



Red Hat Satellite 5.8

Proxy インストールガイド

Red Hat Satellite Proxy Server のインストールおよび設定

Red Hat Satellite 5.8 Proxy インストールガイド

Red Hat Satellite Proxy Server のインストールおよび設定

Red Hat Satellite Documentation Team
satellite-doc-list@redhat.com

法律上の通知

Copyright © 2017 Red Hat.

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, 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 Software Collections 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

本書では、Red Hat Satellite Proxy Server のインストール、設定および更新について解説します。さらに詳しくは、Red Hat Satellite スタートガイドと Red Hat Satellite インストールガイドを参照してください。

目次

第1章 RED HAT SATELLITE PROXY について	3
1.1. RED HAT SATELLITE PROXY SERVER	3
1.2. アーキテクチャーおよび操作	3
第2章 要件	6
2.1. ソフトウェア要件	6
2.2. ハードウェア要件	7
2.3. 追加要件	8
第3章 RED HAT SATELLITE PROXY のインストール	10
3.1. インストール手順の要約	10
3.2. SATELLITE PROXY エンタイトルメントの取得	10
3.3. SATELLITE PROXY エンタイトルメントの SATELLITE へのアップロード	11
3.4. オペレーティングシステムの SATELLITE PROXY ホストへのインストール	11
3.5. RED HAT SATELLITE PROXY SERVER パッケージのインストール	12
3.6. RED HAT SATELLITE PROXY インストールスクリプトの実行	13
3.7. インストール後のタスクの実行	15
3.8. SATELLITE PROXY SERVER インストールの自動化	16
第4章 CUSTOM CHANNEL PACKAGE MANAGER	18
4.1. CUSTOM CHANNEL PACKAGE MANAGER の使用および RED HAT NETWORK PROXY によるローカルパッケージの提供	18
4.2. プロキシのプリキャッシュ設定	20
第5章 CNAME レコードを使用するための SATELLITE PROXY の設定	22
5.1. 前提条件	22
5.2. CNAME レコードの SATELLITE PROXY SERVER 設定への追加	22
5.3. 複数ホスト対応の SSL 証明書の生成と使用	23
第6章 SATELLITE PROXY SERVER の負荷分散	24
6.1. PROXY サービスのロードバランサーへのインストール	24
6.2. SQUID リバースプロキシのインストール	24
6.3. クライアントのセットアップ	27
6.4. 設定のテスト	27
第7章 RED HAT PROXY SERVER インストールのアップグレード	29
7.1. 前提条件	29
7.2. PROXY インストールのアップグレード	29
付録A サンプルの SATELLITE PROXY SERVER 設定ファイル	30
付録B 用語集	31
付録C 改訂履歴	32

第1章 RED HAT SATELLITE PROXY について

1.1. RED HAT SATELLITE PROXY SERVER

Red Hat Satellite Proxy Server はパッケージキャッシングのメカニズムで、Red Hat Satellite の帯域幅要件を低減し、カスタムパッケージの導入を可能にします。Satellite Proxy をお使いのお客様は、Red Hat からのエラータ更新などの RPM パッケージや、ユーザー組織の中央に配置したサーバーで内部生成されたカスタムパッケージをキャッシュします。クライアントシステムはその後、個別にインターネットにアクセスするのではなく、Red Hat Satellite Proxy からそれらの更新を受け取ります。

パッケージは Red Hat Satellite Proxy が管理しますが、クライアントのシステムプロファイルとユーザー情報は安全な中央の Red Hat Satellite Server に格納されます。Satellite Proxy はクライアントシステムと Red Hat Satellite Server を仲介する役目を果たします。Satellite Proxy に格納されるのはパッケージファイルのみになります。すべてのトランザクションは認証され、**Red Hat Update Agent** によりローカルの Satellite Proxy から取り込まれた各パッケージの GPG 署名がチェックされます。

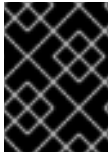
公式の Red Hat パッケージの格納のほかに、Satellite Proxy Server を使用して組織独自のカスタムパッケージをプライベートなチャンネルから配信するよう Satellite Proxy Server を設定することもできます。たとえば、組織独自のソフトウェアを開発し、RPM でパッケージ化し、独自の GPG 署名で署名してから、ローカルの Satellite Proxy Server を利用してネットワーク内のすべての個別システムをこのカスタムソフトウェアの最新バージョンで更新することができます。

Satellite Proxy Server を使用する利点として、以下を挙げるすることができます。

- スケーラビリティ: 1 つの組織が複数のローカル Red Hat Satellite Proxy をサポートできます。
- セキュリティ: クライアントシステムからローカルの Satellite Proxy へ、またローカルの Satellite Proxy から Red Hat Satellite サーバーへのセキュアな接続が維持されます。
- 時間の節約: インターネット経由に比べて、パッケージの配信がローカルエリアネットワークにより大幅に高速化されます。
- 帯域幅の節減: それぞれのクライアントシステムに対してパッケージを別々にダウンロードするのではなく、Red Hat Satellite から一度に複数のパッケージをダウンロードします (ローカル Proxy Server のキャッシングメカニズムを使用)。
- 更新のカスタマイズ: カスタムソフトウェアパッケージの自動パッケージ配信システムを作成するほか、クライアントシステムに必要とされる公式の Red Hat パッケージの配信も自動化します。プライベートの Red Hat Satellite チャンネルをカスタマイズして、組織で社内パッケージの配信を自動化することもできます。
- 設定のカスタマイズ: 特定のアーキテクチャーやオペレーションシステムのバージョンに対して更新を制限したり許可したりできます。

1.2. アーキテクチャーおよび操作

クライアントシステムの Red Hat Update Agent または **Package Updater** は Red Hat Satellite Server と直接は通信しません。クライアント (またはクライアント群) は Satellite Proxy Server に接続し、この Satellite Proxy Server が Red Hat Satellite Server に接続します。したがって、クライアントシステムにはインターネットへの直接アクセスが不要です。クライアントシステムには Satellite Proxy Server へのアクセスのみが必要になります。



重要

Red Hat は、Satellite Proxy サーバーに接続しているクライアントで Red Hat Enterprise Linux の最新の更新を実行し、適切な接続を確保することを強く推奨致します。

Red Hat Satellite Proxy にアクセスするクライアントは、依然として Red Hat Satellite によって認証されますが、この場合 Satellite Proxy は、認証とルート情報の両方を Red Hat Satellite に提供します。認証に成功すると、Red Hat Satellite Server は、Satellite Proxy サーバーに対して、それがクライアントの特定動作を実行する許可を与えられていることを通知します。Satellite Proxy サーバーは、更新されたパッケージすべてをダウンロードし (それらがキャッシュ内がない場合)、それらのパッケージをクライアントシステムに配信します。

クライアントシステムの Red Hat Update Agent または Package Updater による要求はサーバー側で認証されますが、パッケージが HTTP Proxy Caching Server または Satellite Proxy Server (ローカルパッケージの場合) でキャッシュされるためパッケージ配信が大幅に速くなります。Satellite Proxy サーバーとクライアントシステムが LAN 経由で接続されるので、転送速度に影響を与えるのはローカルネットワークの速度のみです。

認証プロセスは以下のように実行されます。

1. クライアントはクライアントセッションの始めにログイン動作を行います。このログインが Red Hat Satellite サーバーに到達するまでに 1 つまたは複数の Satellite Proxy Server を経由します。
2. Red Hat Satellite サーバーはこのクライアントの認証を試みます。認証が成功すると、サーバーは Satellite Proxy Server のチェーンを介してセッショントークンを返します。署名と有効期限を持つこのトークンには、チャンネルサブスクリプションやユーザー名などのユーザー情報が含まれます。
3. それぞれの Satellite Proxy Server はこのトークンを `/var/cache/rhn/` ディレクトリー内のローカルファイルシステムにキャッシュします。キャッシングにより Red Hat Satellite サーバーとの認証で発生するオーバーヘッドが軽減され、Red Hat Satellite のパフォーマンスが大幅に向上します。
4. このセッショントークンはクライアントマシンに送信され、Red Hat Satellite の次の動作で使用されます。

クライアントから見ると、Satellite Proxy Server と Red Hat Satellite サーバーの間には違いがありません。ただし、Red Hat Satellite サーバーから見ると、Satellite Proxy Server は特殊なタイプの Red Hat クライアントになります。したがって、クライアントには、要求が Red Hat Satellite サーバーに到達するまでに使用されるルートによる影響はありません。すべてのロジックは Satellite Proxy Server と Red Hat Satellite サーバーに実装されます。

Custom Channel Package Manager をインストールして、カスタムパッケージを配信するように設定にすることもできます。組織用に特別に作成されたカスタムパッケージを含む、公式の Red Hat パッケージでないすべてのパッケージは、プライベートソフトウェアチャンネル (別名: カスタムソフトウェアチャンネル) からのみ配信できます。プライベートの Red Hat Satellite チャンネルを作成した後に、Red Hat Satellite サーバーにパッケージヘッダーをアップロードして、カスタム RPM パッケージをそのプライベートチャンネルに関連付けます。アップロードするのはヘッダーのみで、実際のパッケージファイルはアップロードしません。ヘッダーにはソフトウェアの依存関係などの重要な RPM 情報が含まれるためそのアップロードが必要となり、これで Red Hat Satellite によるパッケージインストールの自動化が可能になります。実際のカスタム RPM パッケージは Satellite Proxy Server 上に格納され、組織のローカルエリアネットワーク内からクライアントシステムに送信されます。

Satellite Proxy Server を使用できるようにコンピューターネットワークを設定するのは簡単です。クライアントシステムの Red Hat Satellite アプリケーションが Red Hat Satellite サーバーではなく Satellite

Proxy Server に接続するよう設定されている必要があります。詳細については『クライアント設定ガイド』を参照してください。Proxy 側では、チェーン内の次のプロキシを指定する必要があります (最終的には Red Hat Network サーバーで終了します)。Red Hat Package Manager を使用する場合、クライアントシステムはプライベートの Red Hat チャンネルにサブスクライブされる必要があります。

第2章 要件

この章では、Red Hat Satellite Proxy Server のインストール要件に焦点を絞って説明します。

2.1. ソフトウェア要件

インストールを行うには、次のソフトウェア関連のコンポーネントが必要になります。

- ベースオペレーティングシステム: Satellite 5.8 および Satellite Proxy 5.8 は Red Hat Enterprise Linux 6 で対応しています。オペレーティングシステムは、Red Hat でサポートしているいずれかの方法でインストールすることができます。
- Satellite Proxy には、Satellite の同じバージョンまたはそれ以上のバージョンが必要になります。たとえば、Satellite Proxy 5.6 は Satellite 5.6 および Satellite 5.8 と共に使用できますが、Satellite Proxy 5.8 は Satellite 5.8 のみと共に使用できます。
- Satellite Server アカウント内の Satellite Proxy のエンタイトルメント
- Satellite Server アカウント内の Provisioning エンタイトルメント (Satellite Proxy エンタイトルメントと共にパッケージ化されています)
- インストールした Red Hat Enterprise Linux バージョン用の Red Hat Network Tools チャンネルへのアクセス。このチャンネルには、Satellite Proxy のインストールに必要な **configure-proxy.sh** インストールプログラムを格納している `spacewalk-proxy-installer` パッケージが含まれます。
- Satellite Proxy に (Red Hat Network Tools チャンネルから) インストールするすべての `rhncfg*` パッケージ。
- Hosted ユーザーの場合は、Satellite Proxy に (Red Hat Network Tools チャンネルから) インストールする `spacewalk-certs-tools` パッケージか、または Satellite Server ユーザーの場合は親サーバーの証明書の生成に使用する SSL (secure sockets layer) CA 証明書のパスワード。



重要

Red Hat Satellite Proxy Server 5.8 を実行できるのは、Red Hat Enterprise Linux 6 上でのみになります。これは、Satellite 5.8 と併用される場合にすべての RHEL バージョンのクライアントを管理できます。

仮想化サポート

Red Hat がサポートする仮想化環境では、Satellite Proxy を Red Hat Enterprise Linux 5 または 6 にインストールできます。この仮想化環境には、Xen、Hyper-V、KVM および VMware が含まれます。サポートされる環境についての詳細は、<https://access.redhat.com/site/certified-hypervisors> を参照してください。

実稼働環境では、Red Hat は Satellite Proxy を基礎となる物理的なハードウェアで実行される唯一のアプリケーションとして導入し、競合の問題を避けることを推奨します。また、仮想化環境の機能的なサポートは物理的なハードウェアで実行する場合のパフォーマンスと常に同じとは限らないため、仮想化環境の選択や推奨されるチューニングのガイドラインに注意してください。

ご購入される Satellite Proxy には、Red Hat Enterprise Linux Server のサポートされるインスタンスが 1 ずつ含まれます。Satellite Proxy は、Satellite Proxy がオペレーティングシステムによって提供される唯一のアプリケーションおよびサービスとなる Enterprise Linux のフレッシュインストールに対してイン

ストールされる必要があります。Satellite Proxy に含まれる Red Hat Enterprise Linux オペレーティングシステムを使って、その環境内で他のデーモンやアプリケーションまたはサービスを実行することはサポートされていません。

必須パッケージセットの取得

Red Hat Enterprise Linux の各バージョンは、Satellite Proxy をサポートするための特定のパッケージセットを必要とします。他のパッケージを追加すると、インストール時にエラーが発生する場合があります。したがって、Red Hat は以下のように必要なパッケージを取得することをお勧めします。

- キックスタートのインストールの場合は、パッケージグループ **@Base** を指定します。
- CD または ISO イメージを使用して Red Hat Enterprise Linux をインストールする場合は、パッケージグループ **Minimal** を選択します。

2.2. ハードウェア要件

お使いのハードウェアが Satellite Proxy のインストールに適している場合には以下のガイドラインに従ってください。

- Pentium IV Processor または同等のプロセッサ
- 最小 2 GB のメモリー。4 GB のメモリーが推奨されます。
- Red Hat Enterprise Linux のベースインストール用に 最小 5 GB のストレージ
- アーキテクチャー/リリースごとに 6 GB のストレージ

Satellite Proxy が使用するキャッシングのメカニズムは Squid HTTP プロキシになり、クライアント群の帯域幅を大幅に節約します。キャッシュされたパッケージは **/var/spool/squid** ディレクトリーに格納されます。キャッシュで利用できるディスク容量が大きいと、期限切れになる前に RPM ファイルを削除する可能性が低くなります。推奨される容量以下のストレージを設定すると、Proxy サーバーを使用して可能になるパフォーマンスの拡張機能が少なくなります。

Satellite Proxy 5.8 は、チャンネル全体を *事前* キャッシュする機能を特長としています。この機能を使用するには大きなディスク領域が必要です。Red Hat では、ベースチャンネルごとに 20 GB 以上のサイズを推奨しています。この機能をお使いになられる場合は、『[Red Hat Satellite インストールガイド](#)』のハードウェアのガイドラインを参照してください。Proxy の事前キャッシュについては、『[プロキシのプリキャッシュ設定](#)』で論じられています。

Satellite Proxy Server がカスタムパッケージまたはローカルパッケージを配信するように設定する場合には、ローカルパッケージを格納するシステムの **/var** マウントポイントにカスタムパッケージすべてを保持するのに十分なディスク領域があることを確認します。これらカスタムパッケージは **/var/spool/rhn-proxy** 内に格納されています。ローカルパッケージ用に必要となるディスク領域は、配信されるカスタムパッケージの数によって異なります。

Apache Web サーバーに対する負荷は、クライアントシステムが Satellite Proxy に接続する頻度に直接関連してきます。4 時間 (または 240 分) のデフォルト間隔を短くすると、このコンポーネントに対する負荷は大幅に増大します。この値は、各クライアントシステムの **/etc/sysconfig/rhn/rhnsd** 設定ファイル内に設定されます。



注記

このインストール手順では、利用可能なリソースに対してキャッシング機能を最適化できるように設定します。Red Hat Satellite Proxy 内でアクティブなキャッシュを増やすにはメモリーおよびディスク領域を追加します。

2.3. 追加要件

Satellite Proxy Server のインストールが完了したとみなす前に、次の補足要件を満たす必要があります。

完全アクセス

クライアントシステムは、Satellite Proxy Server サービスとポートへの完全なネットワークアクセスが必要になります。

ファイアウォールのルール

Red Hat は、Satellite Proxy とインターネット間にファイアウォールをセットアップすることを強くお勧めします。ただし、お使いの Satellite Proxy の実装によっては、ファイアウォール内に複数の TCP ポートを開く必要があります。

表2.1 Satellite Proxy で開くポート

ポート	方向	理由
80	アウトバウンド	Satellite Proxy はこのポートを使用して Satellite URL にアクセスします。
80	インバウンド	クライアントの要求は HTTP または HTTPS のいずれかを使用して到着します。
443	インバウンド	クライアントの要求は HTTP または HTTPS のいずれかを使用して到着します。
443	アウトバウンド	Satellite Proxy はこのポートを使用して Satellite URL にアクセスします。
5222	インバウンド	Red Hat Network Push 技術を使用している際に、Satellite Proxy 上の jabberd デーモンへの osad クライアントの接続を許可します。
5269	インバウンドとアウトバウンド	Satellite Proxy が Satellite Server に接続されている場合、Red Hat Network Push 技術用の jabberd を使用してサーバー間の接続を可能にするためにこのポートを開いておく必要があります。

同期システム時間

SSL (Secure Sockets Layer) を実行している Web サーバーに接続する際は、時間が非常に重要になります。SSL 証明書が使用前または使用中に期限切れにならないよう、クライアントとサーバーで設定される時間が近接している必要があります。NTP (Network Time Protocol) を使用して時計を同期することが推奨されます。

完全修飾ドメイン名 (FQDN)

Satellite Proxy Server をインストールするシステムは、独自の完全修飾ドメイン名を正しく解決できなければなりません。

配信場所

Satellite Proxy は事実上すべてのローカル HTTP 要求を Red Hat Satellite に転送するため、Satellite Proxy 上での転送が行われない場所 (`/var/www/html/pub/`) に配信用のファイル (キックスタートインストールツリー内のものなど) を置くようにしてください。このディレクトリに配置されたファイルは Satellite Proxy から直接ダウンロードすることができます。これは、GPG キーの配信やキックスタートファイル用のインストールツリーの設定時にとくに便利です。

帯域幅

ネットワークの帯域幅は、Satellite、プロキシーおよびクライアント間での通信に重要なものです。Red Hat では、高ボリュームのトラフィックに対応するために多くのシステムやクライアントにパッケージを提供できるネットワーク上の高い帯域幅の使用を推奨しています。Red Hat では指針として、あるシステムから別のシステムにパッケージを転送する際の速度ごとの推定時間を算出しています。

表2.2 帯域幅による推定時間

	単一パッケージ (10Mb)	マイナーリリース (750Mb)	メジャーリリース (6Gb)
256Kbps	5 分 27 秒	6 時間 49 分 36 秒	2 日と 7 時間 55 分
512Kbps	2 分 43.84 秒	3 時間 24 分 48 秒	1 日と 3 時間 57 分
T1 (1.5Mbps)	54.33 秒	1 時間 7 分 54.78 秒	9 時間 16 分 20.57 秒
10Mbps	8.39 秒	10 分 29.15 秒	1 時間 25 分 53.96 秒
100Mbps	0.84 秒	1 分 2.91 秒	8 分 35.4 秒
1000Mbps	0.08 秒	6.29 秒	51.54 秒

Red Hat では、マイナーおよびメジャーリリースに関しては、少なくとも 100Mbps のネットワーク速度を推奨しています。これにより 10 分以上の転送におけるタイムアウトを避けることができます。ネットワーク速度はお使いのネットワーク設定によって変わることがあります。

さらに Red Hat はコードを実行しているシステムが一般に利用できないようにすることを推奨しています。システム管理者以外のユーザーがこれらのマシンへのシェルアクセスを持つことのないようにしてください。不要なサービスはすべて無効にしてください。サービスを無効にするには、`ntsysv` または `chkconfig` を使用します。

第3章 RED HAT SATELLITE PROXY のインストール

本章では、Red Hat Satellite Proxy Server のインストールおよび基本設定について説明します。ここでは、[2章 要件](#)に記載されている必要条件を満たしているものと仮定しています。ただし、Red Hat Satellite Proxy Server の最新バージョンに [アップグレード](#) する場合は、[7章 Red Hat Proxy Server インストールのアップグレード](#)を参照してください。

3.1. インストール手順の要約

Red Hat Satellite Proxy を機能的に運用するには、ソフトウェアのインストールの他にも必要な事項がいくつかあります。クライアントシステムには、これを使用する設定が必要になり、Satellite Proxy ではカスタムリポジトリのセットアップが必要になることがよくあります。このセクションでは手順一覧の概要を示します。

1. Red Hat Satellite サーバーで Red Hat Satellite Proxy エンタイトルメントを取得し、インストールします。これには、Red Hat カスタマーポータルで更新されたマニフェストを作成することが含まれます。[「Satellite Proxy エンタイトルメントの取得」](#)を参照してください。
2. Red Hat Satellite サーバーで更新されたマニフェストをアップロードし、Satellite Proxy の必要なチャンネルを同期します。[「Satellite Proxy エンタイトルメントの Satellite へのアップロード」](#)を参照してください。
3. Satellite Proxy ホストとして使用するために新規の Red Hat Enterprise Linux システムを作成するか、またはこれをプロビジョニングし、これを Satellite に登録します。[「オペレーティングシステムの Satellite Proxy ホストへのインストール」](#)を参照してください。
4. Satellite Proxy ホストに必要なパッケージをインストールします。[「Red Hat Satellite Proxy Server パッケージのインストール」](#)を参照してください。
5. `configure-proxy.sh` インストールスクリプトを実行します。[「Red Hat Satellite Proxy インストールスクリプトの実行」](#)を参照してください。
6. 必要な設定後のタスクを実行します。[「インストール後のタスクの実行」](#)を参照してください。

3.2. SATELLITE PROXY エンタイトルメントの取得

Satellite Proxy Server は、Satellite サーバーを介したパッケージベースのインストールです。最初の手順として、Proxy エンタイトルメントを Satellite サーバーに追加します。これには、以下の手順で生成する更新されたマニフェストが必要になります。

1. [カスタマーポータル](#)にログインし、[サブスクリプション](#)をクリックします。
2. **Satellite 組織** をクリックします。
3. **Satellite** をクリックします。
4. 必要な Satellite を選択します。
5. **Attach a subscription (サブスクリプションの割り当て)** をクリックします。
6. 既存のサブスクリプションのほかに、**Red Hat Satellite Proxy** 製品が含まれるサブスクリプションを選択します。作成するプロキシの数に対応する必要数を指定し、**ATTACH SELECTED (選択項目の割り当て)** をクリックします。該当バージョンの Red Hat Enterprise

Linux サブスクリプションがすでに割り当てられていることを確認します。サブスクリプションの割り当てには数分の時間がかかる場合があります。

7. **Attach a subscription (サブスクリプションの割り当て)** ウィンドウが表示されたら、**Close (閉じる)** をクリックします。
8. **Download manifest (マニフェストのダウンロード)** をクリックし、マニフェストファイルをローカルに保存します。

更新されたマニフェストには Red Hat Satellite Proxy のエンタイトルメントが含まれるようになります。次の手順では、更新されたマニフェストを Satellite サーバーにアップロードします。

3.3. SATELLITE PROXY エンタイトルメントの SATELLITE へのアップロード

このセクションでは、更新されたマニフェストを Satellite サーバーにアップロードする方法を説明します。

1. 管理ユーザーとして Red Hat Satellite サーバーにログインします。
2. **管理** → **Red Hat Satellite 設定** → **Manifest** に移動します。
3. **参照** をクリックして、更新されたマニフェストを選択します。



注記

コマンドラインからマニフェストをアップロードするには、Satellite サーバーで以下を実行します。

```
[root@satellite ~]# rhn-satellite-activate --
manifest=manifest_file.zip
```

4. RHN Tools チャンネルと Proxy チャンネルを同期します。

```
[root@satellite ~]# cdn-sync --channel rhn-tools-rhel-x86_64-server-6
```

```
[root@satellite ~]# cdn-sync --channel redhat-rhn-proxy-5.8-server-x86_64-6
```

5. 同期が終了したら、エンタイトルメントをアクティベーションキーに追加します。**システム** → **アクティベーションキー** に移動し、**create new key (キーの作成)** をクリックします。Red Hat Enterprise Linux ベースチャンネルなどの、これらのエンタイトルメントのベースチャンネルを追加し、**プロビジョニング** エンタイトルメントを選択します。**アクティベーションキーの作成** をクリックして完了します。

Satellite サーバーには、Red Hat Satellite Proxy のインストールに必要なエンタイトルメントが含まれるようになります。生成されるアクティベーションキーにはこれらのエンタイトルメントが含まれ、このキーは Proxy ホストを必要なチャンネルに登録するために使用されます。

3.4. オペレーティングシステムの SATELLITE PROXY ホストへのインストール

Red Hat Satellite Proxy Server は Red Hat Enterprise Linux オペレーティングシステム上で実行するように設計されています。最初の手順として、ベースオペレーティングシステムをインストールします。詳細は、[Red Hat Enterprise Linux 6 Installation Guide](#) [1] を参照してください。



注記

Satellite サーバーに Red Hat Enterprise Linux 6 のベースチャンネルがある場合、マシンのキックスタートを開始して Red Hat Enterprise Linux 6 をインストールできます。キックスタートについての詳細は、『Red Hat Satellite スタートガイド』を参照してください。

オペレーティングシステムのインストール中およびインストール後には以下のタスクを実行してください。

- 前述のハードウェア要件に従い、パッケージの格納に使用するパーティションに十分な領域を割り当てます。Red Hat パッケージがキャッシュされるデフォルトの場所は `/var/spool/squid` で、カスタムパッケージは `/var/spool/rhn-proxy` になります。



注記

インストールプログラムは、`/var/spool/squid` がマウントされるパーティションで利用可能な容量を自動的に計算し、Satellite Proxy の使用に空き容量の最大 60 パーセントを割り当てます。

- ファイアウォールのルールが「[追加要件](#)」に記載されている要件を満たすように更新されていることを確認します。iptables 設定を変更し、サービスを起動します。
- Red Hat Satellite Proxy Server で必要とされるパッケージをインストールします。



重要

ベースの Satellite Proxy パッケージのみをインストールしてください。追加のパッケージのインストールは Satellite Proxy のインストールの失敗の原因になる可能性があります。

Red Hat Enterprise Linux の各バージョンで必要とされる適切なパッケージグループを取得する方法については、「[ソフトウェア要件](#)」を参照してください。

- **NTP (Network Time Protocol)** を有効にしてから Satellite Proxy とすべてのクライアントシステム上で該当のタイムゾーンを選択します。

ベースオペレーティングシステムのインストールを完了したら、root ユーザーとして Satellite Proxy ホストにログインし、`rhnreg_ks` コマンドを使用してシステムを Red Hat Satellite Server に登録します。以下は例になります。

```
[root@satproxy ~]# rhnreg_ks --
serverUrl=http://satellite.example.com/XMLRPC --activationkey=1-
123456789abcdef123456789abcdef12
```

3.5. RED HAT SATELLITE PROXY SERVER パッケージのインストール

Satellite Proxy のインストールの手順を以下に説明します。

1. インストールする Satellite Proxy システムに、root ユーザーとしてログインします。
2. クライアントが必要なチャンネルにサブスクライブされるようにします。

```
[root@satproxy ~]# spacewalk-channel --add -c rhn-tools-rhel-x86_64-server-6 --user admin --password adminpassword
```

3. Provisioning エンタイトルメントをシステムに割り当てます。
 - a. Satellite サーバー上で、**システム** をクリックします。
 - b. Satellite Proxy の名前をクリックします。
 - c. **プロパティ** タブをクリックします。
 - d. 付属エンタイトルメント セクションで **プロビジョニング** を選択します。
 - e. **プロパティの更新** をクリックします。
4. Satellite Proxy インストールパッケージをインストールします。このパッケージには、実際の Satellite Proxy のインストールをガイドするメインスクリプトが含まれます。

```
[root@satproxy ~]# yum install spacewalk-proxy-installer
```

3.6. RED HAT SATELLITE PROXY インストールスクリプトの実行

コマンドラインのインストールプログラムは、実際の Satellite Proxy のインストールプロセスをガイドします。このプログラムでは、インストールオプションや SSL 証明書の生成などの Satellite Proxy インストールおよび初期設定の詳細について一連のプロンプトが表示されます。このステップを実行するにはサーバーへの Root アクセスが必要です。

重要

インストールスクリプトを実行するには、Satellite Proxy に Satellite Server からの SSL ファイルが必要になります。**/root/ssl-build** ディレクトリーを作成します。

```
[root@satproxy ~]# mkdir /root/ssl-build
```

必要な Red Hat Satellite から公開証明書と CA 証明書を新規ディレクトリーにコピーします。

```
[root@satproxy ~]# scp 'root@satellite.example.com:/root/ssl-build/{RHN-ORG-PRIVATE-SSL-KEY,RHN-ORG-TRUSTED-SSL-CERT,rhn-ca-openssl.cnf}' /root/ssl-build
```

または、インストールを実行する際に **--force-own-ca** オプションを追加します。

次のインストールスクリプトを実行して Red Hat Satellite Proxy Server をインストールします。

```
# configure-proxy.sh
```



注記

プロンプトの表示で **Enter** キーを押すと、プログラムは括弧で囲まれたデフォルトの応答を使用します。別の方法としては、ユーザーの対話をまったく使わずにデフォルトの回答を使用したい場合は、インストールスクリプトと共に **--non-interactive** オプションを使用します。

Satellite Proxy Server のインストール情報の生成

インストールスクリプトは、インストールを実行しているマシンに固有の Satellite Proxy インストールについての詳細情報を要求します。

RHN Parent (親 RHN)

コンテンツを Satellite Proxy に提供するシステムの Satellite Server ドメイン名またはアドレスです。

CA Chain (CA チェーン)

Enter キーを押して、認証局 (CA) チェーンのデフォルトパスを使用します。

Satellite Proxy が Red Hat Satellite と通信している場合、この値は通常 **/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT** になります。そうでない場合、カスタム SSL 証明書を **/usr/share/rhn/** ディレクトリー内で見つける必要があります。

Satellite Proxy version to activate (アクティベートする Satellite Proxy のバージョン)

インストールする Satellite Proxy のバージョンを確認するように求められます。

HTTP Proxy (HTTP プロキシ)

Satellite Proxy サーバーが HTTP プロキシを使用して接続する場合は、*corporate.proxy.example.com:3128* のように、プロキシのホスト名とポート番号を入力します。

Traceback email (トレースバックメール)

エラー関連のトレースバックメッセージが送信されるメールアドレスのコンマで区切られている一覧です。通常は Satellite Proxy 管理者のメールです。

Use SSL (SSL の使用)

Enter キーを押すか、または **y** を入力し、SSL を使用するように Satellite Proxy を設定します。

SSL Certificate (SSL 証明書)

有効な SSL サーバー証明書を生成するのに必要な詳細を入力します。[例3.1「SSL 証明書の生成例」](#) は、この証明書を生成する方法を示しています。

例3.1 SSL 証明書の生成例

```
Regardless of whether you enabled SSL for the connection to the
Satellite Proxy Parent
Server, you will be prompted to generate an SSL certificate.
This SSL certificate will allow client systems to connect to this
Satellite Proxy
securely. See the Satellite Proxy Installation Guide for more
information.
```

```

Organization: Example Company
Organization Unit [proxy1.example.com]:
Common Name: proxy1.example.com
City: New York
State: New York
Country code: US
Email [admin@example.com]:

```

インストールスクリプトが複数セットのパッケージをインストールし、それらをインストールするためのパーミッションを要求します。プロンプトが出されるたびに **Y** を選択します。

3.7. インストール後のタスクの実行

主な Satellite Proxy インストールタスクが完了すると、インストールプログラムは、設定チャンネルの作成、および変更されたデーモンの再起動などの、数多くの追加タスクを実行します。

SSL の設定

次に **configure-proxy.sh** プログラムによって SSL を設定します。この場合、SSL キーと公開証明書を生成する前に証明書認証局のパスワードを作成し、これを確認するよう求められます。

例3.2 認証局 (CA) キーと公開証明書の生成例

```

Generating CA key and public certificate:
CA password: *****
CA password confirmation: *****

```

設定チャンネルの作成

インストールプログラムは、**configure-proxy.sh** の実行中に作成された設定ファイルに基づいて設定チャンネルを作成するかをどうか確認することも要求します。

インストールプログラムは、システムの名前 (**sysID**) に基づいて Satellite Server 設定チャンネルを作成します。ここでは、Satellite Proxy がインストールされ (以下の例では **sysID** は 1000010000)、Satellite Proxy Server の設定チャンネルを構成する **httpd**、**SSL**、**squid**、および **jabberd** サーバーなどの各種ファイルが収集されます。

この設定例は [例3.3 「設定チャンネルの作成例」](#) に示されています。

例3.3 設定チャンネルの作成例

```

Create and populate configuration channel rhn_proxy_config_1000010000?
[Y]:
Using server name satellite.example.com
Red Hat Network username: admin
Password:
Creating config channel rhn_proxy_config_1000010000
Config channel rhn_proxy_config_1000010000 created
using server name satserver.example.com
Pushing to channel rhn_proxy_config_1000010001:
Local file /etc/httpd/conf.d/ssl.conf -> remote file
/etc/httpd/conf.d/ssl.conf

```

```

Local file /etc/rhn/rhn.conf -> remote file /etc/rhn/rhn.conf
Local file /etc/rhn/cluster.ini -> remote file /etc/rhn/cluster.ini
Local file /etc/squid/squid.conf -> remote file /etc/squid/squid.conf
Local file /etc/httpd/conf.d/cobbler-proxy.conf -> remote file
/etc/httpd/conf.d/cobbler-proxy.conf
Local file /etc/httpd/conf/httpd.conf -> remote file
/etc/httpd/conf/httpd.conf
Local file /etc/jabberd/c2s.xml -> remote file /etc/jabberd/c2s.xml
Local file /etc/jabberd/sm.xml -> remote file /etc/jabberd/sm.xml

```

サービスの再起動

インストールプロセスの最終手順として、Satellite Proxy 関連のすべてのサービスを再起動します。インストールプログラムは、このステップが完了すると終了します。

例3.4 Satellite Proxy Server 関連のすべてのサービスの再起動

```

Enabling Satellite Proxy
Shutting down rhn-proxy...
Shutting down Jabber router:                [ OK ]
Stopping httpd:                             [ OK ]
Stopping squid:                             [ OK ]
Done.
Starting rhn-proxy...
init_cache_dir /var/spool/squid... Starting squid: . [ OK ]
Starting httpd:                             [ OK ]
Starting Jabber services                    [ OK ]
Done.

```

すべてのサービスが正常に起動する場合、Satellite Proxy のインストールは正常に行われていることとなります。

3.8. SATELLITE PROXY SERVER インストールの自動化

システム上に Satellite Proxy をインストールするプロセスの一部を自動化したい場合、管理者は **configure-proxy.sh** スクリプトを使用してインストールプログラムのプロンプトに対して事前に指定された応答を含む *回答ファイル (answer files)* を作成できます。

以下は、バージョン番号、親サーバーとして機能する Satellite Server、**SSL** および他の設定パラメータに関する事前指定の回答を含む回答ファイルの例です。回答ファイルの作成と使用の詳細については、**configure-proxy.sh** man ページ (**man configure-proxy.sh**) を参照してください。

```

# example of answer file for configure-proxy.sh
# for full list of possible option see
# man configure-proxy.sh

VERSION=5.8
RHN_PARENT=satellite.example.com
TRACEBACK_EMAIL=jsmith@example.com
USE_SSL=1
SSL_ORG="Red Hat"

```

```
SSL_ORGUNIT="Satellite"  
SSL_CITY=Raleigh  
SSL_STATE=NC  
SSL_COUNTRY=US  
ENABLE_SCOUT=N  
CA_CHAIN=/usr/share/rhn/RHN-ORG-TRUSTED-SSL-CERT  
POPULATE_CONFIG_CHANNEL=Y
```

configure-proxy.sh スクリプトで **--answer-file** オプションを使用し、以下の例にあるように Satellite Proxy のインストールを自動化するために回答ファイル (answer file) を使用します。サンプルの **answers_file.txt** ファイルの名前を回答ファイル (answer file) のパスに置き換えます。

```
# configure-proxy.sh --answer-file=/path/to/answers_file.txt
```

[1] https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/6/html-single/installation_guide/index

第4章 CUSTOM CHANNEL PACKAGE MANAGER

このセクションでは、Red Hat Satellite Proxy を Custom Channel Package Manager と併用する方法について説明します。

4.1. CUSTOM CHANNEL PACKAGE MANAGER の使用および RED HAT NETWORK PROXY によるローカルパッケージの提供

Custom Channel Package Manager はコマンドラインツールであり、これを使用して組織からプライベート Red Hat Satellite チャンネルに関連付けられたローカルパッケージを Red Hat Satellite Proxy Server 経由で提供することができます。Red Hat Satellite Proxy Server の公式 Red Hat パッケージのみを更新する場合は、Custom Channel Package Manager をインストールしないでください。

Custom Channel Package Manager を使用するには、**spacewalk-proxy-package-manager** パッケージとその依存関係をインストールします。

パッケージのヘッダー情報のみを Red Hat Satellite Server にアップロードします。このヘッダーは、Red Hat Satellite がクライアントシステムのパッケージの依存関係を解決するために必要になります。実際のパッケージファイル (*.rpm) は Red Hat Satellite Proxy Server に格納されます。

Custom Channel Package Manager は、`/etc/rhn/rhn.conf` 設定ファイルで定義される Proxy の設定と同じものを使用します。詳細については、`rhn_package_manager` の man ページを参照してください。

Custom Channel Package Manager によるローカルパッケージの提供を可能にするには、以下の手順が必要になります。

1. プライベートチャンネルを作成する。
2. ローカルパッケージをそのチャンネルにアップロードする。

これらの手順は、次のセクションで説明します。

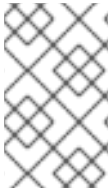
4.1.1. プライベートチャンネルの作成

Red Hat Satellite Proxy Server からローカルパッケージの提供を行う前に、ローカルパッケージを格納するプライベートチャンネルが必要になります。次の手順でプライベートチャンネルを作成します。

1. ネットワーク内のローカル Red Hat Satellite サーバーにログインします。
2. 上部ナビゲーションバーの**チャンネル** をクリックします。左にあるナビゲーションバーに**チャンネルの管理 (Manage Channels)** オプションが表示されない場合には、このユーザーにチャンネルの編集パーミッションがあるかどうかを確認してください。上部ナビゲーションバーから**ユーザー** カテゴリにアクセスしてこれを確認することができます。
3. 左にあるナビゲーションバーで、**ソフトウェアチャンネルの管理** をクリックしてからページの右上部にある**create new channel (チャンネルの作成)** ボタンをクリックします。
4. 親チャンネルとベースチャンネルのアーキテクチャーを選択して、新しいプライベートチャンネルの名前、ラベル、概要、説明を入力します。チャンネルラベルは少なくとも6文字以上の長さで英文字で始まらなければなりません。使用できるのは小文字、数字、ダッシュ(-)、ピリオド(.)のみになります。また、チャンネルの GPG キーの URL も入力します。ただし、このフィールドは必須ではありませんが、セキュリティ強化のため入力推奨されるフィールドです。

5. **チャンネルの作成** をクリックします。

4.1.2. パッケージのアップロード



注記

プライベート Red Hat Satellite チャンネルにパッケージをアップロードするには、組織管理者である必要があります。スクリプトにより Red Hat Satellite の認証情報の入力が必要と求められます。

プライベートチャンネルの作成後に、Red Hat Satellite Server にバイナリー RPM とソース RPM のパッケージヘッダーをアップロードして、パッケージを Red Hat Satellite Proxy Broker Server にコピーします。バイナリー RPM のパッケージヘッダーをアップロードするには、以下のコマンドを実行します。

```
rhn_package_manager -c "label_of_private_channel" pkg-list
```

このコマンドはパッケージのヘッダーを指定されたチャンネル名にアップロードし、パッケージそのものを `/var/spool/rhn-proxy/rhn` にアップロードします。

pkg-list はアップロードするパッケージ一覧です。代わりに、**-d** オプションを使ってチャンネルに追加するパッケージを含むローカルディレクトリーを指定することもできます。このディレクトリーには含まれるべきパッケージ以外のファイルがないことを確認してください。Custom Channel Package Manager は標準入力からパッケージ一覧を読み取ることもできます (**--stdin** を使用)。

ソース RPM のパッケージヘッダーをアップロードするには、以下を実行します。

```
rhn_package_manager -c "label_of_private_channel" --source pkg-list>
```

複数のチャンネルを指定している場合 (**-c** または **--channel** を使用)、アップロードしたパッケージヘッダーはリストされるチャンネルすべてにリンクされます。



注記

チャンネル名が指定されていない場合、パッケージはどのチャンネルにも追加されません。この場合、パッケージは Red Hat Satellite Web インターフェースを使ってチャンネルに追加できます。インターフェースは既存のプライベートチャンネルの変更にも使用できます。

パッケージをアップロードしたら、すぐに Red Hat Satellite Web インターフェースを使ってパッケージがあることを確認することができます。上部ナビゲーションバーの**チャンネル** をクリックして、左に表示されるナビゲーションバーの**ソフトウェアチャンネルの管理** をクリックし、カスタムチャンネルの名前をクリックします。**パッケージ** サブタブをクリックすると、各 RPM が一覧表示されます。

さらに、以下のコマンドラインでローカルディレクトリーがチャンネルの Red Hat Satellite Server のイメージと同期しているかどうかを確認してください。

```
rhn_package_manager -s -c "label_of_private_channel"
```

-s オプションは、欠けているパッケージすべてを一覧表示します (Red Hat Satellite Server にアップロードされたがローカルディレクトリーに表示されないもの)。このコマンドを使用するには、組織管理者である必要があります。スクリプトにより Red Hat Satellite ユーザー名とパスワードの入力が求め

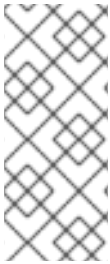
られます。

ローカルパッケージの更新に Custom Channel Package Manager を使用している場合、そのシステムをプライベートチャンネルにサブスクライブさせるには Red Hat Satellite Web サイトに移動する必要があります。

4.2. プロキシのプリキャッシュ設定

Proxy サーバーは RPM ファイルのプリキャッシュまたはミラー化を実行できます。これは、RPM ファイルが Proxy サーバーからクライアントに直接送信されることを意味します。クライアントはファイルが Satellite サーバーから Proxy サーバーにダウンロードされるのを待機する必要はなく、ファイルはクライアントに送信されます。Proxy サーバーは **yum** および **anaconda** (キックスタートのインストールおよびプロビジョニングの場合) の RPM 要求を認識します。詳細は、**rhn_package_manager** の man ページを参照してください。

Proxy のプリキャッシュは、Satellite サーバーへのネットワーク接続が遅い場合や、帯域幅が Premium レベルの場合などにとくに役立ちます。**rhn_package_manager** コマンドを使用して RPM ファイルを Proxy サーバーのキャッシュに手動で読み込んだり、**rsync** コマンドを使用する **cron** ジョブを作成してタスクを自動的に実行したりできます。



注記

Proxy サーバーのプリキャッシュ機能を使用するには、必要な RPM ファイル用に利用可能なディスク領域が常になければなりません。プリキャッシュに対応しない Proxy サーバーの場合には要求された RPM ファイルが存在するのは期限切れになるまでとなりますが、この機能を使用すると、プリキャッシュされた RPM ファイルは使用されるかどうかにかかわらず Proxy サーバー上で無期限に残ります。

4.2.1. RPM ファイルの Proxy キャッシュへの手動読み込み

Satellite Proxy および **rhn_package_manager** は不必要なキャッシュの競合を避けるために更新されています。既存の **rhn_package_manager --copyonly** コマンドを使用してキャッシュを設定することができます (そのオプションのエイリアスがよりユーザーフレンドリーな名前の **--cache-locally** で追加されています)。**rhn_package_manager** に加えられたもう 1 つの大きな変更として、このコマンドにより、たとえば **rhn-satellite-exporter** ユーティリティを使用して Satellite サーバーで作成できる可能姓のあるチャンネルエクスポートからパッケージを読み取ったり、インポートしたりできる点があります。これは、ディレクトリーのすべての RPM ファイルをインポートするための **--dir** オプションや、コマンドラインで指定される RPM ファイルの一覧を含む **rhn_package_manager** が RPM ファイルのインポートに使用できる他のメソッドに追加される機能になります。

以下の例は、Satellite サーバーからチャンネルエクスポートを使用して、*my-channel-1* チャンネルにある RPM ファイルのみをキャッシュする方法を示しています。このエクスポートには Satellite サーバーのすべてのチャンネルが含まれ、これは **/mnt/export** にマウントされます。

```
# rhn_package_manager --cache-locally --from-export /mnt/export --channel
my-channel-1
```

エクスポートに含まれるすべてのチャンネルの RPM ファイルすべてをインポートするには、**--channel** オプションを省略します。

```
# rhn_package_manager --cache-locally --from-export /mnt/export
```

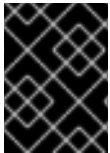

チャンネルエクスポートが複数の ISO イメージに分散する場合、`rhn_package_manager` コマンドを実行する前にそれらのイメージを Proxy 上でローカルにまとめる必要はありません。1 度に 1 つずつイメージをマウントし、それぞれに同じコマンドを実行します。

4.2.2. RPM ファイルの Proxy キャッシュへの自動読み込み

初期のクローンを設定する際に、RPM ファイルを含むチャンネルエクスポートやディレクトリーを使用すると便利であり、かつ速く処理することができますが、その場合 Satellite サーバーが新規コンテンツで更新されるたびに、新規 RPM ファイルを含む新規のチャンネルエクスポートまたはディレクトリーを作成しなければなりません。`rsync` コマンドを使用して cron ジョブや同様のジョブを設定して、更新された RPM ファイルを Proxy キャッシュにダウンロードすることができます。この場合、Satellite サーバー上のすべての RPM ファイルを Proxy にダウンロードする必要があると仮定されます。Proxy サーバーでは root で cron ジョブおよび `rsync` コマンドを実行する必要がありますが、その後 Satellite サーバーには RPM ストアへの読み取りアクセス (すべてのユーザーが持つアクセス) のあるユーザーとしてログインできます。

Satellite への自動接続を有効にする SSH キーペアの設定

Proxy サーバーの `root` ユーザーが Pパスワードを入力せずに Satellite サーバーの `$user` への SSH 接続を確立できるようにするには、SSH キーペアをセットアップする必要があります。



重要

この方法では秘密鍵にパスワードを設定しないようにする必要があります。このキーの安全を確保するようにしてください。

Proxy サーバーで `root` として以下のコマンドを実行します。

```
ssh-keygen -t rsa -N '' -f /root/.ssh/id_dsa \
&& cat /root/.ssh/id_dsa.pub | ssh user@satellite.example.com 'cat >>
.ssh/authorized_keys'
```

プロンプトが出されたら `user` のパスワードを入力します。これで、パスワードを入力せずに Satellite サーバーへの SSH 接続を作成できるようになります。

```
# ssh user@satellite.example.com
```

自動同期のセットアップ

ここでは適切な `rsync` コマンドに関連して `cron` ジョブを追加する必要があります。たとえば、`root` として Proxy で以下を実行します。

```
crontab -e
1 3 * * * /usr/bin/rsync -r
user@satellite.example.com:/var/satellite/redhat/* /var/spool/rhn-
proxy/rhn
# write and quit with :wq
```

この設定により、`rsync` コマンドは毎日午前 3:01 に実行され、Satellite サーバー上にあるすべての RPM ファイルが Proxy キャッシュにダウンロードされます。`/var/satellite` ディレクトリーはデフォルトの Satellite `root` ディレクトリーですが、これは設定可能なディレクトリーです。Satellite インストール上でこれを変更した場合、このコマンドを適宜調整します。

第5章 CNAME レコードを使用するための SATELLITE PROXY の設定

Red Hat Satellite Proxy の特長の 1 つに、正規のホスト名の代わりに CNAME レコードまたはエイリアスを利用できる機能があります。この機能は、その標準のホスト名や設定を使用して Red Hat Satellite Proxy と通信することを妨げるネットワーク関連の問題がある場合に役立ちます。

5.1. 前提条件

Red Hat Satellite Proxy Server インストールで CNAME レコードを使用するには、以下を実行する必要があります。

- 必要な CNAME レコードがサーバー用に設定されていることを確認する。
- CNAME レコードを Satellite Proxy サーバー設定に追加する。
- 複数ホスト対応の証明書を新たに作成し、Satellite Proxy にインストールする。

5.2. CNAME レコードの SATELLITE PROXY SERVER 設定への追加

このセクションでは、Satellite Proxy サーバーの CNAME レコードを Satellite Proxy サーバー設定に追加し、Satellite Proxy の Web インターフェース内でそれらの CNAME を有効なオプションとして認識させる方法について説明します。



注記

本書では、Satellite Proxy サーバーがインストールされているマシン用に DNS CNAME レコードをセットアップする方法については説明しません。必要な CNAME レコードが正しくセットアップされていることを確認するには、システム管理者または DNS 管理者にお問い合わせください。

手順5.1 CNAME レコードを使用するために Satellite Proxy Server を設定する

1. Proxy サーバーのシステム ID を判別します。この値は、Proxy サーバー上の `/etc/sysconfig/rhn/systemid` ファイルにあります。このファイルで以下のスタanzas を見つけてください (実際のシステム ID はこの例とは異なります)。

```
<member>
<name>system_id</name>
<value><string>ID-1036997498</string></value>
</member>
```

システム ID は `<string>` タグ内にあります。実際のシステム ID の一部として「ID-」を含めないでください。

2. 以下の行を `/etc/rhn/rhn.conf` ファイルに追加します。 `systemID` を直前の手順の値に置き換えます。

```
valid_cnames_sytemID= cname1,cname2,cname3
```

3. Tomcat サービスを再起動するために次のコマンドを実行します。

```
# service rhn-proxy restart
```

Tomcat サービスの再起動後に、Satellite Proxy サーバーの Web インターフェースを更新します。システムの詳細 ページの ハードウェア タブに CNAME の一覧が表示されるはずですが、

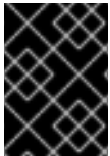
これらの CNAME を使用できるようにするには、一連の証明書を新たに作成し、これらの証明書を使用できるように Satellite Proxy を設定する必要があります。元の証明書は正規のホスト名にのみ有効であり、それぞれの CNAME に有効な証明書を新たに作成する必要があります。これについては、「[複数ホスト対応の SSL 証明書の生成と使用](#)」で説明されています。

5.3. 複数ホスト対応の SSL 証明書の生成と使用

Satellite Proxy サーバーで CNAME レコードを使用する機能を利用できるようにするには、複数ホスト対応の SSL 証明書を生成する必要があります。また、Satellite Proxy サーバーがこれらの証明書を認識し、使用できるように `rhncacert/openssl.cnf` ファイルを更新する必要があります。

手順5.2 複数ホスト対応の証明書を使用できるように SSL 設定ファイルを更新する

1. `/root/ssl-build/rhncacert/openssl.cnf` ファイルを編集し、`[CA_default]` セクションの場所を特定します。
2. エントリー `copy_extensions = copy` が存在し、コメントアウトされていないことを確認します。
3. ファイルを保存してから閉じます。



重要

上記のステップが完了してから、`SSL_CNAME` を設定して `configure-proxy.sh` を実行する必要があります。そうしないと、インストールが失敗します。

以前に作成した新規の SSL 証明書を Satellite Proxy 設定で使用できるようにするには、[回答 \(answers\)](#) ファイルを更新する必要があります。

手順5.3 複数ホスト対応の証明書を使用できるように回答ファイル (Answers File) を更新する

1. 初期の Satellite Proxy インストールで作成した `answers.txt` ファイルを編集します。ファイルを作成しなかった場合は、`/usr/share/doc/spacewalk-setup-<version>/answers.txt` でサンプルのセットアップを見つけることができます。
2. 次の行が存在し、コメントアウトされていないことを確認します。

```
SSL_CNAME = (cname01 cname02 cname03)
```

3. `configure-proxy.sh` スクリプトを `--answer-file` オプションを指定して実行し、複数ホスト対応の SSL 証明書を生成します。以下が例になります。

```
# configure-proxy.sh --answer-file=</path/to/answers.txt>
```



注記

`configure-proxy.sh` スクリプトを複数回実行し、必要に応じて設定のテストと更新を実行できます。

第6章 SATELLITE PROXY SERVER の負荷分散

一部の環境には、Satellite 要求のロードを分散したり、要求を元のクライアントにより近い場所にリダイレクトするために Satellite クライアントと Satellite Proxy 間のロードバランサーが含まれます。Satellite Proxy トポロジーがより複雑になり、CNAME サポート、チェーンなどが含まれる場合、ロードバランサー間で交換される HTTP ヘッダー要求やラウンドロビンプロキシチェーンをテストすることができます。本章では、Squid をリバースプロキシとして設定して、2つの Satellite プロキシ間でのラウンドロビン要求を実行する方法を説明します。ここでは、セットアップ手順や非 SSL および SSL プロキシ要求の両方をサポートする方法についても扱います。

以下の例の環境では 5 つの異なるホストを使用します。

- Red Hat Satellite Proxy A: IP アドレス **192.168.100.16** およびホスト名 **proxya.example.com** で表されます。
- Red Hat Satellite Proxy B: IP アドレス **192.168.100.17** およびホスト名 **proxyb.example.com** で表されます。
- ロードバランサー: ホスト名 **lb.example.com** で表されます。
- Red Hat Satellite Proxy A および B が接続された Red Hat Satellite Server
- クライアントマシン: IP アドレス **192.168.100.19** で表されます。

6.1. PROXY サービスのロードバランサーへのインストール

ロードバランサーには、一部の Red Hat Satellite Proxy と同じサービスが必要になります。以下のコマンドを使用してこれらのサービスをインストールし、設定します。

```
# yum install spacewalk-proxy-installer -y
# configure-proxy.sh
```



重要

ロードバランサーを Red Hat Satellite 5 サーバーに登録し、**rhn-tools-rhel-x86_64-server-6** チャンネルにサブスクライブさせて **spacewalk-proxy-installer** パッケージにアクセスします。

インストールが完了したら、**httpd** パッケージおよびその依存関係をアンインストールします。

```
# yum remove httpd -y
```

これにより、ロードバランシングサービスを設定するためのサービスの準備が整います。

6.2. SQUID リバースプロキシのインストール

Squid サーバーをインストールし、リバースプロキシモードを使用してロードバランサーとして使用します。

```
# yum install squid
```

SSL 証明書を生成し、Satellite CA でそれらに署名する必要もあります。最も簡単な方法としては、CA がすでに利用可能であるため、Satellite サーバーで **rhn-ssl-tool** を使用し、サーバー証明書を生成することができます。

Satellite SSL Maintenance Tool (**rhn-ssl-tool**) は Satellite SSL キーおよび証明書を生成し、これらを保持します。また、これらのキーおよび証明書をデプロイする際に使用する RPM も生成します。このツールは Satellite 環境での使用が意図されていますが、Satellite 外でも使用することができます。

この例では、ロードバランサーの名前は **lb.example.com** です。ホスト名をデプロイメントに適用される名前に置き換え、適切なビルドディレクトリーを入力します。以下のコマンドを Satellite サーバーで実行します。

```
$ rhn-ssl-tool --gen-server --set-hostname=lb.example.com -d /root/ssl-build
```

上記で使用される **rhn-ssl-tool** は **lb.example.com** の SSL ファイルを作成し、これらのファイルを **/root/ssl-build** ディレクトリーに保存します。**server.crt**、**server.key**、および **RHN-ORG-TRUSTED-SSL-CERT** CA 証明書を、**dhcp** ディレクトリーから **lb.example.com** ロードバランサーにコピーします。これらのファイルは、実際のロードバランサーの SSL をセットアップするために使用されます。**RHN-ORG-TRUSTED-SSL-CERT** 証明書により、ロードバランサーとプロキシ間の SSL 通信が可能になります。

lb.example.com サーバーで **/etc/squid/squid.conf** ファイルを変更し、リバースプロキシモードをセットアップします。

例6.1 リバースプロキシモードのセットアップ

```
#
# SSL configuration
#

# Ensure you enter each configuration directive on a single line

acl is_ssl port 443

https_port 443 cert=/etc/pki/tls/certs/lb.crt
key=/etc/pki/tls/certs/lb.key accel vhost name=proxy_ssl

cache_peer proxya.example.com parent 443 0 no-query originserver round-robin
ssl name=proxya.example.com sslcafile=/etc/pki/tls/certs/squid-ca.crt

cache_peer proxyb.example.com parent 443 0 no-query originserver round-robin
ssl name=proxyb.example.com sslcafile=/etc/pki/tls/certs/squid-ca.crt

cache_peer_access proxya.example.com allow is_ssl
cache_peer_access proxya.example.com deny !is_ssl
cache_peer_access proxyb.example.com allow is_ssl
cache_peer_access proxyb.example.com deny !is_ssl

http_access allow is_ssl

#
# Non-SSL configuration
```

```
#
# Ensure you enter each configuration directive on a single line

acl nonssl port 80

http_port 80 accel name=proxy_nonssl defaultsite=dhcp16.example.com

cache_peer 192.168.100.16 parent 80 0 no-query name=proxy_nonssl
originserver

cache_peer_access proxy_nonssl allow nonssl
cache_peer_access proxy_nonssl deny !nonssl

http_access allow nonssl

sslpasword_program /bin/password.out
forwarded_for on
```

直前の例では、2つのリバースプロキシのセットアップについて説明しています。ポート 443 にはラウンドロビンモードで使用される2つのプロキシがあります。要求は2つのプロキシ間で均等に共有されます。**server.crt** および **server.key** ファイルの名前はそれぞれ **lb.crt** および **lb.key** (「load balancer (ロードバランサー) の省略)に変更され、識別が容易になりました。Satellite CA 証明書の名前は **squid-ca.crt** に変更されます。**cache_peer sslcfile** オプションはこのファイルを参照します。

証明書を **squid** グループに追加します。

```
# chgrp squid /etc/pki/tls/certs/{lb.crt,lb.key,squid-ca.crt}
```

ファイルの詳細は以下のように表示されます。

```
-rw-r--r--. 1 root squid 5450 Aug 23 21:23 lb.crt
-rw-r--r--. 1 root squid 1675 Aug 23 21:23 lb.key
-rw-r--r--. 1 root squid 5363 Aug 22 14:19 squid-ca.crt
```

cache_peer ディレクティブはラウンドロビン形式で使用される2つのプロキシをセットアップします。ロードバランサーがプロキシと通信できるように CA 証明書を指定する必要があることに注意してください。また squid の **acl is_ssl** および **cache_peer** ディレクティブを使用して、ポート 443 のトラフィックのみがこれらのプロキシにヒットするように設定されます。

defaultsite ディレクティブを使用して、ポート 80 のすべてのトラフィックが1つのプロキシにリダイレクトされ、dhcp16.example.com プロキシにデフォルト設定されます。Acls は ssl ポートと同様にセットアップされます。

sslpasword_program ディレクティブを使用すると、SSL キーのパスフレーズ (使用される場合、完了についての表示が出されます) を人の介入なしに起動時に squid に送信できます。password.out の内容は、SSL パスフレーズを echo する bash スクリプトです。**forwarded_for** ディレクティブは **forwarded_for** ヘッダーをプロキシに送信するようにロードバランサーを設定します。



重要

`/etc/squid/squid.conf` を編集し、squid が通常リッスンするデフォルトポートの 3128 をコメントアウトします。

```
# Squid normally listens to port 3128
# http_port 3128
```

設定の変更後に squid を再起動します。

```
# service squid restart
```

6.3. クライアントのセットアップ

ロードバランサーに適切に対応するには、クライアントで `/etc/sysconfig/rhn/up2date` ファイルを修正する必要があります。

```
serverURL[comment]=Remote server URL (use FQDN)
serverURL=https://lb.example.com/XMLRPC
```

ロードバランサーは、署名された SSL 証明書に Satellite CA 証明書を使用します。クライアント CA 証明書の値を変更する必要はありません。

6.4. 設定のテスト

このセクションでは、ロードバランサーおよびプロキシー設定の基本的なテストについて説明します。

初期テスト

ロードバランサーおよびプロキシーマシンを起動します。Satellite Proxy ログは最初の要求が送信されるまで作成されないことに注意してください (`/var/log/rhn/`)。zsh などの RPM ファイルをインストールし、削除してみてください。

ロードバランサーの `/var/log/squid/access.log` ファイルには、以下のような情報が含まれる必要があります。

```
1377540630.159
97 192.168.100.19 TCP_MISS/200 1515 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/base64
1377540630.733
529 192.168.100.19 TCP_MISS/200 1409 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxyb.example.com text/xml
1377540639.968
87 192.168.100.19 TCP_MISS/200 3742 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/xml
1377540644.273
83 192.168.100.19 TCP_MEM_HIT/200 2238956 GET
https://lb.example.com/XMLRPC/GET-REQ/rhel-x86_64-server-6/getPackage/zsh-
4.3.10-5.el6.x86_64.rpm - NONE/- application/octet-stream
1377540646.765
100 192.168.100.19 TCP_MISS/200 1515 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxyb.example.com text/base64
```

```
1377540647.291
485 192.168.100.19 TCP_MISS/200 1409 POST https://lb.example.com/XMLRPC -
ROUNDROBIN_PARENT/proxya.example.com text/xml
```

この例は、クライアント要求の均衡が2つの異なるプロキシ間で図られていることを示しています。パッケージのインストール時に複数の要求が処理されるため、単一の **yum** パッケージインストールは複数のプロキシで共有されます。クライアント IP アドレスは 192.168.100.19 であり、2つの異なる Satellite Proxy に対応する名前付きの **proxya.example.com** および **proxyb.example.com** キャッシュのピアが表示されます。

Squid ロードバランサーのログファイル

以下のログファイルを参照し、ロードバランサーのアクティビティを監視します。

- **/var/log/squid/access.log**: クライアントから送信される要求の確認用
- **/var/log/squid/cache.log**: SSL 起動の問題のデバッグおよびポート 80 および 443 のキャッシュピアの確認用

netstat コマンドを使用して squid ロードバランサーが適切なポートでリッスンしていることを確認します。

```
# netstat -tulpn | grep squid | grep tcp
tcp        0      0 :::80          :::*           LISTEN
29120/(squid)
tcp        0      0 :::443         :::*           LISTEN
29120/(squid)
```


第7章 RED HAT PROXY SERVER インストールのアップグレード

本章では、Proxy Server のインストールをアップグレードする方法について説明します。ここでは、完全に稼働中の Proxy Server と必要なエンタイトルメントがあることを前提とします。

7.1. 前提条件

Red Hat Satellite Proxy Server の最新バージョンには以下が必要になります。

- Red Hat Enterprise Linux 6 (64 ビットのみ)。
- 親 Satellite サーバーからこれまでの Proxy サーバーのシステムプロファイルを削除する。

7.2. PROXY インストールのアップグレード

1. 既存の Proxy サーバーをバックアップします。該当する場合は、SSL ビルドディレクトリーをバックアップからディレクトリー `/root/ssl-build` に戻して SSL ビルドを復元します。
2. Proxy を親 Satellite に登録します。Proxy が Red Hat Enterprise Linux Server のベースチャンネルと Red Hat Network Tools の子チャンネルの両方にサブスクライブしていることを確認します。
3. Red Hat Network Tools の子チャンネルから `spacewalk-proxy-installer` パッケージをインストールします。

```
# yum install spacewalk-proxy-installer
```

4. 「[Red Hat Satellite Proxy Server パッケージのインストール](#)」にあるように、最新バージョンのプロキシーをインストールします。



注記

Proxy サーバーが Red Hat Satellite に登録されており、かつこの Proxy が以前にカスタムチャンネルを管理していた場合、アップグレード前のバックアップからこのカスタムパッケージのリポジトリーを復元します。パーミッションと所有権も適切に設定する必要があります。

```
# chmod 0750 /var/spool/rhn-proxy  
# chown apache:apache /var/spool/rhn-proxy  
# mkdir -m 0750 -p /var/spool/rhn-proxy/list  
# chown apache:apache /var/spool/rhn-proxy/list
```

デフォルトのカスタムパッケージのリポジトリーは通常 `/var/spool/rhn-proxy` です。

5. インストール後に、サーバーを最新のエラータ更新に対してアップデートします。

```
# yum update
```

6. Proxy Server サービスを再起動して、Proxy Server の機能をテストします。

```
# /usr/sbin/rhn-proxy restart
```

付録A サンプルの **SATELLITE PROXY SERVER** 設定ファイル

Satellite Proxy の `/etc/rhn/rhn.conf` 設定ファイルを使用すると、管理者はキー設定を確立することができます。ただし、このファイルに設定エラーがあると Satellite Proxy の障害が発生する可能性があるため、変更を加える際には注意してください。

`traceback_mail` および `proxy.rhn_parent` パラメーターにとくに注意してください。詳細については、以下のサンプルとそのコメント (コメントは列 1 のハッシュマーク (#) で識別できる) を確認してください。



重要

テストを目的とする場合に、`use_ssl` オプションを `rhn.conf` に追加することができます。値を 0 に設定すると Satellite Proxy と Satellite サーバー間の SSL がオフになります。これを行うとセキュリティの危険が大幅に高まることに注意してください。デフォルト値の 1 にオプションの設定を戻して SSL を再度有効にするか、または単に設定ファイルからその行を削除してください。

```
# Automatically generated RHN Management Proxy Server configuration file.
# -----
# -----

# SSL CA certificate location
proxy.ca_chain = /usr/share/rhn/RHNS-CA-CERT

# Corporate HTTP proxy, format: corp_gateway.example.com:8080
proxy.http_proxy =

# Password for that corporate HTTP proxy
proxy.http_proxy_password =

# Username for that corporate HTTP proxy
proxy.http_proxy_username =

# Location of locally built, custom packages
proxy.pkg_dir = /var/spool/rhn-proxy

# Hostname of RHN Server or RHN Satellite
proxy.rhn_parent = satellite.example.com

# Destination of all tracebacks, etc.
traceback_mail = user0@domain.com, user1@domain.com
```

付録B 用語集

Red Hat Satellite Proxy をより良く理解するには、Red Hat Satellite の以下の用語を理解しておくことが重要になります。

チャンネル

チャンネルとはソフトウェアパッケージの一覧です。チャンネルには、ベースチャンネルと子チャンネルの 2 種類があります。ベースチャンネルは特定のアーキテクチャーと Red Hat リリースに基づくパッケージ一覧で構成されています。子チャンネルはベースチャンネルに関連するチャンネルで、追加のパッケージが入っています。

組織管理者

組織管理者とは、組織の Red Hat Satellite アカウント全体にわたる最上位レベルのコントロールを備えたユーザーロールです。このロールを持つメンバーは他のユーザー、他のシステムおよびシステムグループを組織に追加したり削除したりすることができます。Red Hat Satellite の各組織には少なくとも 1 人の組織管理者が必要です。

チャンネル管理者

チャンネル管理者とは、チャンネル管理に関する機能すべてにアクセスできるユーザーロールになります。チャンネル管理者は、チャンネルの作成とチャンネルへのパッケージの割り当てができます。組織管理者は、Red Hat Satellite web サイトの **ユーザー** タブを使って、このロールを他のユーザーに割り当てることができます。

Red Hat Update Agent

Red Hat Update Agent は、ユーザーによる更新パッケージや新規パッケージの取得やクライアントシステムへのインストールを可能にするクライアントアプリケーションです。

トレースバック

トレースバックにはエラーについての詳細が記載されています。これは、Satellite Proxy のトラブルシューティングに役立ちます。重大なエラーが発生するとトレースバックファイルが自動的に生成され、Satellite Proxy の設定ファイルで指定されたユーザーにメールでこのトレースバックファイルが送信されます。

さらに詳しくは、『Red Hat リファレンスガイド』 および Satellite Web ユーザーインターフェースにある **Help** ページを参照してください。

付録C 改訂履歴

改訂 1.1-0.1

Fri Nov 17 2017

Terry Chuang

翻訳ファイルを XML ソースバージョン 1.1-0 と同期

改訂 1.1-0

Wed Feb 1 2017

Satellite Documentation Team

Red Hat Satellite 5.8 リリース向けの初版。