフィールドには、プライマリー LVS ノード向けのパブリックで迂回可能な実 IP アドレスを入力します。

Primary server private IP

プライマリー LVS ノード用の代替ネットワークインターフェースの実 IP アドレスを入力します。このアドレスは、バックアップルーターの代替ハートビートチャンネルとしてのみ使用されるもので、「NAT を使って Load Balancer Add-On のネットワークインターフェースを設定する」で割り当てられた実プライベート IP アドレスと関連付ける必要はありません。この欄は空白のままにしてもかまいませんが、そうするとバックアップ LVS ルーターが使用する代替ハートビートチャンネルがないことになり、単一障害点を生み出すことになってしまいます。



注記

ダイレクトルーティング (Direct Routing) の設定には、プライベート IP アドレスは必要ありません。これは、すべての実サーバーも LVS ディレクタも同じ仮想 IP アドレスを共有しており、また同じ IP ルート設定になっている必要があるためです。

注記

プライマリー LVS ルーターのプライベート IP は、イーサネットアダプタやシリアルポートなど TCP/IP を受け付けるインターフェースであれば設定が可能です。

TCP Timeout

TCP セッションがタイムアウトするまでの時間を秒単位で入力します。デフォルト値は 0 です。

TCP Fin Timeout

FIN パケット受信後に TCP セッションがタイムアウトするまでの時間を秒単位で入力します。デフォルト値は 0 です。

UDP Timeout

UDP セッションがタイムアウトするまでの時間を秒単位で入力します。デフォルト値は 0 です。

Use network type

NAT ボタンをクリックすると NAT ルーティングが選択されます。

Direct Routing ボタンをクリックするとダイレクトルーティングが選択されます。

以下の 3 フィールドは、プライベートネットワークと実サーバーを接続する NAT ルーターの仮想ネットワークインターフェースを特定して処理します。これらのフィールドはダイレクトルーティングネットワークタイプには適用**されません**。

NAT Router IP

このテキストフィールドにはプライベートフローティング IP を入力します。このフローティング IP は、実サーバーのゲートウェイとして使用されます。

NAT Router netmask

NAT ルーターのフローティング IP に特定のネットマスクが必要な場合は、ドロップダウンリストから選択します。

NAT Router device

このテキストフィールドを使用して、**eth1:1** などのフローティング IP アドレスのネットワークインターフェースのデバイス名を定義します。

注記

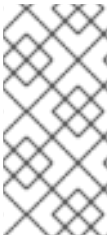
NAT フローティング IP アドレスは、プライベートネットワークに接続されているイーサネットインターフェースへエイリアス化する必要があります。この例では、プライベートネットワークは **eth1** インターフェースにあるので、**eth1:1** がフローティング IP アドレスとなります。

警告

このページを完了した後に **確定 (ACCEPT)** ボタンをクリックして、新規パネルを選択する際に変更が維持されるようにします。

4.5. REDUNDANCY

REDUNDANCY パネルでは、バックアップ LVS ルーターノードの設定と各種ハートビート監視オプションの設定ができます。



注記

この画面が初めて表示される際には、 **Backup** のステータスは「非アクティブ (inactive)」となっており、**有効にする (ENABLE)** ボタンが表示されます。バックアップ LVS ルーターを設定するには、**有効にする (ENABLE)** ボタンをクリックして 図 4.4「**REDUNDANCY** パネル」 のような画面にします。

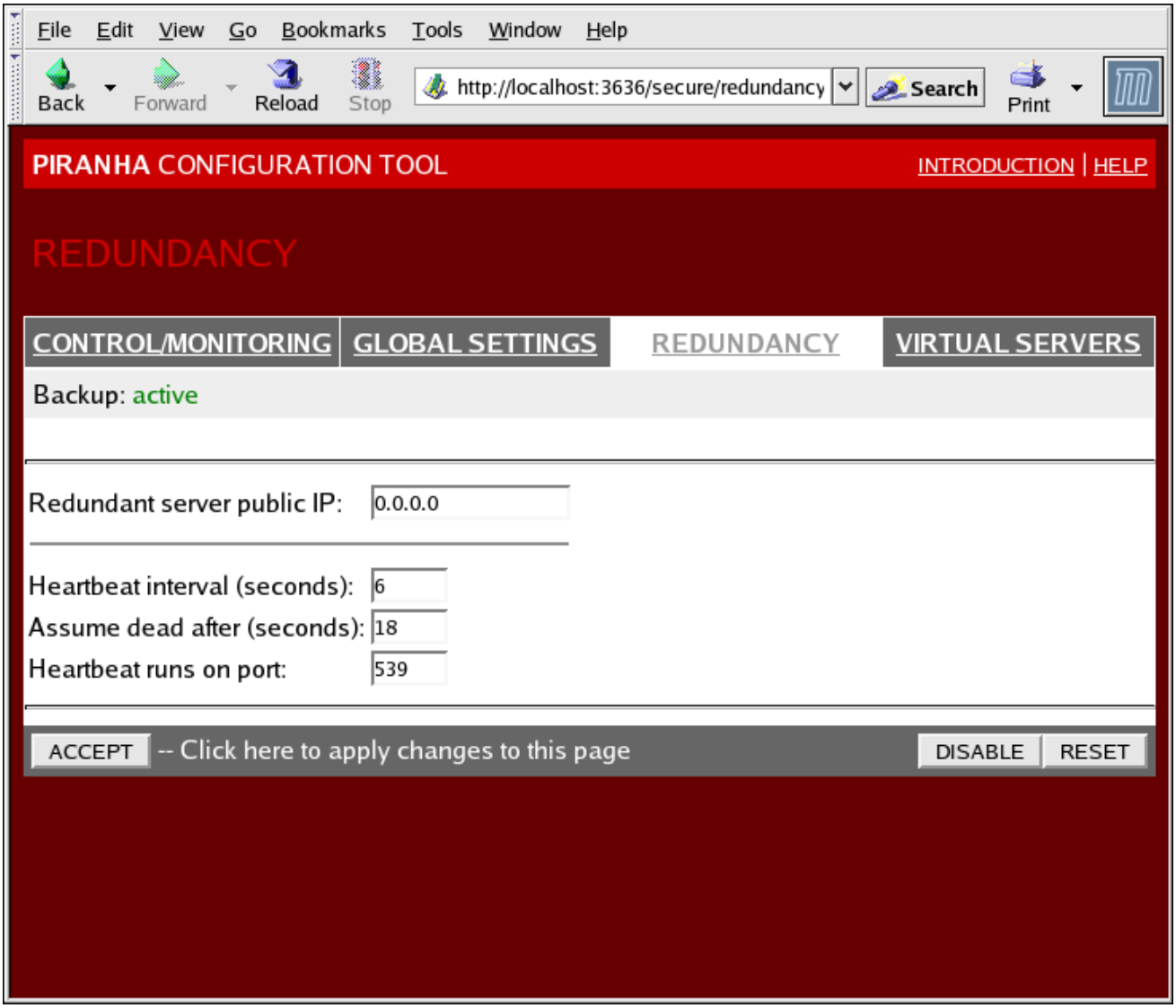


図4.4 REDUNDANCY パネル

Redundant server public IP

バックアップ LVS ルーターのパブリック実 IP アドレスを入力します。

Redundant server private IP

このテキストフィールドには、バックアップノードのプライベート実 IP アドレスを入力します。

Redundant server private IP のフィールドが表示されない場合は、**GLOBAL SETTINGS** パネルに戻り、**Primary server private IP** アドレスを入力して **ACCEPT** をクリックします。

パネルの次のセクションはハートビートチャンネルの設定用で、これはバックアップノードがプライマリーノードの失敗を監視するために使用されます。

Heartbeat Interval (seconds)

このフィールドでは、ハートビート間隔を秒単位でセットします。これは、バックアップノードがプライマリー LVS ノード機能のステータスをチェックする間隔です。

Assume dead after (seconds)

プライマリー LVS ノードがここで設定した秒数間反応しない場合、バックアップ LVS ルーターノードがフェイルオーバーを開始します。

Heartbeat runs on port

このフィールドでは、ハートビートがプライマリー LVS ノードと通信するポートを設定します。フィールドが空白の場合、デフォルトで 539 にセットされます。

このパネルの最後のセクションでは、オプションの同期デーモン (Synchronization Daemon) や各種オプションの有効化および設定を行います。TCP の同期状態を維持するため、同期デーモンでアクティブおよびバックアップ LVS ディレクターを有効にします。有効にすると、アクティブディレクターより設定可能な同期 ID (syncid) が付いたメッセージがマルチキャストで受け取り側のバックアップディレクターにネットワーク送信されます。

警告

Red Hat Enterprise Linux 6.5 では同期メッセージプロトコルの新形式が導入されます。この形式はバックアップノードで永続接続が途中でタイムアウトしてしまいビジネスサービスに障害が発生し、フェールオーバーの際に状態の整合性を欠くことになってしまうのを防ぐ目的で設計されています。

この新形式は Red Hat Enterprise Linux 6.4 またはそれ以前のバージョン、または kernel-2.6.32-406.el6 より以前のカーネルバージョンとの互換性はありません。マスターノードを Red Hat Enterprise Linux 6.5 にアップグレードする前に、まずバックアップノードのアップグレードを行うことをお勧めします。

同期メッセージに旧形式を継続して使用する場合はシェルプロンプトで root になり **echo** コマンドで **sync_version** の値を以下のように設定します。

```
echo 0 > /proc/sys/net/ipv4/vs/sync_version
```

Use Sync Daemon

同期デーモンを有効にする場合は、ボックスにチェックを入れます。

Sync Daemon Interface

同期デーモンがマルチキャストメッセージを送受信する際に使うネットワークインターフェースです。このフィールドのデフォルトインターフェースは **eth0** です。

Sync daemon id

このフィールドでは、マルチキャスト同期メッセージへの識別子 (ID) を設定します。サポート対象となる値は 0 から 255 で、フィールドが空白であればデフォルト値の 0 になります。

警告

このパネルで変更した後は別のパネルに移る前に **ACCEPT** をクリックして変更を保存してください。

4.6. VIRTUAL SERVERS

VIRTUAL SERVERS パネルでは、現行の定義済み仮想サーバーの情報が表示されます。テーブルエントリーではそれぞれ、仮想サーバーの状態、サーバー名、サーバーに割り当てられた仮想 IP、仮想 IP のネットマスク、サーバーが通信するポート番号、使用されるプロトコル、仮想デバイスインターフェース、が表示されます。

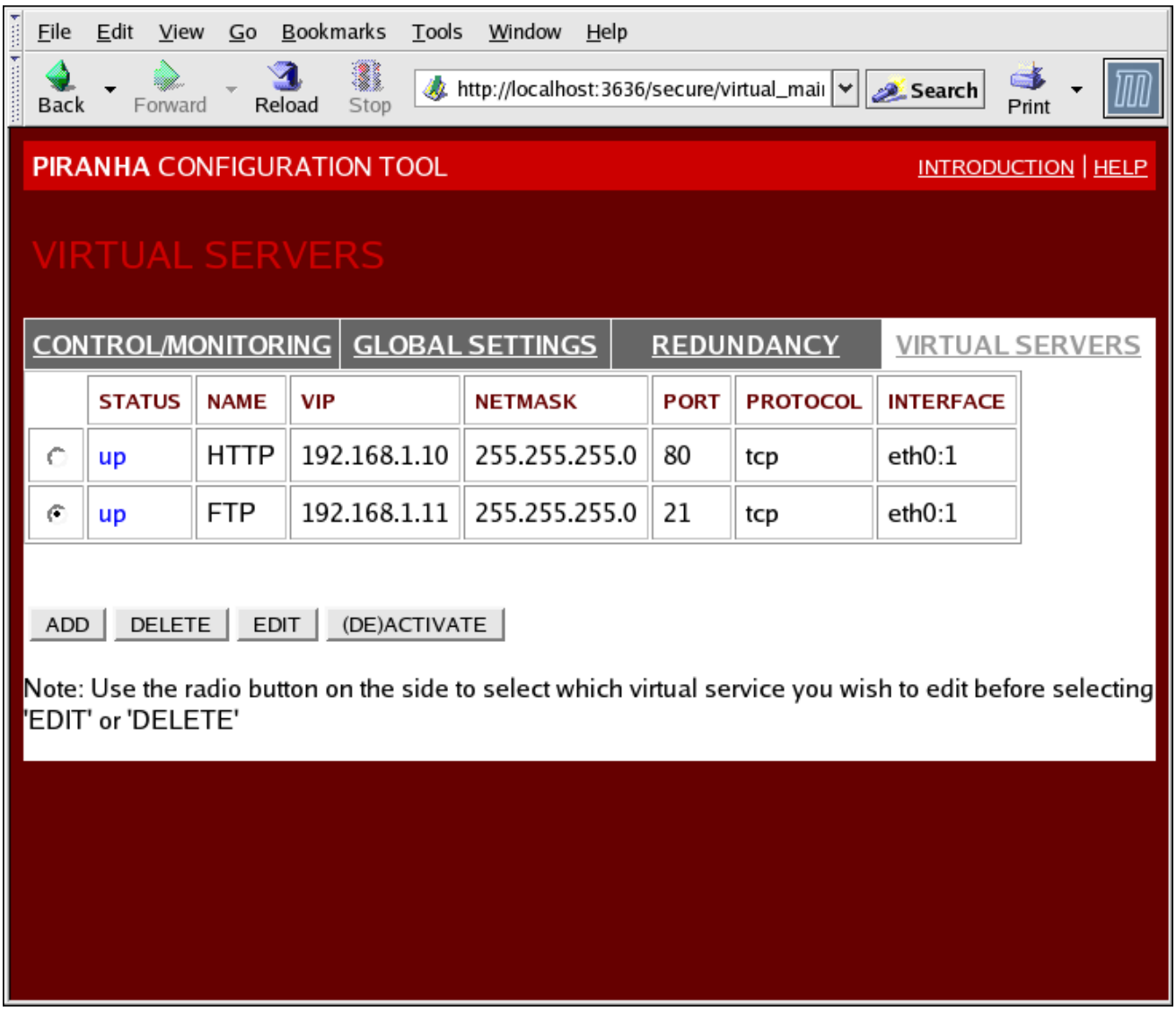


図4.5 VIRTUAL SERVERS パネル

VIRTUAL SERVERS パネルで表示される各サーバーは *サブセクション* と呼ばれるその後に続く画面で設定できます。

サービスを追加するには **ADD** ボタンをクリックします。サービスを削除するには、仮想サーバーの横にあるラジオボタンをクリックして選択し、**DELETE** ボタンをクリックします。

テーブル内の仮想サーバーを有効/無効にするには、そのラジオボタンをクリックしてから **(DE)ACTIVATE** ボタンをクリックします。

仮想サーバーを追加してから設定するには、該当サーバーの左側にあるラジオボタンをクリックして **EDIT** ボタンをクリックし、 **VIRTUAL SERVER** サブセクションを表示させます。

4.6.1. VIRTUAL SERVER サブセクション

図4.6「**VIRTUAL SERVERS** サブセクション」 にある **VIRTUAL SERVER** サブセクションでは、個別の仮想サーバーが設定できます。この仮想サーバーに特定して関連するサブセクションへのリンクは、ページ上部にあります。ただし、この仮想サーバーに関連するサブセクションを設定する前に、このページを完了させて **ACCEPT** ボタンをクリックしてください。

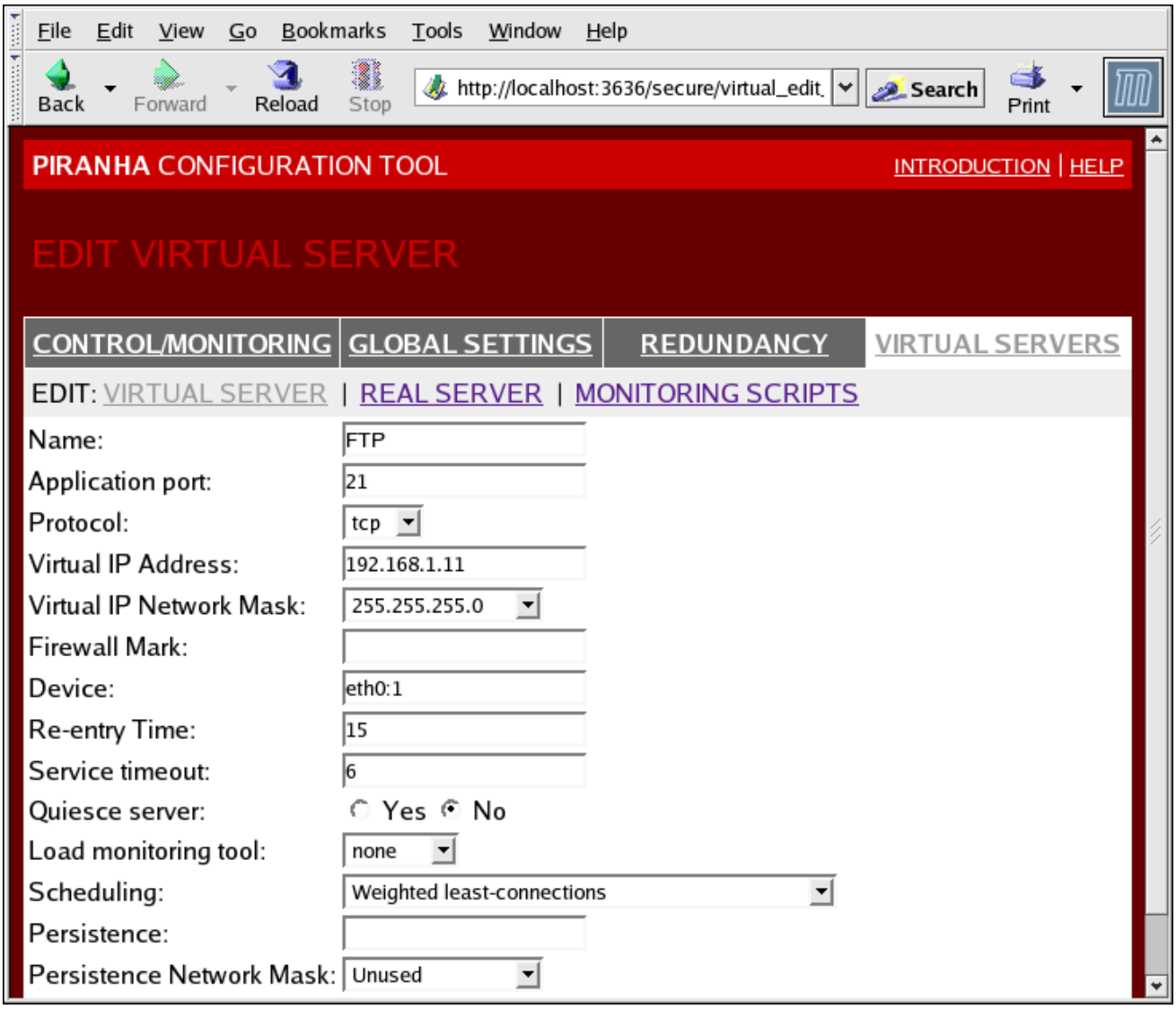


図4.6 VIRTUAL SERVERS サブセクション

Name

仮想サーバーを識別するための説明的な名前を入力します。この名前は、マシンのホスト名 **ではない** ので、説明的で分かりやすいものにします。HTTP などの仮想サーバーが使用するプロトコルを参照する名前でもかまいません。

Application port

サービスアプリケーションがリッスンするポート番号を入力します。この例は HTTP サービスなので、ポート 80 が使用されます。

Protocol

ドロップダウンメニューで UDP か TCP を選択します。Web サーバーは通常、TCP プロトコルで通信するので、上記の例では TCP が選択されています。

Virtual IP Address

このフィールドには、仮想サーバーのフローティング IP アドレスを入力します。

Virtual IP Network Mask

ドロップダウンメニューでこの仮想サーバー用のネットマスクを設定します。

Firewall Mark

マルチポートプロトコルを構築している、または別の関連したプロトコル用に複数ポート仮想サーバーを作成している場合を除いて、このフィールドにはファイアウォールマークの整数を入力 **しない** で下さい。この例では上記の仮想サーバーは、ポート 80 上の HTTP とポート 443 上の HTTPS への接続を構築しているため、**Firewall Mark** を 80 としてあります。永続性と組み合わせることでこの技術は、安全でない Web ページと安全な Web ページの両方にアクセスするユーザーが同じ実サーバーに回され、その状態を保持するようにします。

警告

このフィールドにファイアウォールマークを入力することで、 IPVS はこのファイアウォールマークがあるパケットが同様に処理されていることを認識するようになりますが、ファイアウォールマークを実際に割り当てるには **Piranha Configuration Tool** 外での設定が必要になります。マルチポートサービスの作成に関しては 「マルチポートサービスと Load Balancer Add-On」 を、高可用性の FTP 仮想サーバーの作成に関しては 「FTP の設定」 を参照してください。

Device

Virtual IP Address フィールドで定義してあるフローティング IP アドレスにバインドするネットワークデバイスの名前を入力します。

パブリックフローティング IP アドレスは、パブリックネットワークに接続されたイーサネットインターフェースにエイリアス化する必要があります。この例では、パブリックネットワークは **eth0** インターフェース上にあるため、デバイス名として**eth0:1** を入力することになります。

Re-entry Time

障害の後にアクティブ LVS ルーターが実サーバーをサーバープールに戻すまでの秒数を整数で入力します。

Service Timeout

実サーバーが停止しているとみなされ、サーバープールから削除されるまでの秒数を整数で入力します。

Quiesce server

Quiesce server ラジオボタンを選択した場合、実サーバーが使用できなくなると加重が 0 に設定されます。これによりこのサーバーは実質的に無効になります。その後、実サーバーが利用可能になった場合はオリジナルの加重に戻されサーバーが再度有効になります。**Quiesce server** が無効になっていると、障害が発生している実サーバーはサーバーテーブルから削除されます。利用できなくなっていたサーバーが利用可能になると仮想サーバーテーブルに戻されます。

Load monitoring tool

rup または **ruptime** を使用すると、各種実サーバー上のロードを LVS ルーターで監視できるようになります。ドロップダウンメニューから **rup** を選択した場合は、各実サーバーは **rstatd** サービスを実行する必要があります。**ruptime** を選択した場合は、各実サーバーは **rwhod** サービスを実行する必要があります。

警告

負荷の監視は負荷分散と同じ **ではありません**。そして加重スケジューリングアルゴリズムと組み合わせた場合、スケジューリング動作の予測が困難になります。また、負荷監視を使用する場合、実サーバーは Linux マシンである必要があります。

Scheduling

ドロップダウンメニューからスケジューリングアルゴリズムを選択します。デフォルトは **加重最小接続** です。スケジューリングアルゴリズムについては、「スケジューリングのアルゴリズム」を参照してください。

Persistence

クライアントのトランザクション中に、管理者が仮想サーバーへの永続的な接続を必要とする場合は、このテキストフィールド内に接続がタイムアウトになるまでの非アクティブの時間を秒数で入力します。

重要

上記の **Firewall Mark** フィールドに値を入力している場合は、persistence にも値を入力してください。また、ファイアウォールマークと persistence を一緒に使用する場合は、ファイアウォールマークのある仮想サーバーで persistence の値が同じになるようにしてください。ファイアウォールマークと persistence についての詳細は「永続性とファイアウォールマーク」を参照してください。

Persistence Network Mask

永続性を特定のサブネットに限定するには、ドロップダウンメニューから該当するネットワークマスクを選択します。

注記

ファイアウォールマークが現れる前は、サブネットに制限された永続性が接続をバンドルする基本的な手段でした。現在では、永続性をファイアウォールマークとの関連で使用して同様の目的を達成するのが最善の方法です。

警告

このパネルで変更した後は別のパネルに移る前に **ACCEPT** をクリックして変更を保存してください。

4.6.2. REAL SERVER サブセクション

パネル上部の **REAL SERVER** サブセクションリンクをクリックすると、**EDIT REAL SERVER (実サーバーの編集)** サブセクションが表示されます。これは特定の仮想サービス用の物理サーバーホストのステータスを表示するものです。

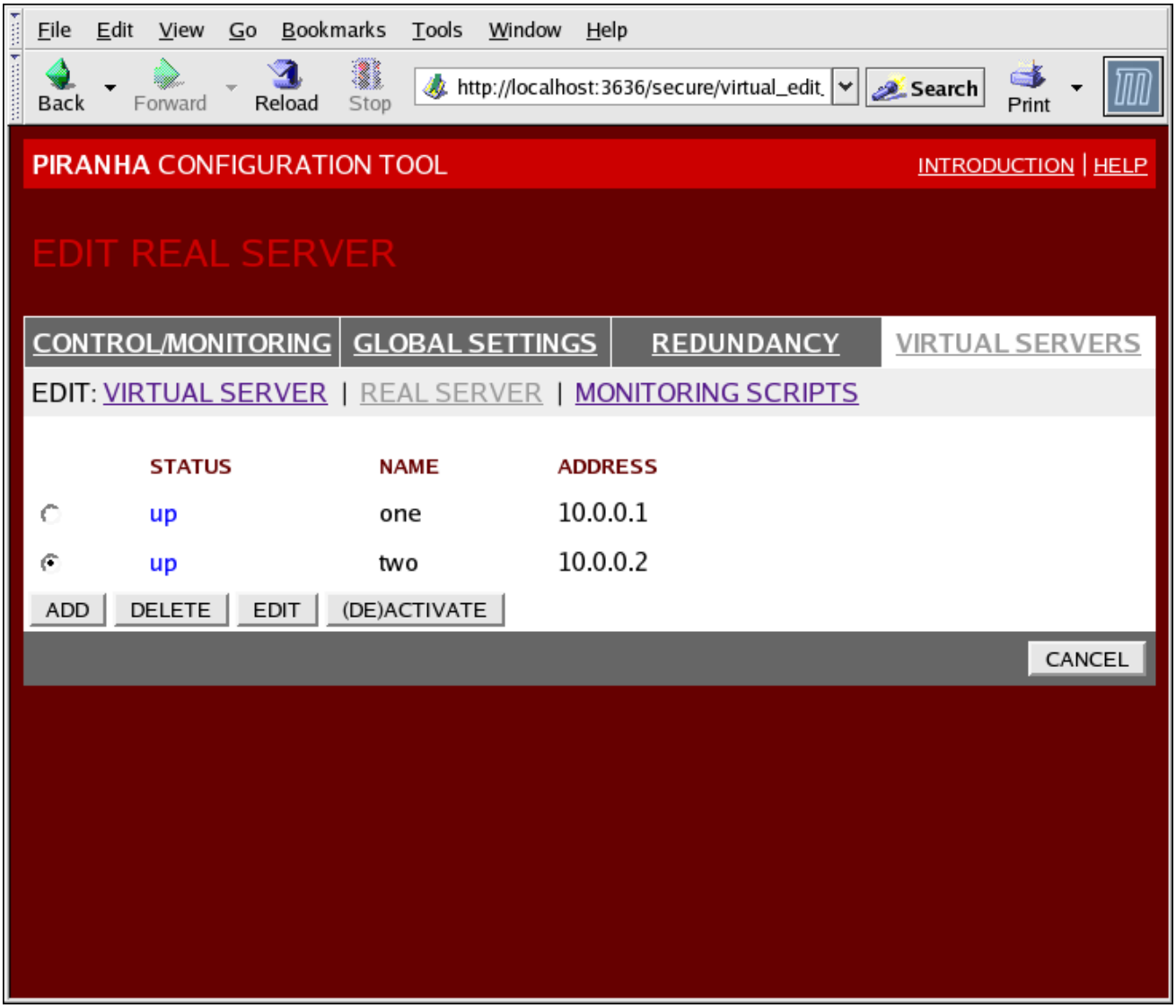


図4.7 REAL SERVER サブセクション

新規サーバーを追加するには **ADD** ボタンをクリックします。既存のサーバーを削除するには、該当サーバーの横のラジオボタンを選択して **DELETE** ボタンをクリックします。図4.8「**REAL SERVER** 設定パネル」 のような **EDIT REAL SERVER** パネルを表示するには **EDIT** ボタンをクリックします。

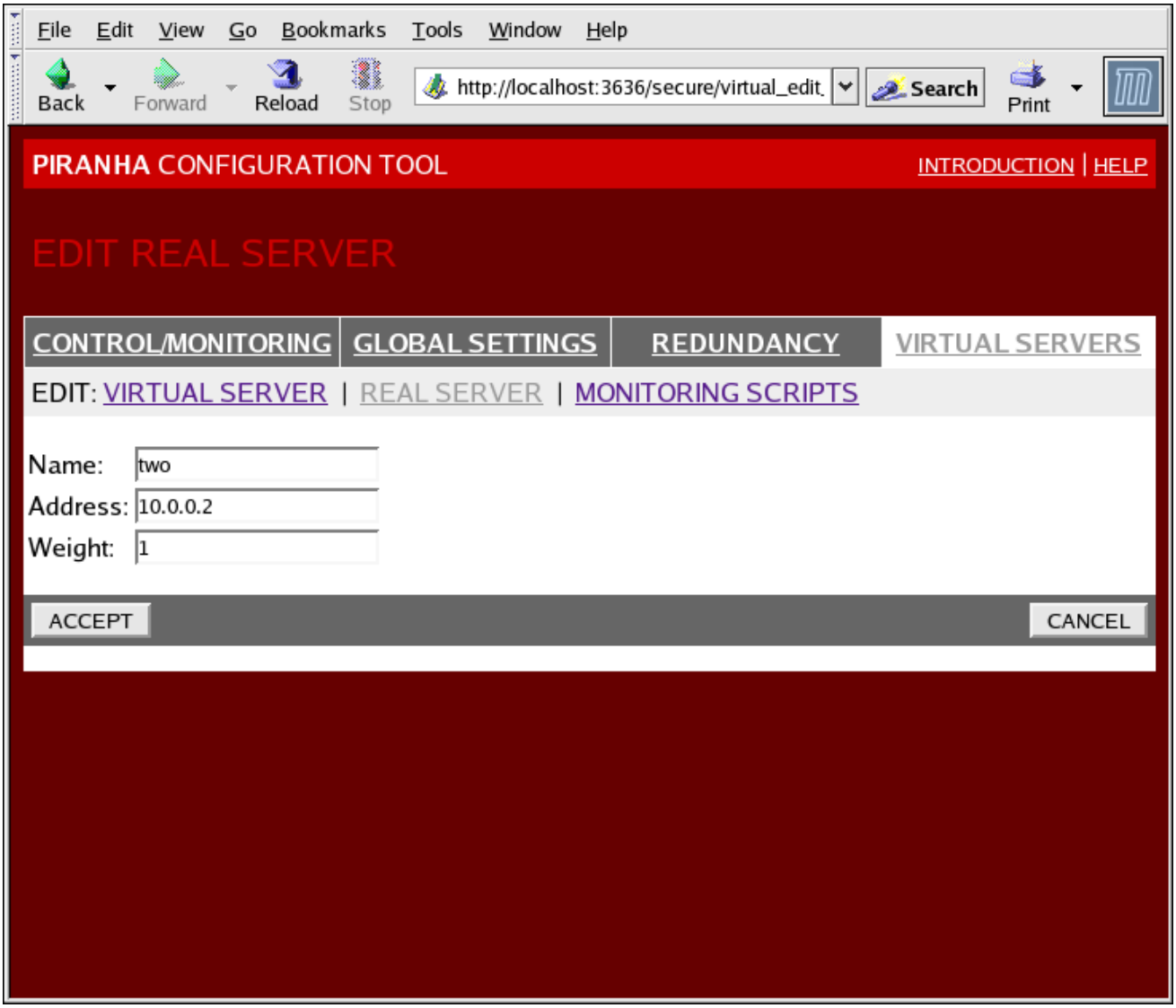
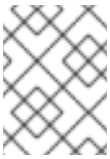


図4.8 REAL SERVER 設定パネル

このパネルは以下の三つのエントリフィールドで構成されます。

`Name`

実サーバー用の説明的名前です。



注記

この名前は、マシンのホスト名 **ではない** ので、説明的で分かりやすいものにします。

`Address`

実サーバーの IP アドレスです。関連付けられた仮想サーバーにはすでにリスニングポートが指定されているので、ポート番号は追加しないでください。

`Weight`

プール内の他のホストに対する該当ホストの相対的なキャパシティを示す整数値です。この値は任意なものですが、プール内の他の実サーバーに対する割合として扱ってください。サーバー加重についての詳細は、「サーバーの重み付けとスケジューリング」を参照してください。

警告

このパネルで変更した後は別のパネルに移る前に **ACCEPT** をクリックして変更を保存してください。

4.6.3. EDIT MONITORING SCRIPTS サブセクション

パネル上部の **MONITORING SCRIPTS** リンクをクリックします。**EDIT MONITORING SCRIPTS** サブセクションでは、管理者が送信/予期の文字列シーケンスを指定して、仮想サーバーのサービスが実サーバー上で機能していることを確認できます。また、管理者がカスタム化したスクリプトを指定し、動的に変化しているデータを必要とするサービスをチェックすることもできます。

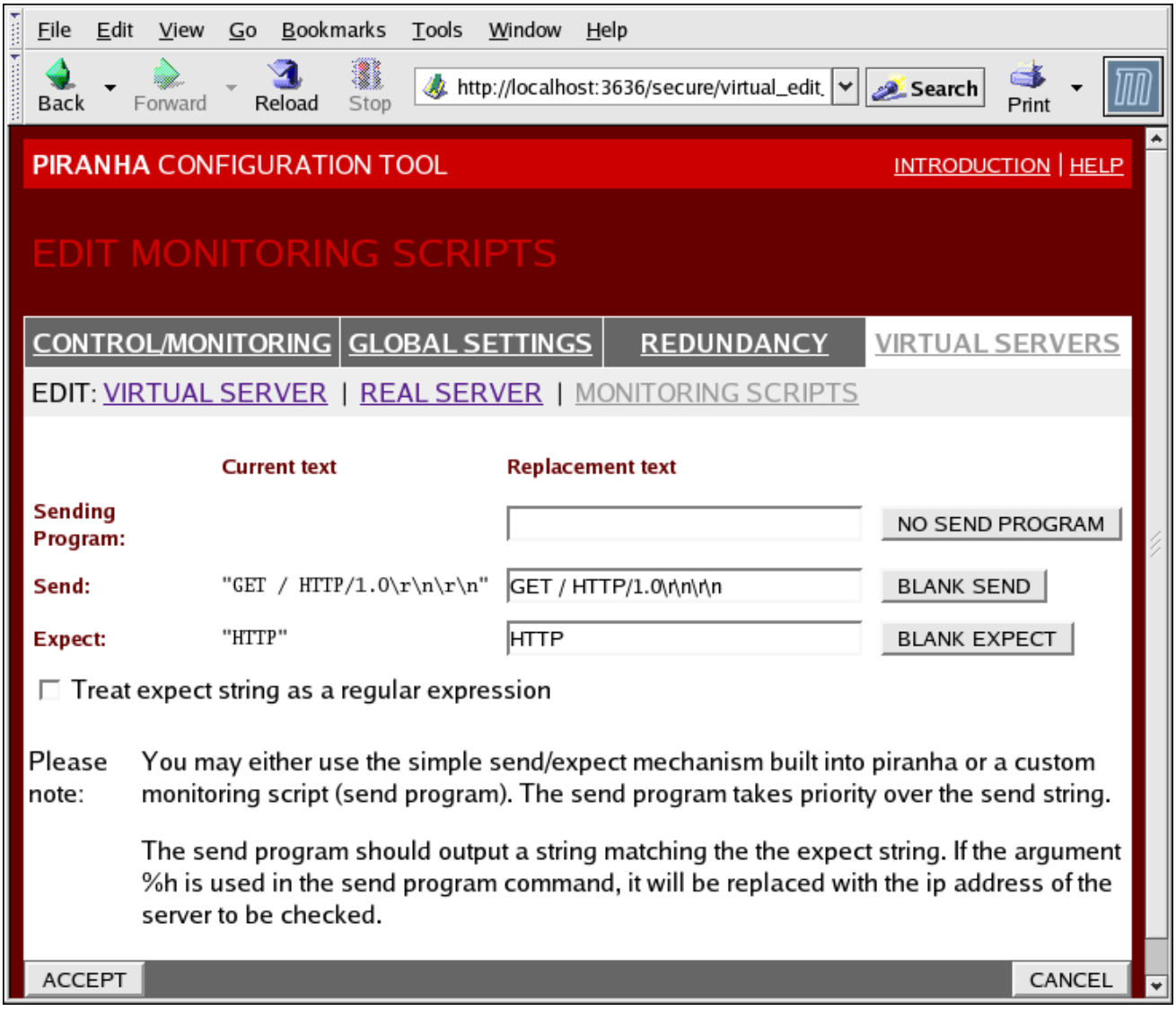
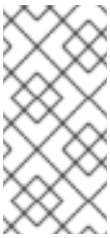


図4.9 EDIT MONITORING SCRIPTS サブセクション

Sending Program

より高度なサービスの検証には、このフィールドを使用してサービスチェックスクリプトへのパスを指定することができます。この機能は特に、HTTPS や SSL などの動的に変化するデータを必要とするサービスに役に立ちます。

この機能を使用するには、テキスト応答を返し、それを実行ファイルになるようにセットし、**Sending Program** フィールド内にそのパスを入力するようなスクリプトを書く必要があります。



注記

実サーバープール内の各サーバーを確実にチェックするには、**Sending Program** フィールド内でスクリプトへのパスの後に特別トークンの **%h** を使います。スクリプトが **nanny** デーモンに呼び出される際にこのトークンは実サーバーの IP アドレスで置き換えられます。

以下のサンプルは、外部サービスチェックスクリプトを書く際のガイドです。

```
#!/bin/sh

TEST=`dig -t soa example.com @$1 | grep -c dns.example.com

if [ $TEST != "1" ]; then
 echo "OK
else
 echo "FAIL"
fi
```

注記

外部プログラムが **Sending Program** フィールド内に記入された場合、**Send** フィールドは無視されます。

Send

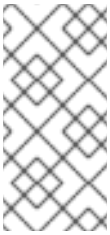
このフィールドには、各実サーバーに送信する **nanny** デーモンの文字列を入力します。デフォルトでは、送信フィールドは HTTP 用になっています。この値は必要に応じて変更できます。このフィールドを空白のままにすると、**nanny** デーモンはポートを開こうとして、これが成功するとサービスが実行中だとみなします。

このフィールドで許可されるのは、1 つの送信シーケンスのみです。このシーケンスに含められるのは、印刷可能な ASCII 文字と以下のエスケープ文字のみです。

- 行送りの **\n**
- 改行の **\r**
- タブ文字の **\t**
- 次に続く文字をエスケープする ****

Expect

サーバーが正常に機能している場合に返すテキスト応答を入力します。ユーザー自身が送信プログラムを書いている場合は、成功した時に送信を指示している応答を入力します。



注記

あるサービスに何を送信するかを判断するには、実サーバー上のポートへの **telnet** 接続を開き、返ってくる応答を待ちます。例えば、FTP は接続時に 220 を報告するので、**Send** フィールドに **quit (終了)** を、**Expect** フィールドに **220** を入力します。

警告

このパネルで変更した後は別のパネルに移る前に **ACCEPT** をクリックして変更を保存してください。

Piranha Configuration Tool を使用して仮想サーバーを設定したら、特定の設定ファイルをバックアップ LVS ルーターにコピーする必要があります。詳細は 「設定ファイルの同期」 を参照してください。

4.7. 設定ファイルの同期

プライマリー LVS ルーターを設定した後は、Load Balancer Add-Onを開始する前に設定ファイルをいくつかバックアップ LVS ルーターにコピーする必要があります。

対象ファイルは以下のものです。

- **/etc/sysconfig/ha/lvs.cf** – LVS ルーターの設定ファイル
- **/etc/sysctl** – カーネル内のパケット転送をオンにする設定ファイル
- **/etc/sysconfig/iptables** – ファイアウォールマークを使用している場合は、使用しているネットワークパケットによってこれらのファイルのいずれかを同期する必要があります。

重要

Piranha Configuration Tool を使用して Load Balancer Add-Onを設定する場合は、**/etc/sysctl.conf** ファイルと **/etc/sysconfig/iptables** ファイルは、**変更されません。**

4.7.1. lvs.cf の同期

LVS 設定ファイル **/etc/sysconfig/ha/lvs.cf** が作成される、もしくは更新される際はいつでも、それをバックアップ LVS ルーターノードにコピーする必要があります。

警告

アクティブとバックアップの LVS ルーターノードには同一の **lvs.cf** ファイルがある必要があります。これらの LVS ルーターノード間で LVS 設定ファイルが一致しない場合は、フェイルオーバーが妨げられる可能性があります。

これを実行する最善の方法は **scp** コマンドの使用です。

重要

scp を使用するには、バックアップルーター上で **sshd** が稼働している必要があります。LVS ルーター上で必要なサービスを正しく設定する詳細な方法については、「LVS ルーターでのサービス設定」 を参照してください。

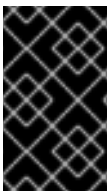
root で プライマリー LVS ルーターから以下のコマンドを実行し、ルーターノード間の **lvs.cf** ファイルを同期します。

```
scp /etc/sysconfig/ha/lvs.cf n.n.n.n:/etc/sysconfig/ha/lvs.cf
```

コマンド内の *n.n.n.n* はバックアップ LVS ルーターの実 IP アドレスで置き換えます。

4.7.2. sysctl の同期

sysctl はほとんどの場合一度だけ修正されるファイルです。このファイルは起動時に読み込まれ、カーネルにパケット転送をオンにするように指示します。



重要

カーネルでパケット転送が有効になっているかどうかわからない場合は、チェック方法と必要な場合はこの主要機能を有効にする方法について 「パケット転送をオンにする」 を参照してください。

4.7.3. ネットワークパケットフィルタルールの同期

iptables を使用している場合、バックアップ LVS ルーター上で適切な設定ファイルを同期する必要があります。

ネットワークパケットフィルタルールを変更するには、root でプライマリー LVS ルーターから以下のコマンドを実行します。

```
scp /etc/sysconfig/iptables n.n.n.n:/etc/sysconfig/
```

コマンド内の *n.n.n.n* はバックアップ LVS ルーターの実 IP アドレスで置き換えます。

次に、バックアップルーターへの **ssh** セッションを開くか、root でマシンにログインして以下のコマンドを実行します。

```
/sbin/service iptables restart
```

これらのファイルのバックアップルーターへのコピーが完了し、適切なサービスが開始されると (このトピックについては 「LVS ルーターでのサービス設定」 を参照してください)、Load Balancer Add-Onを開始する準備が整ったことになります。

4.8. LOAD BALANCER ADD-ONを開始する

Load Balancer Add-Onを開始するには、二つの root ターミナルが同時に開くようにするか、二つの root が同時にプライマリー LVS ルーターへの **ssh** セッションを開くようにするのが最適です。

ターミナルの 1 つで以下のコマンドを使用してカーネルログメッセージを表示させます。

```
tail -f /var/log/messages
```

そして、もう一方のターミナルで以下のコマンドを実行してLoad Balancer Add-Onを開始します。

```
/sbin/service pulse start
```

カーネルログメッセージでターミナル内の **pulse** サービスのスタートアップ進捗状況をチェックします。以下の出力があれば、pulse デーモンは正常に開始しています。

```
gratuitous lvs arps finished
```

/var/log/messages の表示を停止するには、**Ctrl+c** を押します。

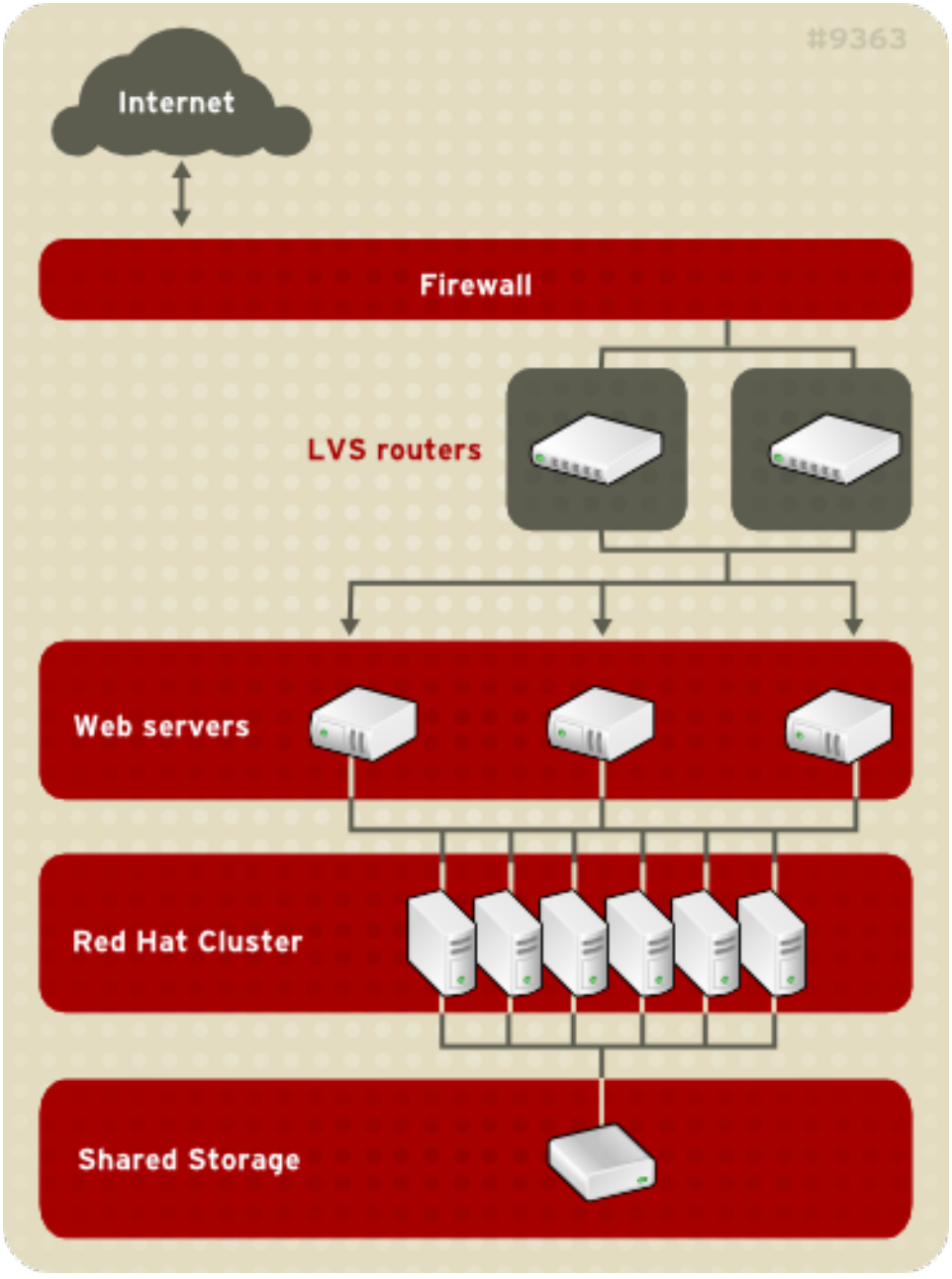
これ以降は、プライマリー LVS ルーターはアクティブ LVS ルーターでもあります。この時点でも Load Balancer Add-On に要求を出すことができますが、Load Balancer Add-Onを使用可能な状態にする前にバックアップ LVS ルーターを開始する必要があります。これを行うには、上記の手順をバックアップ LVS ルーターノード上で繰り返すだけです。

この最終ステップが完了すると、Load Balancer Add-Onは実行中になります。

付録A HIGH AVAILABILITY アドオンを使った LOAD BALANCER ADD-ONの使用

High Availability アドオンとともに Load Balancer Add-Onを使って高可用性の E コマースサイトを導入することができます。このサイトは負荷分散やデータの整合性、アプリケーションの可用性をもたらします。

図A.1「High Availability アドオンを使った Load Balancer Add-On」 にある構成は、ある URL でオンライン商品注文を受け付ける E コマースのサイトを示しています。URL へのクライアントからの要求はファイアウォールを通ってアクティブな LVS ロードバランシングルーターに行き、そこから要求は Web サーバーの 1 つに転送されます。High Availability アドオンノードが動的データを Web サーバーに供給して、そこからデータが要求元のクライアントに転送されます。



図A.1 High Availability アドオンを使った Load Balancer Add-On

動的な Web コンテンツで Load Balancer Add-Onを使用するには三層の (three-tier) 設定が必要になります (図A.1「High Availability アドオンを使った Load Balancer Add-On」 参照)。 Load Balancer Add-On と High Availability アドオンの組み合わせにより、高度な整合性を持ち、単一障害点のない E コ

マースサイトの設定が可能になります。High Availability アドオンはデータベースの高可用性インスタンスや Web サーバーにネットワークアクセス可能なデータベースのセットを実行することができます。

三層設定は、動的コンテンツを提供するために必要となります。Web サーバーが 静的コンテンツ (変化の少ない少量のデータで構成) のみを扱う場合は、二層のロードバランサアドオン設定が適切ですが、Web サーバーが動的コンテンツを扱う場合は、二層設定は不適切なものとなります。動的コンテンツには製品在庫や購入注文、顧客データベース等が含まれ、顧客が最新の正確な情報にアクセスできるようにするために、すべての Web サーバー上で一貫性が必要になります。

各層では、以下の機能が提供されます。

- 第一層 – Web 要求を分配する負荷分散を実行する LVS ルーター
- 第二層 – 要求を処理する Web サーバーセット
- 第三層 – Web サーバーへのデータを扱う High Availability アドオン

図A.1「High Availability アドオンを使った Load Balancer Add-On」 にあるようなロードバランサーアドレス設定では、World Wide Web 上でクライアントシステムが要求を発行します。セキュリティの理由により、これらの要求はファイアウォールから Web サイトに入ります。このファイアウォールはその役割を果たしている Linux システムでも専用のファイアウォールデバイスでもかまいません。冗長性のために、ファイアウォールデバイスはフェイルオーバー設定で設定可能です。ファイアウォールの背後には LVS ルーターがあり、これは負荷分散を提供するもので、アクティブスタンバイモードでの設定ができます。アクティブな負荷分散ルーターは、要求を Web サーバーセットに転送します。

Web サーバーはクライアントからの HTTP 要求を個別に処理して、クライアントに応答を送信します。Load Balancer Add-Onを使うと、LVS ルーターの背後にある Web サーバーを追加することで Web サイトのキャパシティが拡大できます。つまり、LVS ルーターが幅広い Web サーバーセット全体で負荷分散を実行する事になります。さらに、ある Web サーバーが失敗すると、そのサーバーは削除されます。Load Balancer Add-Onは、縮小された Web サーバーセット内で負荷分散を継続します。

付録B 改訂履歴

改訂 1-15.3 **Tue Feb 10 2015** **Noriko Mizumoto**
「NAT を使って Load Balancer Add-On のネットワークインターフェースを設定する」のセクションにある重要欄の翻訳を著者の意向にしたがい修正

改訂 1-15.2 **Fri Feb 6 2015**
翻訳ドラフト

改訂 1-15.1 **Fri Feb 6 2015**
翻訳ファイルを XML ソースバージョン 1-15 と同期

改訂 1-15 **Tue Dec 16 2014** **Steven Levine**
RHEL 6 スプラッシュページに sort_order を実装するため更新

改訂 1-13 **Thu Oct 9 2014** **Steven Levine**
Red Hat Enterprise Linux 6.6 の GA リリース

改訂 1-10 **Tue Nov 19 2013** **John Ha**
Red Hat Enterprise Linux 6.5 の GA リリース

改訂 1-8 **Fri Sep 27 2013** **John Ha**
Red Hat Enterprise Linux 6.5 の Beta リリース

改訂 1-4 **Wed Nov 28 2012** **John Ha**
Red Hat Enterprise Linux 6.4 Beta 向けにリリース

改訂 1-3 **Mon Jun 18 2012** **John Ha**
Red Hat Enterprise Linux 6.3 GA 向けリリース

改訂 1-2 **Fri Dec 2 2011** **John Ha**
Red Hat Enterprise Linux 6.2 の GA リリース

改訂 1-1 **Wed Nov 10 2010** **Paul Kennedy**
Red Hat Enterprise Linux 6 用の初期リリース

索引

シンボル

A

S