



Red Hat Satellite 6.15

オフラインネットワーク環境での Satellite Server のインストール

インターネットアクセスがないネットワークに Satellite Server をインストールして
設定する

Red Hat Satellite 6.15 オフラインネットワーク環境での Satellite Server のインストール

インターネットアクセスがないネットワークに Satellite Server をインストールして設定する

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

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、オフラインネットワークでの Red Hat Satellite Server のインストール方法、初期設定の実行方法、および外部サービスの設定方法を説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	3
第1章 インストールのための環境準備	4
1.1. システム要件	4
1.2. ストレージ要件	5
1.3. ストレージのガイドライン	6
1.4. サポート対象オペレーティングシステム	7
1.5. サポート対象のブラウザ	7
1.6. ポートとファイアウォールの要件	8
1.7. クライアントから SATELLITE SERVER への接続の有効化	12
1.8. DNS 解決の検証	13
1.9. 事前定義済みプロファイルを使用した SATELLITE SERVER の調整	14
第2章 SATELLITE SERVER のインストール	17
2.1. バイナリー DVD イメージのダウンロード	17
2.2. RHEL 8 でのオフラインリポジトリを使用したベースオペレーティングシステムの設定	18
2.3. オプション: SATELLITE SERVER での FAPOLICYD の使用	19
2.4. オフラインリポジトリからの SATELLITE パッケージのインストール	19
2.5. パッケージの依存関係エラーの解決	20
2.6. CHRONYD によるシステムクロックの同期	21
2.7. ベースオペレーティングシステムへの SOS パッケージのインストール	22
2.8. SATELLITE SERVER の設定	22
2.9. サブスクリプション接続の無効化	23
2.10. RED HAT サブスクリプションマニフェストの SATELLITE SERVER へのインポート	24
第3章 SATELLITE SERVER での追加設定の実行	25
3.1. カスタム CDN からコンテンツを使用するための SATELLITE SERVER の設定	25
3.2. INTER-SATELLITE SYNCHRONIZATION の設定方法	26
3.3. ネットワーク経由でコンテンツを同期するように SATELLITE SERVER を設定する	28
3.4. エクスポートを使用してコンテンツを同期するように SATELLITE SERVER を設定する	29
3.5. キックスタートリポジトリのインポート	30
3.6. SATELLITE CLIENT 6 リポジトリの有効化と同期	39
3.7. SATELLITE SERVER でプルクライアントのリモート実行を設定する	44
3.8. ホストでの電源管理の有効化	45
3.9. DNS、DHCP、および TFTP の設定	45
3.10. SATELLITE SERVER での送信メールの設定	47
3.11. カスタムの SSL 証明書を使用した SATELLITE SERVER の設定	49
3.12. SATELLITE での外部データベースの使用	53
第4章 外部サービスを使用した SATELLITE SERVER の設定	58
4.1. 外部 DNS を使用した SATELLITE SERVER の設定	58
4.2. 外部 DHCP を使用した SATELLITE SERVER の設定	59
4.3. 外部 TFTP を使用した SATELLITE SERVER の設定	63
4.4. 外部 IDM DNS を使用した SATELLITE SERVER の設定	64
付録A RED HAT SATELLITE へのカスタム設定の適用	72
付録B PUPPET 実行で上書きされた手動変更の復元	73
付録C RED HAT CDN からコンテンツをダウンロードするように SATELLITE SERVER を戻す	74

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. [Bugzilla](#) のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

第1章 インストールのための環境準備

Satellite をインストールする前に、環境が以下の要件を満たしていることを確認する必要があります。

1.1. システム要件

ネットワーク接続されたベースのオペレーティングシステムには、以下の要件が適用されます。

- x86_64 アーキテクチャー
- Red Hat Enterprise Linux 8 の最新バージョン
- 最低 4 コア 2.0 GHz CPU
- Satellite Server が機能するには、最低 20 GB のメモリーが必要です。また、最低 4 GB のスワップ領域が推奨されます。最低値よりも少ないメモリーで実行している Satellite は正常に動作しないことがあります。
- 一意なホスト名 (小文字、数字、ドット (.)、ハイフン (-) を使用できます)
- 現在の Red Hat Satellite サブスクリプション
- 管理ユーザー (root) アクセス
- 完全修飾ドメイン名を使用した完全な正引きおよび逆引きの DNS 解決

Satellite は **UTF-8** エンコーディングのみをサポートします。地域が米国で言語が英語の場合、システム全体のロケール設定として **en_US.utf-8** を設定します。Red Hat Enterprise Linux でのシステムロケールの設定に関する詳細は、[Configuring System Locale guide](#) を参照してください。

Satellite には、カスタマーポータルに Red Hat Satellite Infrastructure サブスクリプションマニフェストが必要です。Satellite では、satellite-capsule-6.x リポジトリが有効化され、同期されている必要があります。カスタマーポータルで Red Hat サブスクリプションマニフェストを作成、管理、およびエクスポートするには、[Subscription Central](#) での [接続された Satellite Server のマニフェストの作成と管理](#) を参照してください。

Satellite Server および Capsule Server では、ホスト名の短縮名はサポートされません。カスタム証明書を使用する場合には、カスタム証明書の Common Name (CN) は短縮名ではなく完全修飾ドメイン名 (FQDN) である必要があります。これは Satellite のクライアントには適用されません。

Satellite Server をインストールする前に、環境がインストール要件を満たしていることを確認する必要があります。

Satellite Server は、新たにプロビジョニングしたシステムにインストールしておく。Satellite Server が作成するローカルのユーザーとの競合を回避するため、新たにプロビジョニングしたシステムには、以下のユーザーを外部アイデンティティプロバイダーで設定して使用しないようにしてください。

- apache
- foreman
- foreman-proxy
- postgres
- pulp

- puppet
- redis
- tomcat

認定ハイパーバイザー

Satellite Server は、Red Hat Enterprise Linux の実行をサポートするハイパーバイザーで稼働する物理システムおよび仮想マシン両方で完全にサポートされます。認定ハイパーバイザーの詳細は、[Certified Guest Operating Systems in Red Hat OpenStack Platform, Red Hat Virtualization, Red Hat OpenShift Virtualization and Red Hat Enterprise Linux with KVM](#) を参照してください。

SELinux モード

SELinux は、Enforcing モードまたは Permissive モードのいずれかで有効化されている必要があります。無効化された SELinux でのインストールはサポートされません。

FIPS モード

FIPS モードで稼働する Red Hat Enterprise Linux システムに、Satellite をインストールできます。Satellite のインストール後に FIPS モードを有効にすることはできません。詳細は、[セキュリティ強化の FIPS モードが有効な RHEL 8 システムのインストール](#) を参照してください。



注記

Satellite は、DEFAULT および FIPS 暗号化ポリシーをサポートしています。FUTURE 暗号化ポリシーは、Satellite および Capsule のインストールではサポートされていません。Future ポリシーは、考えられる将来のポリシーをテストすることを目的とする、今後焦点を合わせたより厳格なセキュリティレベルです。詳細は、Red Hat Enterprise Linux ガイドの [システム全体の暗号化ポリシーの使用](#) を参照してください。

Inter-Satellite Synchronization (ISS)

エアギャップされた Satellite Server を使用するシナリオでは、ISS エクスポート同期が機能するために、すべての Satellite Server が同じ Satellite バージョン上にある必要があります。ISS Network Sync は、それをサポートするすべての Satellite バージョンで動作します。詳細は、[コンテンツ管理ガイドの Satellite Server 間でのコンテンツ同期](#) を参照してください。

1.2. ストレージ要件

以下の表には、特定のディレクトリーのストレージ要件が詳細に記載されています。これらの値は、想定ユースケースシナリオに基づいており、各環境ごとに異なることがあります。

ランタイムサイズは Red Hat Enterprise Linux 6、7、および 8 のリポジトリーと同期して測定されました。

表1.1 Satellite Server インストールのストレージ要件

ディレクトリー	インストールサイズ	ランタイムサイズ
/var/log	10 MB	10 GB
/var/lib/pgsql	100 MB	20 GB

ディレクトリー	インストールサイズ	ランタイムサイズ
/usr	5 GB	Not Applicable
/opt/puppetlabs	500 MB	Not Applicable
/var/lib/pulp	1 MB	300 GB

外部データベースサーバーの場合: インストールサイズが 100 MB でランタイムサイズが 20 GB の `/var/lib/pgsql`。

パーティショニングとサイズの詳細は、Red Hat Enterprise Linux 8 システム設計ガイドの [パーティショニングのリファレンス](#) を参照してください。

1.3. ストレージのガイドライン

Satellite Server をインストールして効率性を向上させる場合は、以下のガイドラインを考慮してください。

- `/tmp` ディレクトリーを別のファイルシステムとしてマウントする場合は、`/etc/fstab` ファイルの `exec` マウントオプションを使用する必要があります。`/tmp` が、`noexec` オプションを指定してすでにマウントされている場合は、オプションを `exec` に変更して、ファイルシステムを再マウントする必要があります。これは、`puppetserver` サービスが機能するために必要です。
- Satellite Server データの多くは `/var` ディレクトリーに格納されるため、LVM ストレージに `/var` をマウントして、システムがスケーリングできるようにしてください。
- `/var/lib/pulp` ディレクトリーには、帯域幅が高く、レイテンシーの低いストレージを使用してください。Red Hat Satellite には I/O を大量に使用する操作が多数あるため、高レイテンシーで低帯域幅のストレージを使用すると、パフォーマンス低下の問題が発生します。インストールに、毎秒 60 - 80 メガバイトのスピードがあることを確認してください。

`storage-benchmark` スクリプトを使用して、このデータを取得できます。`storage-benchmark` スクリプトの使用の詳細は、[Impact of Disk Speed on Satellite Operations](#) を参照してください。

ファイルシステムのガイドライン

- 入出力レイテンシーが高すぎるため、GFS2 ファイルシステムは使用しないでください。

ログファイルのストレージ

ログファイルは、`/var/log/messages/`、`/var/log/httpd/`、および `/var/lib/foreman-proxy/openscap/content/` に書き込まれます。`logrotate` を使用して、これらのファイルのサイズを管理できます。詳細は、[How to use logrotate utility to rotate log files](#) を参照してください。

ログメッセージに必要なストレージの正確な容量は、インストール環境および設定により異なります。

NFS マウントに関する SELinux の考慮事項

NFS 共有を使用して `/var/lib/pulp` ディレクトリーをマウントすると、SELinux は同期プロセスをブロックします。これを避けるには、以下の行を `/etc/fstab` に追加して、ファイルシステムテーブル内の `/var/lib/pulp` ディレクトリーの SELinux コンテキストを指定します。

```
nfs.example.com:/nfsshare /var/lib/pulp nfs context="system_u:object_r:var_lib_t:s0" 1 2
```

NFS 共有がすでにマウントされている場合は、上記の方法を使用して再マウントし、以下のコマンドを入力します。

```
# restorecon -R /var/lib/pulp
```

重複パッケージ

同じパッケージが異なるリポジトリで重複して存在する場合には、ディスク上に一度しか保存されません。そのため、重複するパッケージを別のリポジトリに追加するときに必要な追加ストレージが少なく済みます。ストレージの多くは、`/var/lib/pulp/` ディレクトリにあります。これらのエンドポイントは手動で設定できません。ストレージの問題を回避するために、ストレージが `/var` ファイルシステムで利用可能であることを確認してください。

シンボリックリンク

`/var/lib/pulp/` にはシンボリックリンクは使用できません。

1.4. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする方法であれば他の方法でもインストールできます。Red Hat Satellite Server は、Satellite Server のインストール時に利用可能な Red Hat Enterprise Linux 8 の最新バージョンでサポートされています。EUS または z-stream を含む以前の Red Hat Enterprise Linux バージョンはサポートされません。

以下のオペレーティングシステムはインストーラーでサポートされ、パッケージがあり、Satellite のデプロイ用にテストされています。

表1.2 satellite-installer でサポートされるオペレーティングシステム

オペレーティングシステム	アーキテクチャー	注記
Red Hat Enterprise Linux 8	x86_64 のみ	

Red Hat は、Satellite インストーラーがいくつかのコンポーネントの設定に影響を与えるため、既存のシステムを使用しないことを推奨しています。Red Hat Satellite Server には、**@Base** パッケージグループを含む Red Hat Enterprise Linux インストールが必要です。他のパッケージセットの変更や、サーバーの運用に直接必要でないサードパーティーの設定やソフトウェアは含めないようにしてください。この制限は、ハード化や Red Hat 以外の他社のセキュリティソフトウェアが該当します。インフラストラクチャーにこのようなソフトウェアが必要な場合は、Satellite Server が完全に機能することを最初に確認し、その後でシステムのバックアップを作成して、Red Hat 以外のソフトウェアを追加します。

Red Hat では、このシステムを Satellite Server の実行以外に使用するサポートはしていません。

1.5. サポート対象のブラウザー

Satellite は、最新版の Firefox および Google Chrome ブラウザーをサポートします。

Satellite Web UI とコマンドラインインターフェイスは、英語、ポルトガル語、中国語(簡体)、中国語(繁体)、韓国語、日本語、イタリア語、スペイン語、ロシア語、フランス語、ドイツ語に対応しています。

1.6. ポートとファイアウォールの要件

Satellite アーキテクチャーのコンポーネントで通信を行うには、ベースオペレーティングシステム上で、必要なネットワークポートが開放/解放されているようにしてください。また、ネットワークベースのファイアウォールでも、必要なネットワークポートを開放する必要があります。

この情報を使用して、ネットワークベースのファイアウォールを設定してください。クラウドソリューションによっては、ネットワークベースのファイアウォールと同様にマシンが分離されるので、特にマシン間の通信ができるように設定する必要があります。アプリケーションベースのファイアウォールを使用する場合には、アプリケーションベースのファイアウォールで、テーブルに記載のアプリケーションすべてを許可して、ファイアウォールに既知の状態にするようにしてください。可能であれば、アプリケーションのチェックを無効にして、プロトコルをベースにポートの通信を開放できるようにしてください。

統合 Capsule

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下のセクションのコンテキストでは Satellite のクライアントになります。これには、Capsule Server が実行されているベースオペレーティングシステムが含まれます。

Capsule のクライアント

Satellite と統合された Capsule ではない Capsule のクライアントであるホストには、Satellite Server へのアクセスは必要ありません。Satellite トポロジーとポート接続の図に関する詳細は、[概要、概念、およびデプロイメントの考慮事項](#)の [Capsule のネットワーク](#) を参照してください。

使用している設定に応じて、必要なポートは変わることがあります。

以下の表は、宛先ポートとネットワークトラフィックの方向を示しています。

表1.3 Satellite Server の受信トラフィック

送信先ポート	プロトコル	サービス	ソース	用途	説明
53	TCP および UDP	DNS	DSN サーバーおよびクライアント	名前解決	DNS (オプション)
67	UDP	DHCP	クライアント	動的 IP	DHCP (オプション)
69	UDP	TFTP	クライアント	TFTP サーバー (オプション)	
443	TCP	HTTPS	Capsule	Red Hat Satellite API	Capsule からの通信

443、80	TCP	HTTPS, HTTP	クライアント	グローバル登録	Satellite へのホストの登録 ポート 443 は、登録の開始、ファクトのアップロード、インストールされたパッケージとトレースの送信に必要です。 ポート 80 は、 /unattended/built エンドポイントで登録が完了したことを通知します。
443	TCP	HTTPS	Red Hat Satellite	コンテンツミラーリング	管理
443	TCP	HTTPS	Red Hat Satellite	Capsule API	スマートプロキシ機能
443、80	TCP	HTTPS, HTTP	Capsule	コンテンツの取得	コンテンツ
443、80	TCP	HTTPS, HTTP	クライアント	コンテンツの取得	コンテンツ
1883	TCP	MQTT	クライアント	プルベースの REX (オプション)	REX ジョブ通知用のコンテンツホスト (オプション)
5910 - 5930	TCP	HTTPS	ブラウザー	コンピュートリソースの仮想コンソール	
8000	TCP	HTTP	クライアント	プロビジョニングテンプレート	クライアントインストーラー、iPXE または UEFI HTTP ブートのテンプレート取得
8000	TCP	HTTPS	クライアント	PXE ブート	インストール
8140	TCP	HTTPS	クライアント	puppet-agent	クライアントの更新 (オプション)

9090	TCP	HTTPS	Red Hat Satellite	Capsule API	スマートプロキシ機能
9090	TCP	HTTPS	クライアント	OpenSCAP	クライアントの設定 (OpenSCAP プラグインがインストールされている場合)
9090	TCP	HTTPS	検出されたノード	検出	ホストの検出とプロビジョニング (検出プラグインがインストールされている場合)

Satellite Server に直接接続されたホストは、統合された Capsule のクライアントとなるため、このコンテキストではクライアントになります。これには、Capsule Server が稼働しているベースオペレーティングシステムが含まれます。

DHCP Capsule は、DHCP IPAM が設定されたサブネット内のホストに対して ICMP ping または TCP Echo 接続の試行を実行し、使用が検討されている IP アドレスが空いているかどうかを確認します。この動作は、**satellite-installer --foreman-proxy-dhcp-ping-free-ip=false** を使用してオフにできます。



注記

発信トラフィックの一部は Satellite に戻り、内部通信とセキュリティ操作を有効にします。

表1.4 Satellite Server の発信トラフィック

送信先ポート	プロトコル	サービス	宛先	用途	説明
	ICMP	ping	クライアント	DHCP	解放されている IP チェック (オプション)
7	TCP	echo	クライアント	DHCP	解放されている IP チェック (オプション)
22	TCP	SSH	ターゲットホスト	リモート実行	ジョブの実行
22, 16514	TCP	SSH SSH/TLS	Compute Resource (コンピュータリソース)	libvirt のコンピュートリソースに対する Satellite による通信	

送信先ポート	プロトコル	サービス	宛先	用途	説明
53	TCP および UDP	DNS	インター ネット上の DNS サー バー	DNS サーバー	DNS レコードの解 決 (オプション)
53	TCP および UDP	DNS	DNS サー バー	--capsule-dns	DNS 競合の検証 (オプション)
53	TCP および UDP	DNS	DNS サー バー	オーケストレー ション	DNS 競合の検証
68	UDP	DHCP	クライアン ト	動的 IP	DHCP (オプショ ン)
80	TCP	HTTP	リモートリ ポジトリー	コンテンツ同期	リモートリポジト リー
389、636	TCP	LDAP、 LDAPS	外部 LDAP サーバー	LDAP	LDAP 認証。外部 認証が有効になっ ている場合にのみ 必要で す。 LDAPAuthS ource が定義され ている場合、ポー トをカスタマイズ できます
443	TCP	HTTPS	Satellite	Capsule	Capsule 設定管理 テンプレートの取 得 OpenSCAP リモート実行結果 のアップロード
443	TCP	HTTPS	Amazon EC2, Azure, Google GCE	コンピュートリ ソース	仮想マシンのイン タラクション (ク エリー/作成/破棄) (オプション)
443	TCP	HTTPS	Capsule	コンテンツのミ ラーリング	開始

送信先ポート	プロトコル	サービス	宛先	用途	説明
443	TCP	HTTPS	Infoblox DHCP サー バー	DHCP 管理	DHCP に Infoblox を使用する場合、 DHCP リースの管 理 (オプション)
623			クライアン ト	電源管理	BMC のオン/オ フ/サイクル/ス テータス
5000	TCP	HTTPS	OpenStack Compute Resource	コンピュートリ ソース	仮想マシンのイン タラクション (ク エリー/作成/破棄) (オプション)
5900-5930	TCP	SSL/TLS	ハイパーバ イザー	noVNC コンソ ール	noVNC コンソ ールの起動
7911	TCP	DHCP、 OMAPI	DHCP サー バー	DHCP	DHCP ターゲット は、 --foreman- proxy-dhcp- server を使用し て設定される。デ フォルトは localhost。 ISC と remote_isc は、 デフォルトが 7911 で、OMAPI を使用 する設定可能な ポートを使用する
8443	TCP	HTTPS	クライアン ト	検出	Capsule は、検出 されたホストに再 起動コマンドを送 信する (オプショ ン)
9090	TCP	HTTPS	Capsule	Capsule API	Capsule の管理

1.7. クライアントから SATELLITE SERVER への接続の有効化

Satellite Server の内部 Capsule のクライアントである Capsule とコンテンツホストは、Satellite のホストベースのファイアウォールとすべてのネットワークベースのファイアウォールを介したアクセスを必要とします。

以下の手順を使用して、Satellite のインストール先のシステムでホストベースのファイアウォールを設定し、クライアントからの受信接続を有効にして、これらの設定をシステムの再起動後にも保持する方法について説明します。使用されるポートの詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の [ポートとファイアウォールの要件](#) を参照してください。

手順

1. Satellite Server 上のクライアント用のポートを開きます。

```
# firewall-cmd \
--add-port="5647/tcp" \
--add-port="8000/tcp" \
--add-port="9090/tcp"
```

2. Satellite Server 上のサービスへのアクセスを許可します。

```
# firewall-cmd \
--add-service=dns \
--add-service=dhcp \
--add-service=tftp \
--add-service=http \
--add-service=https \
--add-service=puppetmaster
```

3. 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

検証

- 以下のコマンドを入力します。

```
# firewall-cmd --list-all
```

詳細は、Red Hat Enterprise Linux 8 のネットワークの保護の [firewalld の使用および設定](#) を参照してください。

1.8. DNS 解決の検証

完全修飾ドメイン名を使用して完全な正引きおよび逆引き DNS 解決を検証すると、Satellite のインストール中の問題を回避できます。

手順

1. ホスト名とローカルホストが正しく解決されることを確認します。

```
# ping -c1 localhost
# ping -c1 `hostname -f` # my_system.domain.com
```

名前解決に成功すると、以下のような出力が表示されます。

```
# ping -c1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.043 ms

--- localhost ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.043/0.043/0.043/0.000 ms
```

```
# ping -c1 `hostname -f`
PING hostname.gateway (XX.XX.XX.XX) 56(84) bytes of data:
64 bytes from hostname.gateway (XX.XX.XX.XX): icmp_seq=1 ttl=64 time=0.019 ms

--- localhost.gateway ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.019/0.019/0.019/0.000 ms
```

2. 静的および一時的なホスト名との不一致を避けるには、次のコマンドを入力して、システム上のすべてのホスト名を設定します。

```
# hostnamectl set-hostname name
```

詳細は、[Red Hat Enterprise Linux 8 ネットワークの設定と管理の hostnamectl を使用したホスト名の変更](#) を参照してください。



警告

Satellite の運用には名前解決が非常に重要です。Satellite が完全修飾ドメイン名を適切に解決できない場合には、コンテンツ管理、サブスクリプション管理、プロビジョニングなどのタスクに失敗します。

1.9. 事前定義済みプロファイルを使用した SATELLITE SERVER の調整

Satellite のデプロイメントに 5000 台を超えるホストが含まれる場合には、事前定義済みの tuning プロファイルを使用して Satellite のパフォーマンスを向上できます。

Capsule では tuning プロファイルを使用できない点に注意してください。

Satellite が管理するホストの数と利用可能なハードウェアリソースに応じて、プロファイルの1つを選択できます。

tuning プロファイルは、`/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes` ディレクトリにあります。

`--tuning` オプションを指定して `satellite-installer` コマンドを実行した場合には、デプロイメント設定が以下の順番で Satellite Server に適用されます。

1. `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` ファイルで定義したデフォルトの tuning プロファイル
2. `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/` ディレクトリで定義され、デプロイメントに適用する tuning プロファイル
3. オプション: `/etc/foreman-installer/custom-hiera.yaml` ファイルを設定した場合、Satellite はこれらの設定を適用します。

`/etc/foreman-installer/custom-hiera.yaml` ファイルで定義した設定は、tuning プロファイルで定義した設定を上書きすることに注意してください。

したがって、tuning プロファイルを適用する前に、`/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` のデフォルトの tuning プロファイルに定義されている設定、適用する tuning プロファイル、および `/etc/foreman-installer/custom-hiera.yaml` ファイルを比較して、重複する設定内容を `/etc/foreman-installer/custom-hiera.yaml` ファイルから削除する必要があります。

default

ホスト数: 0 - 5000

RAM: 20G

CPU コア数: 4

medium

ホスト数: 5001 - 10000

RAM: 32G

CPU コア数: 8

large

ホスト数: 10001 - 20000

RAM: 64G

CPU コア数: 16

extra-large

ホスト数: 20001 - 60000

RAM: 128G

CPU コア数: 32

extra-extra-large

ホスト数: 60000 以上

RAM: 256G

CPU コア数: 48+

手順

1. オプション: Satellite Server で、**custom-hiera.yaml** ファイルを設定した場合、**/etc/foreman-installer/custom-hiera.yaml** ファイルを **custom-hiera.original** にバックアップします。**/etc/foreman-installer/custom-hiera.yaml** ファイルが破損した場合には、バックアップファイルを使用して、ファイルを元の状態に戻します。

```
# cp /etc/foreman-installer/custom-hiera.yaml \  
/etc/foreman-installer/custom-hiera.original
```

2. オプション: Satellite Server で **custom-hiera.yaml** ファイルを設定した場合、**/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** のデフォルト tuning プロファイルの定義と、**/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** に適用する tuning プロファイルを確認します。**/etc/foreman-installer/custom-hiera.yaml** ファイルの設定内容と比較して、**/etc/foreman-installer/custom-hiera.yaml** ファイルで重複設定を削除します。

- 適用するプロファイルに対して、**--tuning** オプションを指定して **satellite-installer** コマンドを入力します。たとえば、medium tuning プロファイル設定を適用するには、以下のコマンドを入力します。

```
# satellite-installer --tuning medium
```

第2章 SATELLITE SERVER のインストール

Satellite Server に使用予定のホストがオフライン環境にある場合には、外部のコンピューターを使用してパッケージの ISO イメージをダウンロードし、Satellite Server のインストール先にパッケージをコピーし、Satellite Server をインストールします。他の状況では、ISO イメージに最新の更新、バグ修正、機能が含まれていない可能性があるため、この手法は推奨していません。

以下の手順を使用して、Satellite Server をインストールし、初期設定を実行して、サブスクリプション マニフェストをインポートします。

作業を進める前に、使用している環境に適切なマニフェストを確認します。マニフェストの詳細は、[コンテンツの管理](#) の [Red Hat サブスクリプションの管理](#) を参照してください。



注記

Satellite Server に自己登録することはできません。

2.1. バイナリー DVD イメージのダウンロード

以下の手順を使用して、Red Hat Enterprise Linux Server と Red Hat Satellite をダウンロードします。

手順

1. [Red Hat カスタマーポータル](#) に移動し、ログインします。
2. **ダウンロード** をクリックします。
3. **Red Hat Enterprise Linux** を選択します。
4. 製品とバージョンがご使用の環境に適切であることを確認します。
 - **Product Variant (製品のバリエーション)** は **Red Hat Enterprise Linux for x86_64** に設定します。
 - **Version (バージョン)** は、ベースオペレーティングシステムとして使用する予定の製品の最新マイナーバージョンに設定します。
 - **Architecture (アーキテクチャー)** は 64 ビットバージョンに設定します。
5. **Product Software (製品ソフトウェア)** タブで、最新の **Red Hat Enterprise Linux for x86_64** バージョン向けのバイナリー DVD イメージをダウンロードします。
6. **DOWNLOADS (ダウンロード)** をクリックし、**Red Hat Satellite** を選択します。
7. 製品とバージョンがご使用の環境に適切であることを確認します。
 - **Product Variant (製品のバリエーション)** は **Red Hat Satellite** に設定します。
 - **Version (バージョン)** は、使用する予定の製品の最新マイナーバージョンに設定します。
8. **Product Software (製品ソフトウェア)** タブで、最新の Red Hat Satellite バージョン向けのバイナリー DVD イメージをダウンロードします。
9. ISO ファイルを Satellite ベースオペレーティングシステムの **/var/tmp** または他のアクセス可能なストレージデバイスにコピーします。

```
# scp localfile username@hostname:remotefile
```

2.2. RHEL 8 でのオフラインリポジトリを使用したベースオペレーティングシステムの設定

以下の手順を使用して、Red Hat Enterprise Linux 8 および Red Hat Satellite ISO イメージのオフラインリポジトリを設定します。

手順

1. ベースオペレーティングシステムのバージョンに対応する ISO ファイルのマウントポイントとして使用するディレクトリを作成します。

```
# mkdir /media/rhel8
```

2. Red Hat Enterprise Linux の ISO イメージをマウントポイントにマウントします。

```
# mount -o loop rhel8-DVD.iso /media/rhel8
```

3. ISO ファイルのリポジトリデータファイルをコピーして権限を変更するには、次のように入力します。

```
# cp /media/rhel8/media.repo /etc/yum.repos.d/rhel8.repo
# chmod u+w /etc/yum.repos.d/rhel8.repo
```

4. リポジトリデータファイルを編集し、**baseurl** ディレクティブを追加します。

```
[RHEL8-BaseOS]
name=Red Hat Enterprise Linux BaseOS
mediaid=None
metadata_expire=-1
gpgcheck=0
cost=500
baseurl=file:///media/rhel8/BaseOS/

[RHEL8-AppStream]
name=Red Hat Enterprise Linux Appstream
mediaid=None
metadata_expire=-1
gpgcheck=0
cost=500
baseurl=file:///media/rhel8/AppStream/
```

5. リポジトリが設定されたことを確認します。

```
# yum repolist
```

6. Satellite Server の ISO ファイルのマウントポイントとして使用するディレクトリを作成します。

```
# mkdir /media/sat6
```

7. Satellite Server の ISO イメージをマウントポイントにマウントします。

```
# mount -o loop sat6-DVD.iso /media/sat6
```

2.3. オプション: SATELLITE SERVER での FAPOLICYD の使用

Satellite Server で **fapolicyd** を有効にすると、ファイルとディレクトリーへのアクセスを監視および制御して、セキュリティをさらに強化できます。fapolicyd デーモンは、信頼できるバイナリーとスクリプトのリポジトリーとして、RPM データベースを使用します。

Satellite Server または Capsule Server 上の fapolicyd はいつでもオンまたはオフにできます。

2.3.1. Satellite Server への fapolicyd のインストール

fapolicyd は、Satellite Server と一緒にインストールすることも、既存の Satellite Server にインストールすることもできます。新しい Satellite Server と一緒に **fapolicyd** をインストールする場合、インストールプロセスによって Red Hat Enterprise Linux ホスト内の fapolicyd が検出され、Satellite Server ルールが自動的にデプロイされます。

前提条件

- ホストが Red Hat Enterprise Linux の BaseOS リポジトリーにアクセスできる。

手順

1. fapolicyd をインストールします。

```
# dnf install fapolicyd
```

2. **fapolicyd** サービスを開始します。

```
# systemctl enable --now fapolicyd
```

検証

- **fapolicyd** サービスが正しく実行されていることを確認します。

```
# systemctl status fapolicyd
```

新しい Satellite Server または Capsule Server のインストール

新しい Satellite Server または Capsule Server をインストールする場合は、Red Hat Enterprise Linux ホストに fapolicyd をインストールして有効にした後、標準のインストール手順に従ってください。

関連情報

fapolicyd の詳細は、[Red Hat Enterprise Linux 8 セキュリティー強化の fapolicyd を使用したアプリケーションの拒否および許可](#) を参照してください。

2.4. オフラインリポジトリーからの SATELLITE パッケージのインストール

以下の手順を使用して、オフラインのリポジトリーから Satellite パッケージをインストールします。

手順

1. Red Hat Enterprise Linux Server と Red Hat Satellite の ISO イメージがマウントされていることを確認します。

```
# findmnt -t iso9660
```

2. Red Hat GPG キーをインポートします。

```
# rpm --import /etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
```

3. バイナリー DVD イメージを使用してベースオペレーティングシステムを最新の状態にします。

```
# dnf update
```

4. Satellite ISO がマウントされたディレクトリーに移動します。

```
# cd /media/sat6/
```

5. マウントされたディレクトリーでインストールスクリプトを実行します。

```
# ./install_packages
```



注記

このスクリプトには、**satellite:el8** モジュールを有効にするコマンドが含まれています。モジュール **satellite:el8** を有効にすると、**postgresql:10** および **ruby:2.5** との競合について警告が表示されます。これは、これらのモジュールが Red Hat Enterprise Linux 8 でデフォルトのモジュールバージョンに設定されているためです。モジュール **satellite:el8** には、モジュール **postgresql:12** および **ruby:2.7** への依存関係があり、**satellite:el8** モジュールで有効になります。これらの警告はインストールプロセスの失敗の原因にはならないため、安全に無視できます。Red Hat Enterprise Linux 8 のモジュールとライフサイクルストリームの詳細は、[Red Hat Enterprise Linux Application Streams のライフサイクル](#) を参照してください。

Satellite パッケージが正常にインストールされると、以下のメッセージが表示されません。 **Install is complete. Please run satellite-installer --scenario satellite.**

2.5. パッケージの依存関係エラーの解決

Satellite Server パッケージのインストール中にパッケージの依存関係のエラーが発生した場合に、Red Hat カスタマーポータルからパッケージをダウンロードしてインストールすることで、エラーを解決できます。依存関係エラーの解決に関する詳細は、KCS ソリューション [How can I use the yum output to solve yum dependency errors?](#) を参照してください。

Satellite パッケージを正常にインストールした場合は、この手順をスキップしてください。

手順

1. [Red Hat カスタマーポータル](#) に移動して、ログインします。
2. **ダウンロード** をクリックします。

- ダウンロードするパッケージが含まれる製品をクリックします。
- 環境に適した **製品バリエーション**、**バージョン**、**アーキテクチャー** が選択されていることを確認します。
- パッケージ** タブをクリックします。
- 検索** フィールドに、パッケージの名前を入力します。
- パッケージをクリックします。
- バージョン** リストからパッケージのバージョンを選択します。
- ページの下部で、**今すぐダウンロード** をクリックします。
- パッケージを Satellite のベースオペレーティングシステムにコピーします。
- Satellite Server でパッケージが配置されているディレクトリーに、移動します。

```
# cd /path-to-package/
```

- ローカルでパッケージをインストールします。

```
# dnf install package_name
```

- Satellite ISO がマウントされたディレクトリーに移動します。

```
# cd /media/sat6/
```

- Satellite Server パッケージをインストールして、パッケージの依存関係エラーが解決されたことを確認します。パッケージの依存関係エラーがさらにある場合は、この手順を繰り返します。

```
# ./install_packages
```



注記

このスクリプトには、**satellite:el8** モジュールを有効にするコマンドが含まれています。モジュール **satellite:el8** を有効にすると、**postgresql:10** および **ruby:2.5** との競合について警告が表示されます。これは、これらのモジュールが Red Hat Enterprise Linux 8 でデフォルトのモジュールバージョンに設定されているためです。モジュール **satellite:el8** には、モジュール **postgresql:12** および **ruby:2.7** への依存関係があり、**satellite:el8** モジュールで有効になります。これらの警告はインストールプロセスの失敗の原因にはならないため、安全に無視できます。Red Hat Enterprise Linux 8 のモジュールとライフサイクルストリームの詳細は、[Red Hat Enterprise Linux Application Streams のライフサイクル](#) を参照してください。

Satellite パッケージが正常にインストールされると、以下のメッセージが表示されます。**Install is complete.Please run satellite-installer --scenario satellite.**

2.6. CHRONYD によるシステムクロックの同期

時間のずれを最小限に抑えるには、Satellite Server をインストールするベースオペレーティングシステムのシステムクロックを Network Time Protocol (NTP) サーバーと同期する必要があります。ベースオペレーティングシステムのクロックが正しく設定されていない場合には、証明書の検証に失敗する可能性があります。

chrony スイートの詳細は、Red Hat Enterprise Linux 8 の基本的なシステム設定の [Chrony スイートを使用した NTP の設定](#) を参照してください。

手順

1. **chrony** パッケージをインストールします。

```
# dnf install chrony
```

2. **chronyd** サービスを起動して、有効にします。

```
# systemctl enable --now chronyd
```

2.7. ベースオペレーティングシステムへの SOS パッケージのインストール

ベースオペレーティングシステムに **sos** パッケージをインストールし、Red Hat Enterprise Linux システムから設定および診断情報を取得できるようにします。このパッケージを使用すると、Red Hat テクニカルサポートへのサービスリクエストの起票時に必要な初期システム分析を提示できます。**sos** の使用方法に関する詳細は、カスタマーポータルナレッジベースソリューション [What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later?](#) を参照してください。

手順

- **sos** パッケージをインストールするには、以下のコマンドを実行します。

```
# satellite-maintain packages install sos
```

2.8. SATELLITE SERVER の設定

satellite-installer インストールスクリプトを使用して Satellite Server をインストールします。以下の手法から1つ選択します。

- 「[Satellite インストールの設定](#)」。この手法では、1つまたは複数のコマンドオプションを指定して、インストールスクリプトを実行します。コマンドオプションは、対応するデフォルトの初期設定オプションを上書きし、Satellite 応答ファイルに記録されます。必要なオプションの設定に、必要に応じてスクリプトは何回でも実行することができます。

2.8.1. Satellite インストールの設定

初期設定の手順では、組織、ロケーション、ユーザー名、およびパスワードが作成されます。初期設定後に、必要に応じて追加の組織とロケーションを作成できます。初期設定では、PostgreSQL データベースも同じサーバーにインストールします。

インストールプロセスの完了には、数十分かかることがあります。システムにリモートで接続する場合は、リモートシステムから切断された場合にインストールの進捗を確認できるように、通信セッションの一時中断または再接続を許可できる **tmux** などのユーティリティーを使用してください。インストールコマンドを実行しているシェルへの接続が切断された場合は、`/var/log/foreman-installer/satellite.log` のログを参照してプロセスが正常に完了したかどうかを確認します。

留意事項

- **satellite-installer --scenario satellite --help** コマンドを使用して、利用可能なオプションとすべてのデフォルト値を表示します。値を指定しない場合は、デフォルト値が使用されます。
- **--foreman-initial-organization** オプションに、意味を持つ値を指定します。たとえば、会社名を指定できます。値に一致する内部ラベルが作成されますが、このラベルは後で変更できません。値を指定しない場合は、ラベルが **Default_Organization** の **Default Organization** という名前の組織が作成されます。組織名は変更できますが、ラベルは変更できません。
- デフォルトでは、インストーラーが設定するすべての設定ファイルが管理されます。**satellite-installer** を実行すると、管理対象のファイルに手動で加えられた変更が初期値で上書きされます。これは、破損したシステム上でインストーラーを実行すると、変更が加えられたかどうかに関係なく、インストーラーを動作可能な状態に復元する必要があります。他のサービスにカスタム設定を適用する方法は、[Satellite へのカスタム設定の適用](#) を参照してください。

手順

1. 使用する追加オプションを指定し、以下のコマンドを入力します。

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "My_Organization" \
--foreman-initial-location "My_Location" \
--foreman-initial-admin-username admin_user_name \
--foreman-initial-admin-password admin_password
```

このスクリプトは、進捗を表示し、`/var/log/foreman-installer/satellite.log` にログを記録します。

2. ISO イメージをアンマウントします。

```
# umount /media/sat6
# umount /media/rhel8
```

2.9. サブスクリプション接続の無効化

Red Hat ポータルへの接続を回避するには、オフラインの Satellite Server でサブスクリプション接続を無効にします。これにより、マニフェストの更新やアップストリームのエンタイトルメントの更新もできなくなります。

手順

1. Satellite Web UI で、**Administer > Settings** に移動します。
2. **Content** タブをクリックします。
3. **Subscription Connection Enabled** の値を **No** に設定します。

CLI 手順

- Satellite Server で以下のコマンドを入力します。

```
# hammer settings set --name subscription_connection_enabled --value false
```

2.10. RED HAT サブスクリプションマニフェストの SATELLITE SERVER へのインポート

以下の手順を使用して、Red Hat サブスクリプションマニフェストを Satellite Server にインポートします。



注記

Simple Content Access (SCA) は、マニフェストではなく組織で設定されます。マニフェストをインポートしても、組織の Simple Content Access のステータスは変更されません。

前提条件

- [Red Hat カスタマーポータル](#) から Red Hat サブスクリプションマニフェストファイルをエクスポートしている。詳細は、[Using Red Hat Subscription Management](#) の [Creating and Managing Manifests](#) を参照してください。
- Satellite Server でサブスクリプション接続を無効にしていることを確認してください。詳細は、「[サブスクリプション接続の無効化](#)」を参照してください。

手順

1. Satellite Web UI で、コンテキストが、使用する組織に設定されていることを確認します。
2. Satellite Web UI で、**Content > Subscriptions** に移動し、**Manage Manifest** をクリックします。
3. **Manage Manifest** ウィンドウで、**Choose File** をクリックします。
4. Red Hat サブスクリプションマニフェストファイルが保存されている場所に移動し、**Open** をクリックします。

CLI 手順

1. Red Hat サブスクリプションマニフェストファイルをローカルマシンから Satellite Server にコピーします。

```
$ scp ~/manifest_file.zip root@satellite.example.com:~/.
```

2. Satellite Server に **root** ユーザーとしてログインし、Red Hat サブスクリプションマニフェストファイルをインポートします。

```
# hammer subscription upload \  
--file ~/manifest_file.zip \  
--organization "My_Organization"
```

リポジトリを有効にし、Red Hat コンテンツをインポートすることができるようになりました。詳細は、[コンテンツの管理](#) の [コンテンツのインポート](#) を参照してください。

第3章 SATELLITE SERVER での追加設定の実行

3.1. カスタム CDN からコンテンツを使用するための SATELLITE SERVER の設定

内部コンテンツ配信ネットワーク (CDN) がある場合、またはアクセス可能な Web サーバーでコンテンツを提供する場合、Red Hat CDN の代わりにこの CDN サーバーから Red Hat リポジトリを使用するように、Satellite Server を設定できます。CDN サーバーには、Red Hat CDN と同じディレクトリー構造のリポジトリをミラーリングする Web サーバーを使用できます。

各組織のコンテンツのソースを設定できます。Satellite は、組織内のサブスクリプションマニフェストからどの Red Hat リポジトリが CDN サーバーで利用できるかを自動的に認識します。

前提条件

- Red Hat コンテンツを提供し、Satellite Server からアクセスできる CDN サーバーがある。
- CDN サーバーが HTTPS を使用する場合は、SSL 証明書を Satellite にアップロードしていることを確認する。詳細は、[コンテンツの管理](#) の [カスタム SSL 証明書のインポート](#) を参照してください。
- マニフェストを組織にアップロードしている。

手順

1. Satellite Web UI で、[コンテンツ > サブスクリプション](#) に移動します。
2. [マニフェストの管理](#) をクリックします。
3. [CDN 設定](#) タブを選択します。
4. [カスタム CDN](#) タブを選択します。
5. **URL** フィールドには、Satellite Server が Red Hat リポジトリを消費する CDN サーバーの URL を入力します。
6. オプション: [SSL CA コンテンツ認証情報](#) で、CDN サーバーの SSL 証明書をを選択します。
7. **Update** をクリックします。
8. 内部 CDN サーバーから消費された Red Hat リポジトリを有効化できるようになりました。

CLI 手順

1. SSH を使用して Satellite Server に接続します。
2. CDN 設定をカスタム CDN サーバーに設定します。

```
# hammer organization configure-cdn --name="My_Organization" \  
--type=custom_cdn \  
--url https://my-cdn.example.com \  
--ssl-ca-credential-id "My_CDN_CA_Cert_ID"
```

関連情報

- [概要、概念、およびデプロイメントの考慮事項](#) の [コンテンツ配信ネットワーク構造](#)

3.2. INTER-SATELLITE SYNCHRONIZATION の設定方法

Red Hat Satellite は Inter-Satellite Synchronization (ISS) を使用して、エアギャップされたものを含む 2 つの Satellite Server 間でコンテンツを同期します。

ISS は、次のような場合に使用できます。

- Satellite Server の一部のコンテンツのみを他の Satellite Server にコピーする場合。たとえば、IT 部門が Satellite Server から使用するコンテンツビューがあり、そのコンテンツビューから他の Satellite Server にコンテンツをコピーする場合があります。
- Satellite Server から他の Satellite Server にすべてのライブラリーコンテンツをコピーする場合。たとえば、IT 部門がライブラリー内の Satellite Server から消費する製品とリポジトリーがあり、その組織内のすべての製品およびリポジトリーを他の Satellite Server にコピーしたいとします。



注記

ISS は、Satellite Server から Capsule Server へのコンテンツの同期には使用できません。Capsule Server はネイティブで同期をサポートします。詳細は、[概要、概念、およびデプロイメントの考慮事項](#) の [Capsule Server の概要](#) を参照してください。

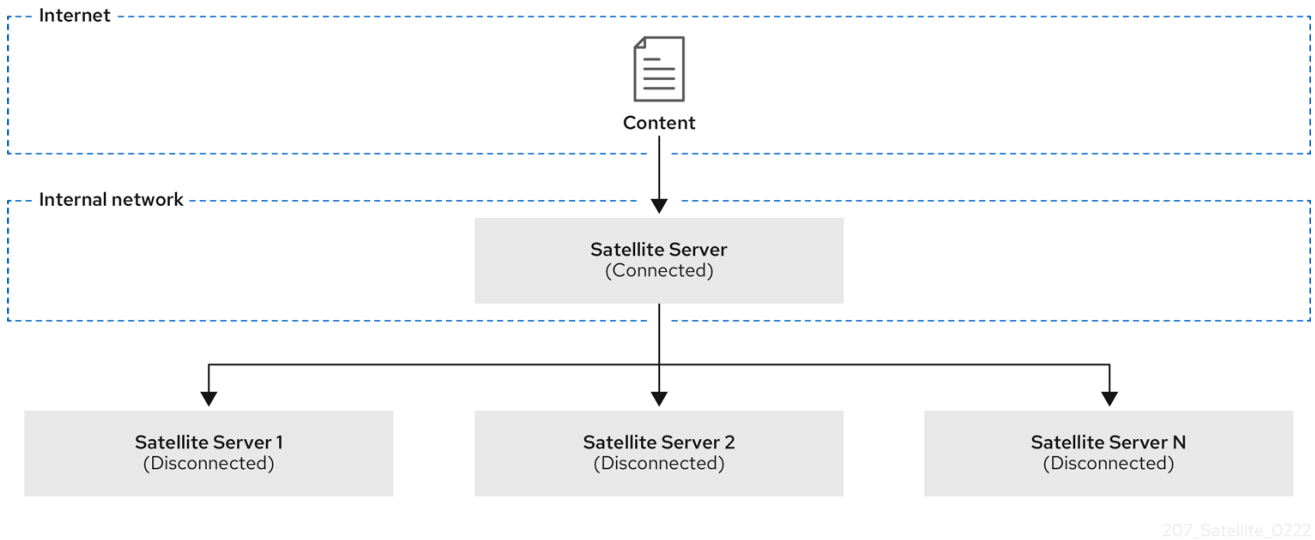
ISS の使用方法はいくつかあります。使用方法は、次のいずれかのシナリオに該当する可能性のあるマルチサーバー設定によって異なります。

3.2.1. 非接続シナリオでの ISS ネットワーク同期

非接続シナリオでは、次の設定があります。

- アップストリームの Satellite Server はインターネットに接続されています。このサーバーは、Red Hat Content Delivery Network (CDN) またはカスタムソースからのコンテンツを消費しません。
- ダウンストリームの Satellite Server は、すべての外部ネットワークから完全に分離されています。
- ダウンストリーム Satellite Server は、内部ネットワークを介して接続されたアップストリーム Satellite Server と通信できます。

図3.1 Satellite ISS の非接続シナリオ



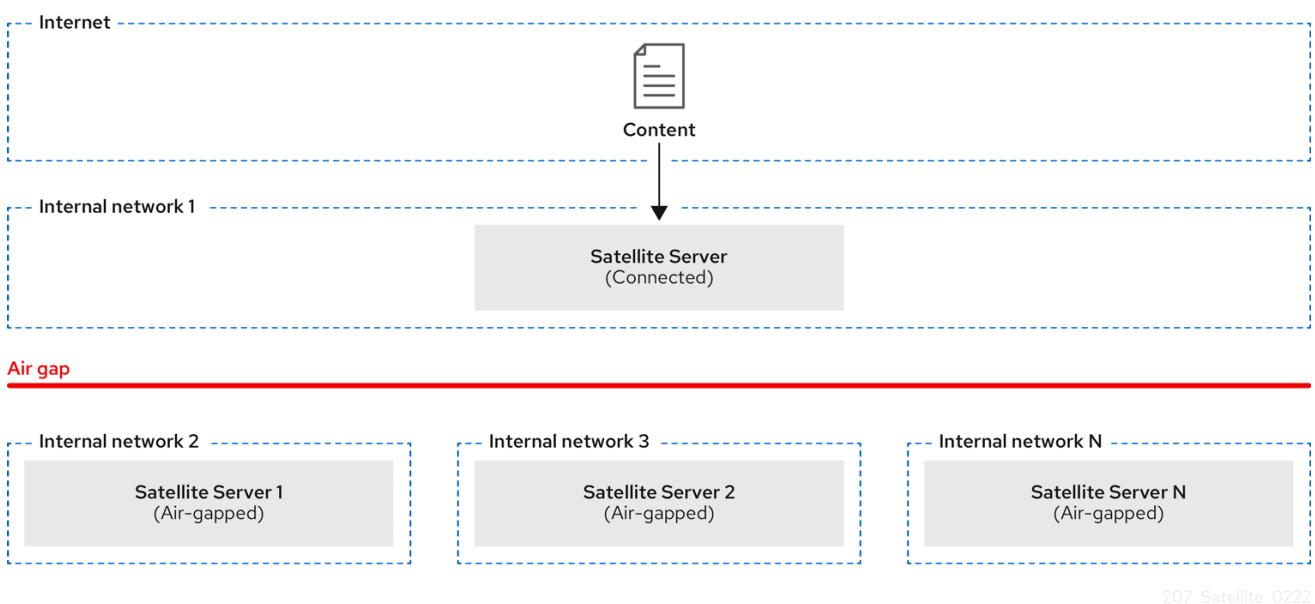
ネットワークを介してアップストリーム Satellite Server からのコンテンツを同期するようにダウンストリーム Satellite Server を設定できます。「[ネットワーク経由でコンテンツを同期するように Satellite Server を設定する](#)」を参照してください。

3.2.2. エアギャップシナリオでの ISS エクスポート同期

エアギャップシナリオでは、次の設定があります。

- アップストリームの Satellite Server はインターネットに接続されています。このサーバーは、Red Hat CDN またはカスタムソースからのコンテンツを消費します。
- ダウンストリームの Satellite Server は、すべての外部ネットワークから完全に分離されています。
- ダウンストリーム Satellite Server には、接続されたアップストリーム Satellite Server へのネットワーク接続がありません。

図3.2 Satellite ISS のエアギャップシナリオ



エアギャップのあるダウストリーム Satellite Server がコンテンツの更新を受け取る唯一の方法は、アップストリーム Satellite Server からペイロードをエクスポートし、それをダウストリーム Satellite Server に物理的に移動し、ペイロードをインポートすることです。詳細は、[コンテンツ管理ガイドの Satellite Server 間でのコンテンツ同期](#) を参照してください。

エクスポートを使用してコンテンツを同期するようにダウストリーム Satellite Server を設定します。「[エクスポートを使用してコンテンツを同期するように Satellite Server を設定する](#)」を参照してください。

3.3. ネットワーク経由でコンテンツを同期するように SATELLITE SERVER を設定する

接続されたアップストリーム Satellite Server からのリポジトリを HTTPS 経由で同期するようにダウストリーム Satellite Server を設定します。

前提条件

- アップストリームの Satellite Server とダウストリームの Satellite Server の間にネットワーク接続が存在します。
- サブスクリプションマニフェストをアップストリームとダウストリームの両方の Satellite Server にインポートしました。
- アップストリームの Satellite Server で組織に必要なリポジトリを有効にしました。詳細は、[コンテンツの管理の Red Hat リポジトリの有効化](#) を参照してください。
- アップストリームユーザーとは、admin または以下の権限を持つユーザーのことです。
 - **view_organizations**
 - **view_products**
 - **export_content**
 - **view_lifecycle_environments**
 - **view_content_views**
- ダウストリームの Satellite Server に、**http://upstream-satellite.example.com/pub/katello-server-ca.crt** の内容でアップストリームの Satellite Server の SSL 証明書をインポートしています。詳細は、[コンテンツの管理の SSL 証明書のインポート](#) を参照してください。
- ダウストリームのユーザーは、管理者、または製品リポジトリや組織を作成する権限を持っているユーザーとなります。

手順

1. [コンテンツ > サブスクリプション](#) に移動します。
2. [マニフェストの管理](#) をクリックします。
3. [CDN Configuration](#) タブに移動します。
4. [Network Sync](#) タブを選択します。
5. URL フィールドに、アップストリームの Satellite Server のアドレスを入力します。

6. **Username** に、アップストリームログイン用のユーザー名を入力します。
7. **Password** に、アップストリームログイン用のパスワードまたは Personal Access Token を入力します。
8. **Organization label** フィールドに、アップストリーム組織のラベルを入力します。
9. オプション: **Lifecycle Environment Label** フィールドに、アップストリームライフサイクル環境のラベルを入力します。デフォルトは **Library** です。
10. オプション: **Content view label** フィールドに、アップストリームのコンテンツビューのラベルを入力します。デフォルトは **Default_Organization_View** です。
11. **SSL CA Content Credential** メニューから、アップストリームの Satellite Server が使用する CA 証明書を選択します。
12. **Update** をクリックします。
13. Satellite Web UI で、**Content > Products** に移動します。
14. 同期するリポジトリが含まれている製品を選択します。
15. **Select Action** メニューから **Sync Now** を選択して、製品内のすべてのリポジトリを同期します。
同期計画を作成して、定期的に更新することもできます。詳細は、[コンテンツの管理の同期計画の作成](#) を参照してください。

CLI手順

1. ダウンストリームの Satellite Server に SSH で接続します。
2. アップストリーム CA 証明書に関する情報を表示します。

```
# hammer content-credential show \
--name="My_Upstream_CA_Cert" \
--organization="My_Downstream_Organization"
```

次のステップのために、CA 証明書の ID をメモしておいてください。

3. CDN 設定をアップストリーム Satellite Server に設定します。

```
# hammer organization configure-cdn --name="My_Downstream_Organization" \
--type=network_sync \
--url https://upstream-satellite.example.com \
--username upstream_username --password upstream_password \
--ssl-ca-credential-id "My_Upstream_CA_Cert_ID" \ --upstream-organization-
label="My_Upstream_Organization" \
[--upstream-lifecycle-environment-label="My_Lifecycle_Environment"] \
[--upstream-content-view-label="My_Content_View"]
```

デフォルトのライフサイクル環境ラベルは **Library** です。デフォルトのコンテンツビューラベルは **Default_Organization_View** です。

3.4. エクスポートを使用してコンテンツを同期するように SATELLITE SERVER を設定する

ダウンロードの Satellite Server をエアギャップとしてデプロイした場合は、ネットワークからコンテンツを消費しようとしなないように、Satellite Server をそのように設定します。

手順

1. Satellite Web UI で、**コンテンツ > サブスクリプション** に移動します。
2. **マニフェストの管理** をクリックします。
3. **CDN Configuration** タブに切り替えます。
4. **Export Sync** タブを選択します。
5. **Update** をクリックします。

CLI 手順

1. SSH を使用して Satellite Server にログインします。
2. エクスポートを使用して CDN 設定を同期するように設定します。

```
# hammer organization configure-cdn --name="My_Organization" --type=export_sync
```

関連情報

エクスポートを使用したコンテンツの同期の詳細は、**コンテンツの管理** の [エクスポートとインポートを使用してコンテンツを同期する方法](#) を参照してください。

3.5. キックスタートリポジトリーのインポート

キックスタートリポジトリーは、コンテンツ ISO イメージでは提供されません。オフラインの Satellite でキックスタートリポジトリーを使用するには、使用する Red Hat Enterprise Linux のバージョンのバイナリー DVD ISO ファイルをダウンロードし、キックスタートファイルを Satellite にコピーする必要があります。

Red Hat Enterprise Linux 9 のキックスタートリポジトリーをインポートするには、[「Red Hat Enterprise Linux 9 のキックスタートリポジトリーのインポート」](#) を実行します。

Red Hat Enterprise Linux 8 のキックスタートリポジトリーをインポートするには、[「Red Hat Enterprise Linux 8 のキックスタートリポジトリーのインポート」](#) を実行します。

Red Hat Enterprise Linux 7 のキックスタートリポジトリーをインポートするには、[「Red Hat Enterprise Linux 7 のキックスタートリポジトリーのインポート」](#) を実行します。

3.5.1. Red Hat Enterprise Linux 9 のキックスタートリポジトリーのインポート

以下の手順を使用して、Red Hat Enterprise Linux 9 のキックスタートリポジトリーをインポートします。

手順

1. Red Hat カスタマーポータル access.redhat.com/downloads に移動し、ログインします。
2. **Red Hat Enterprise Linux** をクリックします。

3. 一覧から製品バリエーションと製品バージョンを選択します。(例: 製品バリエーション **Red Hat Enterprise Linux for x86_64** および製品バージョン **9.0**)
4. **Red Hat Enterprise Linux 9.0 Binary DVD**など、完全インストールイメージを見つけ、**Download Now** をクリックします。最小限の ISO を使用してホストをプロビジョニングすることはできないことに注意してください。
5. ダウンロードが完了したら、ISO イメージを Satellite Server にコピーします。
6. Satellite Server で、マウントポイントを作成し、そのロケーションに ISO イメージを一時的にマウントします。

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

rhel-binary-dvd.iso は、ISO イメージの名前に置き換えます。

7. Red Hat Enterprise Linux 9 AppStream および BaseOS Kickstart リポジトリのディレクトリを作成します。

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

8. ISO イメージから **kickstart** ファイルをコピーします。

```
# cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart
# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart
```

BaseOS の場合は、**/mnt/iso/images/** ディレクトリのコンテンツもコピーする必要があることに注意してください。

9. 次のエントリをリストファイルに追加します。

/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/listing ファイルで改行して **kickstart** を追加します。

/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/listing ファイルで改行して **kickstart** を追加します。

/var/www/html/pub/satellite-import/content/dist/rhel8/listing ファイルで改行してバージョン番号を追加します。たとえば、Red Hat Enterprise Linux 9.0 バイナリー ISO の場合は **9.0** を追加します。

10. ISO イメージから **.treeinfo** ファイルをコピーします。

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo
```

11. **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo** ファイルを開いて編集します。

12. **[general]** セクションで、以下の変更を加えます。
 - **packagedir = AppStream/Packages** を **packagedir = Packages** に変更します。
 - **repository = AppStream** を **repository = .** に変更します。
 - **variant = AppStream** を **variant = BaseOS** に変更します。
 - **variants = AppStream,BaseOS** を **variants = BaseOS** に変更します。
13. **[tree]** セクションで、**variants = AppStream,BaseOS** を **variants = BaseOS** に変更します。
14. **[variant-BaseOS]** セクションで、以下の変更を加えます。
 - **packages = BaseOS/Packages** を **packages = Packages** に変更します。
 - **repository = BaseOS** を **repository = .** に変更します。
15. **[media]** および **[variant-AppStream]** のセクションを削除します。
16. ファイルを保存してから閉じます。
17. **/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/baseos/kickstart/treeinfo** ファイルが以下の形式であることを確認します。

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b
```

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 9.0.0
```

```
[header]
type = productmd.treeinfo
version = 1.2
```

```
[images-x86_64]
efiboot.img = images/efiboot.img
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz
```

```

[images-xen]
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 9.0.0

[stage2]
mainimage = images/install.img

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = BaseOS

[variant-BaseOS]
id = BaseOS
name = BaseOS
packages = Packages
repository = .
type = variant
uid = BaseOS

```

18. `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo` ファイルを開いて編集します。
19. `[general]` セクションで、以下の変更を加えます。
 - `packagedir = AppStream/Packages` を `packagedir = Packages` に変更します。
 - `repository = AppStream` を `repository = .` に変更します。
 - `variants = AppStream,BaseOS` を `variants = AppStream` に変更します。
20. `[tree]` セクションで、`variants = AppStream,BaseOS` を `variants = AppStream` に変更します。
21. `[variant-AppStream]` セクションで、以下の変更を加えます。
 - `packages = AppStream/Packages` を `packages = Packages` に変更します。
 - `repository = AppStream` を `repository = .` に変更します。
22. ファイルから次のセクションを削除します: `[checksums]`、`[images-x86_64]`、`[images-xen]`、`[media]`、`[stage2]`、`[variant-BaseOS]`。
23. ファイルを保存してから閉じます。
24. `/var/www/html/pub/satellite-import/content/dist/rhel9/9.0/x86_64/appstream/kickstart/treeinfo` ファイルが以下の形式であることを確認します。

```

[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 9.0.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
version = 9.0.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 9.0.0

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = AppStream

[variant-AppStream]
id = AppStream
name = AppStream
packages = Packages
repository = .
type = variant
uid = AppStream

```

25. マウントされたバイナリー DVD ISO イメージを使用する予定がない場合は、ディレクトリーをアンマウントして削除します。

```

# umount /mnt/iso
# rmdir /mnt/iso

```

26. Satellite Web UI で、キックスタートリポジトリーを有効にします。

3.5.2. Red Hat Enterprise Linux 8 のキックスタートリポジトリーのインポート

以下の手順を使用して、Red Hat Enterprise Linux 8 のキックスタートリポジトリーをインポートします。

手順

1. Red Hat カスタマーポータル access.redhat.com/downloads に移動し、ログインします。
2. Red Hat Enterprise Linux をクリックします。

3. 一覧から製品バリエーションと製品バージョンを選択します。(例: 製品バリエーション Red Hat Enterprise Linux for x86_64 および製品バージョン 8.1)
4. Red Hat Enterprise Linux 8.1 Binary DVD など、完全インストールイメージを見つけ、**Download Now** をクリックします。
5. ダウンロードが完了したら、ISO イメージを Satellite Server にコピーします。
6. Satellite Server で、マウントポイントを作成し、そのロケーションに ISO イメージを一時的にマウントします。

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

rhel-binary-dvd.iso は、ISO イメージの名前に置き換えます。

7. Red Hat Enterprise Linux 8 AppStream および BaseOS Kickstart リポジトリのディレクトリを作成します。

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
```

8. ISO イメージから **kickstart** ファイルをコピーします。

```
# cp -a /mnt/iso/AppStream/* /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart
# cp -a /mnt/iso/BaseOS/* /mnt/iso/images/ /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart
```

BaseOS の場合は、**/mnt/iso/images/** ディレクトリ内のコンテンツもコピーする必要があることに注意してください。

9. 次のエントリをリストファイルに追加します。

/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/listing ファイルで改行して **kickstart** を追加します。

/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/listing ファイルで改行して **kickstart** を追加します。

/var/www/html/pub/satellite-import/content/dist/rhel8/listing ファイルで改行してバージョン番号を追加します。たとえば、Red Hat Enterprise Linux 8.1 バイナリー ISO の場合は **8.1** を追加します。

10. ISO イメージから **.treeinfo** ファイルをコピーします。

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo
```

11. **/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo** ファイルを開いて編集します。

12. **[general]** セクションで、以下の変更を加えます。
 - **packagedir = AppStream/Packages** を **packagedir = Packages** に変更します。
 - **repository = AppStream** を **repository = .** に変更します。
 - **variant = AppStream** を **variant = BaseOS** に変更します。
 - **variants = AppStream,BaseOS** を **variants = BaseOS** に変更します。
13. **[tree]** セクションで、**variants = AppStream,BaseOS** を **variants = BaseOS** に変更します。
14. **[variant-BaseOS]** セクションで、以下の変更を加えます。
 - **packages = BaseOS/Packages** を **packages = Packages** に変更します。
 - **repository = BaseOS** を **repository = .** に変更します。
15. **[media]** および **[variant-AppStream]** のセクションを削除します。
16. ファイルを保存してから閉じます。
17. **/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/baseos/kickstart/treeinfo** ファイルが以下の形式であることを確認します。

```
[checksums]
images/efiboot.img =
sha256:c01c18acc6778d6e66c8d0872bac59bfd7219ccf3cfa70a5c605c0fb37f33a83
images/install.img =
sha256:ddd08e5a5d92edee150f91ff4f12f39253eae72ff496465cf1b2766fe4a4df49
images/pxeboot/initrd.img =
sha256:a09a8ec89d485d71ed1bdad83584d6d816e67448221172d9aad97886cd70adca
images/pxeboot/vmlinuz =
sha256:6e523d7c3266e26c695923ab12b2873b16b0c61fb2e48ade608ad8998821584b
```

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = BaseOS
variants = BaseOS
version = 8.1.0
```

```
[header]
type = productmd.treeinfo
version = 1.2
```

```
[images-x86_64]
efiboot.img = images/efiboot.img
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz
```



```

[images-xen]
initrd = images/pxeboot/initrd.img
kernel = images/pxeboot/vmlinuz

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[stage2]
mainimage = images/install.img

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = BaseOS

[variant-BaseOS]
id = BaseOS
name = BaseOS
packages = Packages
repository = .
type = variant
uid = BaseOS

```

18. `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo` ファイルを開いて編集します。
19. `[general]` セクションで、以下の変更を加えます。
 - `packagedir = AppStream/Packages` を `packagedir = Packages` に変更します。
 - `repository = AppStream` を `repository = .` に変更します。
 - `variants = AppStream,BaseOS` を `variants = AppStream` に変更します。
20. `[tree]` セクションで、`variants = AppStream,BaseOS` を `variants = AppStream` に変更します。
21. `[variant-AppStream]` セクションで、以下の変更を加えます。
 - `packages = AppStream/Packages` を `packages = Packages` に変更します。
 - `repository = AppStream` を `repository = .` に変更します。
22. ファイルから次のセクションを削除します: `[checksums]`、`[images-x86_64]`、`[images-xen]`、`[media]`、`[stage2]`、`[variant-BaseOS]`。
23. ファイルを保存してから閉じます。
24. `/var/www/html/pub/satellite-import/content/dist/rhel8/8.1/x86_64/appstream/kickstart/treeinfo` ファイルが以下の形式であることを確認します。

```
[general]
; WARNING.0 = This section provides compatibility with pre-productmd treeinfos.
; WARNING.1 = Read productmd documentation for details about new format.
arch = x86_64
family = Red Hat Enterprise Linux
name = Red Hat Enterprise Linux 8.1.0
packagedir = Packages
platforms = x86_64,xen
repository = .
timestamp = 1571146127
variant = AppStream
variants = AppStream
version = 8.1.0

[header]
type = productmd.treeinfo
version = 1.2

[release]
name = Red Hat Enterprise Linux
short = RHEL
version = 8.1.0

[tree]
arch = x86_64
build_timestamp = 1571146127
platforms = x86_64,xen
variants = AppStream

[variant-AppStream]
id = AppStream
name = AppStream
packages = Packages
repository = .
type = variant
uid = AppStream
```

25. マウントされたバイナリー DVD ISO イメージを使用する予定がない場合は、ディレクトリーをアンマウントして削除します。

```
# umount /mnt/iso
# rmdir /mnt/iso
```

26. Satellite Web UI で、キックスタートリポジトリーを有効にします。

3.5.3. Red Hat Enterprise Linux 7 のキックスタートリポジトリーのインポート

以下の手順を使用して、Red Hat Enterprise Linux 7 のキックスタートリポジトリーをインポートします。

手順

1. Red Hat カスタマーポータル access.redhat.com/downloads に移動し、ログインします。
2. Red Hat Enterprise Linux をクリックします。

3. **Product Variant** リストの上にある **Switch to version 7 and below** をクリックします。
4. 一覧から製品バリエーションと製品バージョンを選択します。(例: 製品バリエーション **Red Hat Enterprise Linux for x86_64** および製品バージョン **7.9**)
5. **Red Hat Enterprise Linux 7.9 Binary DVD** など、完全インストールイメージを見つけ、**Download Now** をクリックします。
6. ダウンロードが完了したら、ISO イメージを Satellite Server にコピーします。
7. Satellite Server で、マウントポイントを作成し、そのロケーションに ISO イメージを一時的にマウントします。

```
# mkdir /mnt/iso
# mount -o loop rhel-binary-dvd.iso /mnt/iso
```

rhel-binary-dvd.iso は、ISO イメージの名前に置き換えます。

8. kickstart ディレクトリーを作成します。

```
# mkdir --parents /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
```

9. ISO イメージから **kickstart** ファイルをコピーします。

```
# cp -a /mnt/iso/* /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/
```

10. 次のエントリーをリストファイルに追加します。
/var/www/html/pub/satellite-import/content/dist/rhel/server/7/listing ファイルで改行してバージョン番号を追加します。たとえば、Red Hat Enterprise Linux 7.9 バイナリー ISO の場合は **7.9** を追加します。

/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/listing ファイルで改行してアーキテクチャーを追加します。(例: **x86_64**)

/var/www/html/pub/satellite-import/content/dist/rhel/server/7/7.9/x86_64/listing ファイルで改行して **kickstart** を追加します。

11. ISO イメージから **.treeinfo** ファイルをコピーします。

```
# cp /mnt/iso/.treeinfo /var/www/html/pub/satellite-
import/content/dist/rhel/server/7/7.9/x86_64/kickstart/treeinfo
```

12. マウントされたバイナリー DVD ISO イメージを使用する予定がない場合は、ディレクトリーをアンマウントして削除します。

```
# umount /mnt/iso
# rmdir /mnt/iso
```

13. Satellite Web UI で、キックスタートリポジトリーを有効にします。

3.6. SATELLITE CLIENT 6 リポジトリーの有効化と同期

Satellite Client 6 リポジトリは、Satellite に登録されているホスト用の **katello-host-tools** および **puppet** パッケージを提供します。Red Hat コンテンツ配信ネットワーク (CDN) から Satellite Server にリポジトリを定期的に同期し、ホスト上でリポジトリを有効にする必要があります。

3.6.1. Red Hat Enterprise Linux 9 と Red Hat Enterprise Linux 8 の Satellite Client 6 リポジトリの同期

Satellite Web UI の代わりに CLI を使用するには、Red Hat Enterprise Linux バージョンに関連する手順を参照してください。

- [Red Hat Enterprise Linux 9 の CLI 手順](#)
- [Red Hat Enterprise Linux 8 の CLI 手順](#)

手順

1. Satellite Web UI で、**Content** > **Sync Status** に移動します。
2. **Red Hat Enterprise Linux for x86_64**製品の横にある矢印をクリックして、利用可能なコンテンツを表示します。
3. **Red Hat Satellite Client 6 for RHEL 9 x86_64 RPMs**または **Red Hat Satellite Client 6 for RHEL 8 x86_64 RPMs** を選択します。
4. **Synchronize Now** をクリックします。

Red Hat Enterprise Linux 9 の CLI 手順

- Satellite Client 6 リポジトリを同期します。

```
# hammer repository synchronize \  
--name "Red Hat Satellite Client 6 for RHEL 9 x86_64 RPMs" \  
--organization "My_Organization" \  
--product "Red Hat Enterprise Linux for x86_64"
```

Red Hat Enterprise Linux 8 の CLI 手順

- Satellite Client 6 リポジトリを同期します。

```
# hammer repository synchronize \  
--name "Red Hat Satellite Client 6 for RHEL 8 x86_64 RPMs" \  
--organization "My_Organization" \  
--product "Red Hat Enterprise Linux for x86_64"
```

関連情報

- **hammer repository synchronize** コマンドの詳細を確認するには、**hammer repository synchronize --help** と入力してください。

3.6.2. Red Hat Enterprise Linux 7 と Red Hat Enterprise Linux 6 の Satellite Client 6 リポジトリの同期



注記

Red Hat Enterprise Linux 6 のリポジトリを同期するには、**Red Hat Enterprise Linux 延長ライフサイクルサポート (ELS) アドオン** サブスクリプションが必要です。詳細は、[Red Hat Enterprise Linux 延長ライフサイクルサポート \(ELS\) アドオン ガイド](#)を参照してください。

Satellite Web UI の代わりに CLI を使用するには、Red Hat Enterprise Linux バージョンに関連する手順を参照してください。

- [Red Hat Enterprise Linux 7 の CLI 手順](#)
- [Red Hat Enterprise Linux 6 の CLI 手順](#)

手順

1. Satellite Web UI で、**Content > Sync Status** に移動します。
2. **Red Hat Enterprise Linux Server** または **Red Hat Enterprise Linux Server - Extended Lifecycle Support** の横にある矢印をクリックします。
3. お使いのオペレーティングシステムのバージョンに応じて、**Red Hat Satellite Client 6 (for RHEL 7 Server) RPMs x86_64** または **Red Hat Satellite Client 6 for RHEL 6 Server - ELS RPMs x86_64** を選択します。
4. **Synchronize Now** をクリックします。

Red Hat Enterprise Linux 7 の CLI 手順

- Satellite Client 6 リポジトリを同期します。

```
# hammer repository synchronize \
--async \
--name "Red Hat Satellite Client 6 for RHEL 7 Server RPMs x86_64" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server"
```

Red Hat Enterprise Linux 6 の CLI 手順

- Satellite Client 6 リポジトリを同期します。

```
# hammer repository synchronize \
--async \
--name "Red Hat Satellite Client 6 for RHEL 6 Server - ELS RPMs x86_64" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server - Extended Lifecycle Support"
```

関連情報

- **hammer repository synchronize** コマンドの詳細を確認するには、**hammer repository synchronize --help** と入力してください。

3.6.3. Red Hat Enterprise Linux 9 と Red Hat Enterprise Linux 8 の Satellite Client 6 リポジトリの有効化

Satellite Web UI の代わりに CLI を使用するには、Red Hat Enterprise Linux バージョンに関連する手順を参照してください。

- [Red Hat Enterprise Linux 9 の CLI 手順](#)
- [Red Hat Enterprise Linux 8 の CLI 手順](#)

前提条件

- Satellite Server に必要なコンテンツの ISO イメージすべてをインポートするようにしてください。

手順

1. Satellite Web UI で、**コンテンツ > Red Hat リポジトリ**に移動します。
2. Available Repositories ペインで、**Recommended Repositories** を有効にして、リポジトリのリストを取得します。
3. **Red Hat Satellite Client 6 for RHEL 9 x86_64 (RPMs)**または **Red Hat Satellite Client 6 for RHEL 8 x86_64 (RPMs)** をクリックして、リポジトリセットをデプロイします。
4. **x86_64** アーキテクチャーの場合、**+** アイコンをクリックしてリポジトリを有効にします。**Satellite Client 6**の項目が表示されていない場合は、カスタマーポータルから取得した Red Hat サブスクリプションマニフェストにその項目が含まれないことが原因として考えられます。この問題を修正するには、カスタマーポータルにログインし、これらのリポジトリを追加し、Red Hat サブスクリプションマニフェストをダウンロードして、Satellite にインポートします。詳細は、**コンテンツの管理**の [Red Hat サブスクリプションの管理](#) を参照してください。

ホストで実行している Red Hat Enterprise Linux の各サポート対象メジャーバージョンに対して Satellite Client 6 リポジトリを有効にします。Red Hat リポジトリの有効後に、このリポジトリの製品が自動的に作成されます。

Red Hat Enterprise Linux 9 の CLI 手順

- Satellite Client 6 リポジトリを有効にします。

```
# hammer repository-set enable \  
--basearch="x86_64" \  
--name "Red Hat Satellite Client 6 for RHEL 9 x86_64 (RPMs)" \  
--organization "My_Organization" \  
--product "Red Hat Enterprise Linux for x86_64"
```

Red Hat Enterprise Linux 8 の CLI 手順

- Satellite Client 6 リポジトリを有効にします。

```
# hammer repository-set enable \  
--basearch="x86_64" \  
--name "Red Hat Satellite Client 6 for RHEL 8 x86_64 (RPMs)" \  
--organization "My_Organization" \  
--product "Red Hat Enterprise Linux for x86_64"
```

関連情報

- **hammer repository-set enable** コマンドの詳細を確認するには、**hammer repository-set enable --help** と入力してください。

3.6.4. Red Hat Enterprise Linux 7 と Red Hat Enterprise Linux 6 の Satellite Client 6 リポジトリの有効化



注記

Red Hat Enterprise Linux 6 のリポジトリを有効にするには、**Red Hat Enterprise Linux 延長ライフサイクルサポート (ELS) アドオン** サブスクリプションが必要です。詳細は、[Red Hat Enterprise Linux 延長ライフサイクルサポート \(ELS\) アドオン](#) ガイドを参照してください。

Satellite Web UI の代わりに CLI を使用するには、Red Hat Enterprise Linux バージョンに関連する手順を参照してください。

- [Red Hat Enterprise Linux 7 の CLI 手順](#)
- [Red Hat Enterprise Linux 6 の CLI 手順](#)

前提条件

- Satellite Server に必要なコンテンツの ISO イメージすべてをインポートするようにしてください。
 1. Satellite Web UI で、**コンテンツ > Red Hat リポジトリ** に移動します。
 2. **Available Repositories** ペインで、**Recommended Repositories** を有効にして、リポジトリのリストを取得します。
 3. **Available Repositories** ペインで、**Satellite Client 6 (for RHEL 7 Server) (RPMs)** または **Satellite Client 6 (for RHEL 6 Server - ELS) (RPMs)** をクリックしてリポジトリセットをデプロイします。
Satellite Client 6 の項目が表示されていない場合は、カスタマーポータルから取得した Red Hat サブスクリプションmanifestにその項目が含まれないことが原因として考えられます。この問題を修正するには、カスタマーポータルにログインし、これらのリポジトリを追加し、Red Hat サブスクリプションmanifestをダウンロードして、Satellite にインポートします。詳細は、**コンテンツの管理** の [Red Hat サブスクリプションの管理](#) を参照してください。
 4. **x86_64** アーキテクチャーの場合、**+** アイコンをクリックしてリポジトリを有効にします。ホストで実行している Red Hat Enterprise Linux の各サポート対象メジャーバージョンに対して Satellite Client 6 リポジトリを有効にします。Red Hat リポジトリの有効後に、このリポジトリの製品が自動的に作成されます。

Red Hat Enterprise Linux 7 の CLI 手順

- Satellite Client 6 リポジトリを有効にします。

```
# hammer repository-set enable \
--basearch="x86_64" \
--name "Red Hat Satellite Client 6 (for RHEL 7 Server) (RPMs)" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server"
```

■

Red Hat Enterprise Linux 6 の CLI 手順

- Satellite Client 6 リポジトリを有効にします。

```
# hammer repository-set enable \
--basearch="x86_64" \
--name "Red Hat Satellite Client 6 (for RHEL 6 Server - ELS) (RPMs)" \
--organization "My_Organization" \
--product "Red Hat Enterprise Linux Server - Extended Lifecycle Support"
```

関連情報

- **hammer repository-set enable** コマンドの詳細を確認するには、**hammer repository-set enable --help** と入力してください。

3.7. SATELLITE SERVER でプルクライアントのリモート実行を設定する

デフォルトでは、リモート実行はスクリプトプロバイダーのトランスポートメカニズムとして SSH を使用します。ただし、リモート実行にはプルベースのトランスポート機能があり、インフラストラクチャーが Satellite からホストへの発信接続を禁止している場合に使用できます。

これは、Satellite 上の **pull-mqtt** モードとホスト上で実行されるプルクライアントの組み合わせで構成されます。



注記

pull-mqtt モードは、スクリプトプロバイダーでのみ機能します。Ansible およびその他のプロバイダーは、引き続きデフォルトのトランスポート設定を使用します。

Satellite Server で **pull-mqtt** モードを使用するには、以下の手順に従います。

手順

1. Satellite Server でプルベースのトランスポートを有効にします。

```
# satellite-installer --foreman-proxy-plugin-remote-execution-script-mode pull-mqtt
```

2. ポート 1883 で MQTT サービスを許可するようにファイアウォールを設定します。

```
# firewall-cmd --add-service=mqtt
```

pull-mqtt モードでは、ホストがその登録先の Satellite または Capsule Server へのジョブ通知をサブスクライブします。したがって、Satellite Server がリモート実行ジョブを同じ Satellite Server または Capsule Server に送信することを確認することを推奨します。

3. 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

4. Satellite Web UI で、**Administer > Settings** に移動します。

5. **Content** タブで、**リモート実行用に Capsule 経由で登録推奨**の値を **Yes** に設定します。

Satellite にプルベースのトランスポートを設定したら、各ホストでも設定する必要があります。詳細は、**ホストの管理** の **リモート実行のトランスポートモード** を参照してください。

3.8. ホストでの電源管理の有効化

Intelligent Platform Management Interface (IPMI) または類似するプロトコルを使用してホストで電源管理タスクを実行するには、Satellite Server でベースボード管理コントローラー (BMC) モジュールを有効にする必要があります。

前提条件

- すべてのホストに BMC タイプのネットワークインターフェイスがある。Satellite Server はこの NIC を使用して、適切な認証情報をホストに渡します。詳細は、**ホストの管理** の **ベースボード管理コントローラー (BMC) インターフェイスの追加** を参照してください。

手順

- BMC を有効にするには、以下のコマンドを入力します。

```
# satellite-installer --foreman-proxy-bmc "true" \  
--foreman-proxy-bmc-default-provider "freeipmi"
```

3.9. DNS、DHCP、および TFTP の設定

DNS、DHCP、TFTP は、Satellite 環境内で一元的に管理することも、Satellite 上でそれらのメンテナンスを無効にした後で個別に管理することもできます。DNS、DHCP、TFTP を Satellite 環境の外部で実行することもできます。

3.9.1. Satellite Server での DNS、DHCP および TFTP の設定

DNS、DHCP および TFTP サービスを Satellite Server で設定するには、お使いの環境に適したオプションを指定して **satellite-installer** コマンドを使用します。

設定を変更するには、**satellite-installer** コマンドを再び実行する必要があります。コマンドは複数回実行でき、実行するたびにすべての設定ファイルが変更された値で更新されます。

前提条件

- 以下の情報が利用可能であることを確認する。
 - DHCP IP アドレス範囲
 - DHCP ゲートウェイ IP アドレス
 - DHCP ネームサーバー IP アドレス
 - DNS 情報
 - TFTP サーバー名
- ネットワークの変更の場合は、可能な限り、IP アドレスの代わりに FQDN を使用します。

- ネットワーク管理者に連絡して正しい設定が行われていることを確認する。

手順

- お使いの環境に適したオプションで、**satellite-installer** コマンドを入力してください。以下の例では、完全なプロビジョニングサービスの設定を示しています。

```
# satellite-installer \
--foreman-proxy-dns true \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-reverse 2.0.192.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-range "192.0.2.100 192.0.2.150" \
--foreman-proxy-dhcp-gateway 192.0.2.1 \
--foreman-proxy-dhcp-nameservers 192.0.2.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername 192.0.2.3
```

プロンプトに表示される **satellite-installer** コマンドの進行状況を監視できます。/var/log/foreman-installer/satellite.log でログを表示できます。

関連情報

- **satellite-installer --scenario satellite** コマンドの詳細を確認するには、**satellite-installer --scenario satellite --help** と入力してください。

3.9.2. 管理対象外のネットワークの DNS、DHCP、および TFTP を無効にする

TFTP、DHCP および DNS サービスを手動で管理する場合には、Satellite がオペレーティングシステム上でこれらのサービスを管理しないようにし、オーケストレーションを無効にして、DHCP および DNS バリデーションエラーを回避する必要があります。ただし、Satellite ではオペレーティングシステムのバックエンドサービスは削除されません。

手順

1. Satellite Server で以下のコマンドを入力します。

```
# satellite-installer --foreman-proxy-dhcp false \
--foreman-proxy-dns false \
--foreman-proxy-tftp false
```

2. Satellite Web UI で、インフラストラクチャー > Capsule に移動し、サブネットを選択します。
3. Capsules タブで、DHCP Capsule、TFTP Capsule、および 逆引き DNS Capsule を選択します。
4. Satellite Web UI で、インフラストラクチャー > ドメイン に移動し、ドメインを選択します。
5. DNS Capsule フィールドの内容を消去します。
6. オプション: サードパーティーが提供する DHCP サービスを使用する場合は、以下のオプションを渡すように DHCP サーバーを設定します。

Option 66: IP address of Satellite or Capsule

Option 67: /pxelinux.0

DHCP オプションの詳細は [RFC 2132](#) を参照してください。



注記

Satellite は、Capsule が該当するサブネットとドメインに設定されていない場合にオーケストレーションを実行しません。Capsule の関連付けを有効または無効にした場合に、想定のリコードと設定ファイルが存在しないと、既存のホストのオーケストレーションコマンドが失敗することがあります。オーケストレーションを有効にするために Capsule を関連付ける場合は、今後、ホストの削除に失敗しないように、既存の Satellite ホストに対して必要な DHCP レコード、DNS レコード、TFTP ファイルが所定の場所にあることを確認します。

3.9.3. 関連情報

- DNS、DHCP、TFTP を外部で設定する方法の詳細は、[4章 外部サービスを使用した Satellite Server の設定](#) を参照してください。
- DHCP、DNS、および TFTP サービスの設定の詳細は、[ホストのプロビジョニングの ネットワークサービスの設定](#) を参照してください。

3.10. SATELLITE SERVER での送信メールの設定

Satellite Server からメールメッセージを送信するには、SMTP サーバーまたは **sendmail** コマンドのいずれかを使用できます。

前提条件

- スпам対策保護またはグレイリスティング機能を備えた SMTP サーバーの一部で、問題が発生することが知られています。このようなサービスでの送信メールの設定には、リレー用に Satellite Server に vanilla SMTP サービスをインストールして設定するか、代わりに **sendmail** コマンドを使用します。

手順

1. Satellite Web UI で、**Administer > Settings** に移動します。
2. **Email** タブをクリックして、希望する配信方法に一致する設定オプションを設定します。変更は即座に反映されます。
 - a. 以下の例は、SMTP サーバーを使用する場合の設定オプションの例を示しています。

表3.1 配信方法に SMTP サーバーを使用する例

名前	値例
配信方法	SMTP
SMTP アドレス	smtp.example.com
SMTP 認証	ログイン

名前	値例
SMTP HELO/EHLO ドメイン	example.com
SMTP パスワード	パスワード
SMTP ポート	25
SMTP ユーザー名	user@example.com

SMTP ユーザー名 と **SMTP パスワード** では、SMTP サーバーのログイン認証情報を指定します。

- b. 以下の例では、**gmail.com** が SMTP サーバーとして使用されています。

表3.2 gmail.com を SMTP サーバーとして使用する例

名前	値例
配信方法	SMTP
SMTP アドレス	smtp.gmail.com
SMTP 認証	plain
SMTP HELO/EHLO ドメイン	smtp.gmail.com
SMTP enable StartTLS auto	あり
SMTP パスワード	パスワード
SMTP ポート	587
SMTP ユーザー名	user@gmail.com

- c. 以下の例では、**sendmail** コマンドが配信方法として使用されています。

表3.3 配信方法に sendmail を使用する例

名前	値例
配信方法	Sendmail
Sendmail の場所	/usr/sbin/sendmail
Sendmail の引数	-i

セキュリティ上の理由から、Sendmail の場所と Sendmail 引数の設定はどちらも読み取り専用であり、`/etc/foreman/settings.yaml` でのみ設定できます。現在、両方の設定を `satellite-installer` で設定することはできません。詳細は、`sendmail1` の man ページを参照してください。

3. TLS 認証を使用する SMTP サーバーで電子メールを送信する場合は、以下のいずれかの手順を実行してください。

- SMTP サーバーの CA 証明書を信頼済みとしてマークします。このようにマークするには、Satellite Server で以下のコマンドを実行します。

```
# cp mailca.crt /etc/pki/ca-trust/source/anchors/
# update-ca-trust enable
# update-ca-trust
```

ここで、`mailca.crt` は SMTP サーバーの CA 証明書です。

- または、Satellite Web UI で、**SMTP enable StartTLS auto** オプションを **No** に設定します。
4. **Test email** をクリックしてユーザーのメールアドレスにテストメッセージを送信し、設定が機能していることを確認します。メッセージの送信に失敗した場合、Satellite Web UI はエラーを表示します。詳細については、`/var/log/foreman/production.log` のログを確認してください。

関連情報

- 個々のユーザーまたはユーザーグループに対するメール通知の設定に関する詳細は、**Red Hat Satellite の管理** の [メール通知の設定](#) を参照してください。

3.11. カスタムの SSL 証明書を使用した SATELLITE SERVER の設定

デフォルトでは、Red Hat Satellite は自己署名の SSL 証明書を使用して、Satellite Server、外部の Capsule Server および全ホストの間で暗号化した通信ができるようにします。Satellite の自己署名証明書を使用できない場合には、外部の認証局 (CA) で署名した SSL 証明書を使用するように Satellite Server を設定できます。

カスタム SSL 証明書を使用して Red Hat Satellite を設定する場合は、次の要件を満たす必要があります。

- SSL 証明書には、Privacy-Enhanced Mail (PEM) エンコードを使用する必要があります。
- Satellite Server と Capsule Server の両方に同じ SSL 証明書を使用しないでください。
- Satellite Server と Capsule Server の証明書には同じ CA が署名する必要があります。
- SSL 証明書は CA 証明書ではありません。
- SSL 証明書には、共通名 (CN) と一致するサブジェクト代替名 (SAN) エントリーが含まれている必要があります。
- SSL 証明書は、鍵用途エクステンションを使用した鍵暗号化が許可されている必要があります。
- SSL 証明書は、CN に短縮名を使用することはできません。
- 秘密鍵にパスフレーズを設定しないでください。

カスタムの証明書で Satellite Server を設定するには、以下の手順を実行します。

1. 「Satellite Server 用のカスタム SSL 証明書の作成」
2. 「Satellite Server へのカスタム SSL 証明書のデプロイ」
3. 「ホストへのカスタム SSL 証明書のデプロイ」
4. Satellite Server に外部の Capsule Server を登録した場合には、カスタムの SSL 証明書を使用して設定します。詳細は、**Capsule Server のインストール**の [カスタム SSL 証明書を使用した Capsule Server の設定](#) を参照してください。

3.11.1. Satellite Server 用のカスタム SSL 証明書の作成

この手順を使用して、Satellite Server 用にカスタムの SSL 証明書を作成します。Satellite Server 用のカスタムの SSL 証明書がある場合にはこの手順は省略してください。

手順

1. ソースの証明書ファイルすべてを保存するには、**root** ユーザーだけがアクセスできるディレクトリを作成します。

```
# mkdir /root/satellite_cert
```

2. 証明書署名要求 (CSR) に署名する秘密鍵を作成します。
秘密鍵は暗号化する必要がないことに注意してください。パスワードで保護された秘密鍵を使用する場合は、秘密鍵のパスワードを削除します。

この Satellite Server の秘密鍵がすでにある場合は、この手順を省略します。

```
# openssl genrsa -out /root/satellite_cert/satellite_cert_key.pem 4096
```

3. CSR 用の **/root/satellite_cert/openssl.cnf** 設定ファイルを作成して、以下のコンテンツを追加します。

```
[ req ]
req_extensions = v3_req
distinguished_name = req_distinguished_name
prompt = no

[ req_distinguished_name ]
commonName = satellite.example.com

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
subjectAltName = @alt_names

[ alt_names ]
DNS.1 = satellite.example.com
```

4. オプション: CSR に識別名 (DN) の詳細を追加する場合は、**[req_distinguished_name]** セクションに次の情報を追加します。

-

```
[req_distinguished_name]
CN = satellite.example.com
countryName = My_Country_Name ①
stateOrProvinceName = My_State_Or_Province_Name ②
localityName = My_Locality_Name ③
organizationName = My_Organization_Or_Company_Name
organizationalUnitName = My_Organizational_Unit_Name ④
```

- ① 2文字コード
- ② 名前
- ③ フルネーム (例: ニューヨーク)
- ④ 証明書を担当する部門 (例: IT 部門)

5. CSR を生成します。

```
# openssl req -new \
-key /root/satellite_cert/satellite_cert_key.pem \ ①
-config /root/satellite_cert/openssl.cnf \ ②
-out /root/satellite_cert/satellite_cert_csr.pem ③
```

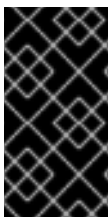
- ① 秘密鍵へのパス
- ② 設定ファイルへのパス
- ③ 生成する CSR へのパス

6. 認証局 (CA) に証明書署名要求を送信します。Satellite Server と Capsule Server の証明書には同じ CA が署名する必要があります。

要求を送信する場合は、証明書の有効期限を指定してください。証明書要求の送信方法にはさまざまなものがあるため、推奨される方法について CA にお問い合わせください。要求すると、CA バンドルと署名済み証明書を別々のファイルで受け取るようになります。

3.11.2. Satellite Server へのカスタム SSL 証明書のデプロイ

この手順を使用して、Satellite Server が、認証局で署名されたカスタムの SSL 署名書を使用するように設定します。**katello-certs-check** コマンドは、入力した証明書ファイルを検証して、Satellite Server にカスタムの SSL 証明書をデプロイするのに必要なコマンドを返します。



重要

SSL 証明書や .tar バンドルを **/tmp** や **/var/tmp** ディレクトリーに保存しないでください。オペレーティングシステムは、これらのディレクトリーからファイルを定期的に削除します。その結果、機能の有効化または Satellite Server のアップグレード中に、**satellite-installer** の実行が失敗します。

手順

1. カスタムの SSL 証明書入力ファイルを検証します。**katello-certs-check** コマンドが正しく実行されるには、証明書の共通名 (CN) が Satellite Server の FQDN と一致する必要があることに注意してください。

```
# katello-certs-check \
-c /root/satellite_cert/satellite_cert.pem \ ①
-k /root/satellite_cert/satellite_cert_key.pem \ ②
-b /root/satellite_cert/ca_cert_bundle.pem ③
```

- ① 認証局が署名した Satellite Server の証明書ファイルへのパス
- ② Satellite Server 証明書の署名に使用された秘密鍵へのパス。
- ③ 認証局バンドルへのパス

このコマンドに成功すると、2つの **satellite-installer** コマンドが返されます。1つは、Satellite Server に証明書をデプロイするのに使用する必要があります。

katello-certs-check の出力例

```
Validation succeeded.
```

To install the Red Hat Satellite Server with the custom certificates, run:

```
satellite-installer --scenario satellite \
--certs-server-cert "/root/satellite_cert/satellite_cert.pem" \
--certs-server-key "/root/satellite_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/satellite_cert/ca_cert_bundle.pem"
```

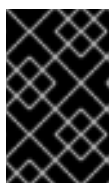
To update the certificates on a currently running Red Hat Satellite installation, run:

```
satellite-installer --scenario satellite \
--certs-server-cert "/root/satellite_cert/satellite_cert.pem" \
--certs-server-key "/root/satellite_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/satellite_cert/ca_cert_bundle.pem" \
--certs-update-server --certs-update-server-ca
```

/root/ssl-build にアクセスしたり変更したりしないでください。

2. 要件に合わせて **katello-certs-check** コマンドの出力から、**satellite-installer** コマンドを入力し、カスタムの SSL 証明書で新しい Satellite をインストールするか、現在実行中の Satellite の証明書を更新します。

実行するコマンドが不明な場合には、**/etc/foreman-installer/scenarios.d/.installed** が存在するかをチェックし、Satellite がインストールされていることが確認できます。ファイルが存在する場合には、2 番目の **satellite-installer** コマンドを実行すると証明書が更新されます。



重要

証明書をデプロイした後、**satellite-installer** には証明書アーカイブファイルが必要になります。変更したり削除したりしないでください。Satellite Server のアップグレード時などに必要です。

3. Satellite Server にネットワークでアクセスできるコンピューターで、この URL (**https://satellite.example.com**) に移動します。

4. ブラウザーで、証明書の詳細を表示して、デプロイした証明書を確認します。

3.11.3. ホストへのカスタム SSL 証明書のデプロイ

カスタム SSL 証明書を使用するように Satellite を設定したら、Satellite に登録されているホストに証明書をデプロイする必要があります。

手順

- 各ホストの SSL 証明書を更新します。

```
# dnf install http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

3.12. SATELLITE での外部データベースの使用

Red Hat Satellite のインストールプロセスの一部として、`satellite-installer` コマンドは PostgreSQL のデータベースを Satellite と同じサーバー上にインストールします。Satellite のデプロイメントによっては、デフォルトのローカルにあるデータベースの代わりに外部データベースを使用すると、サーバーの負荷が軽減される場合があります。

Red Hat では、外部データベースのメンテナンスのサポートやそのためのツールは提供していません。これにはバックアップ、アップグレード、データベースのチューニングが含まれます。外部データベースをサポートし、管理する自社のデータベース管理者が必要です。

Satellite 用に外部データベースを作成して使用するには、以下の手順を実行します。

1. 「[外部データベース用のホストの準備](#)」。外部データベースをホストするように Red Hat Enterprise Linux 8 サーバーを準備します。
2. 「[PostgreSQL のインストール](#)」。Satellite、Candlepin、Pulp のデータベースを使用して PostgreSQL を準備し、それらを所有する専用ユーザーを配置します。
3. 「[外部データベースを使用するための Satellite Server の設定](#)」。新規データベースを参照するように `satellite-installer` のパラメーターを編集し、`satellite-installer` を実行します。

3.12.1. 外部データベースとして PostgreSQL を使用する際の注意点

Foreman、Katello、および Candlepin は PostgreSQL データベースを使用します。PostgreSQL を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判断してください。Satellite は PostgreSQL バージョン 12 をサポートします。

外部 PostgreSQL の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- PostgreSQL データベースで `shared_buffers` を高い値に設定しても、Satellite 上の他のサービスの妨げるリスクがありません。
- Satellite 操作にマイナスの影響をもたらすことなく PostgreSQL サーバーのシステムを調整する柔軟性が得られます。

外部 PostgreSQL のデメリット

- デプロイメントの複雑性が増し、問題解決がより困難になります。

- 外部 PostgreSQL サーバーの場合は、パッチおよびメンテナンス対象に新たなシステムが加わることとなります。
- Satellite または PostgreSQL データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite Server とデータベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出ます。

お使いの Satellite 上の PostgreSQL データベースが原因でパフォーマンスの低下が生じている可能性がある場合は、[Satellite 6: How to enable postgres query logging to detect slow running queries](#) を参照して時間のかかっているクエリーがあるかどうか判定します。1秒以上かかるクエリーがある場合は、通常、大規模インストールのパフォーマンスが原因であることが多く、外部データベースに移行しても問題解決が期待できません。時間のかかっているクエリーがある場合は、Red Hat サポートチームまでお問い合わせください。

3.12.2. 外部データベース用のホストの準備

新しくプロビジョニングされたシステムに最新の Red Hat Enterprise Linux 8 をインストールして、外部データベースをホストします。

Red Hat Enterprise Linux のサブスクリプションでは、外部データベースと Satellite を併用する場合に、正しいサービスレベルアグリーメントが提供されません。外部データベースに使用するベースオペレーティングシステムにも、Satellite サブスクリプションをアタッチする必要があります。

前提条件

- 準備したホストが Satellite の [ストレージ要件](#) を満たしている。

手順

1. [Satellite Infrastructure サブスクリプションの割り当て](#) の手順に従って、Satellite サブスクリプションをサーバーに割り当てます。
2. すべてのリポジトリを無効にし、以下のリポジトリのみを有効にします。

```
# subscription-manager repos --disable '*'
# subscription-manager repos \
--enable=satellite-6.15-for-rhel-8-x86_64-rpms \
--enable=satellite-maintenance-6.15-for-rhel-8-x86_64-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms
```

3. 次のモジュールを有効にします。

```
# dnf module enable satellite:el8
```



注記

モジュール **satellite:el8** を有効にすると、**postgresql:10** および **ruby:2.5** との競合について警告が表示されます。これは、これらのモジュールが Red Hat Enterprise Linux 8 でデフォルトのモジュールバージョンに設定されているためです。モジュール **satellite:el8** には、モジュール **postgresql:12** および **ruby:2.7** への依存関係があり、**satellite:el8** モジュールで有効になります。これらの警告はインストールプロセスの失敗の原因にはならないため、安全に無視できます。Red Hat Enterprise Linux 8 のモジュールとライフサイクルストリームの詳細は、[Red Hat Enterprise Linux Application Streams のライフサイクル](#) を参照してください。

3.12.3. PostgreSQL のインストール

インストール可能な PostgreSQL は、内部データベースのインストール中に **satellite-installer** ツールでインストールされたものと同じバージョンの PostgreSQL のみになります。Satellite は PostgreSQL バージョン 12 をサポートします。

手順

1. PostgreSQL をインストールするには、以下のコマンドを入力します。

```
# dnf install postgresql-server postgresql-eva postgresql-contrib
```

2. PostgreSQL を初期化するには、以下のコマンドを入力します。

```
# postgresql-setup initdb
```

3. **/var/lib/pgsql/data/postgresql.conf** ファイルで以下を行います。

```
# vi /var/lib/pgsql/data/postgresql.conf
```

Satellite で機能するには、外部 PostgreSQL のデフォルト設定を調整する必要があることに注意してください。基本的に推奨される外部データベース設定の調整は次のとおりです。

- checkpoint_completion_target: 0.9
- max_connections: 500
- shared_buffers: 512MB
- work_mem: 4MB

4. **#** を削除して、着信接続をリッスンするようにします。

```
listen_addresses = '*'
```

5. **/var/lib/pgsql/data/pg_hba.conf** ファイルを編集します。

```
# vi /var/lib/pgsql/data/pg_hba.conf
```

6. 以下の行をファイルに追加します。

```
host all all Satellite_ip/32 md5
```

- PostgreSQL サービスを起動し、有効にするには、以下のコマンドを実行します。

```
# systemctl enable --now postgresql
```

- 外部 PostgreSQL サーバーで `postgresql` ポートを開きます。

```
# firewall-cmd --add-service=postgresql
```

- 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

- `postgres` ユーザーに切り替え、PostgreSQL クライアントを起動します。

```
$ su - postgres -c psql
```

- 3つのデータベースと専用のロールを作成します。1つは Satellite 用、1つは Candlepin 用、もう1つは Pulp 用です。

```
CREATE USER "foreman" WITH PASSWORD 'Foreman_Password';
CREATE USER "candlepin" WITH PASSWORD 'Candlepin_Password';
CREATE USER "pulp" WITH PASSWORD 'Pulpcore_Password';
CREATE DATABASE foreman OWNER foreman;
CREATE DATABASE candlepin OWNER candlepin;
CREATE DATABASE pulpcore OWNER pulp;
```

- Pulp データベースに接続します。

```
postgres=# \c pulpcore
You are now connected to database "pulpcore" as user "postgres".
```

- `hstore` エクステンションを作成します。

```
pulpcore=# CREATE EXTENSION IF NOT EXISTS "hstore";
CREATE EXTENSION
```

- `postgres` ユーザーをログアウトします。

```
# \q
```

- Satellite Server から、データベースにアクセスできることをテストします。接続に成功した場合には、コマンドは **1** を返します。

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com -p 5432 -U
foreman -d foreman -c "SELECT 1 as ping"
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U
candlepin -d candlepin -c "SELECT 1 as ping"
# PGPASSWORD='Pulpcore_Password' psql -h postgres.example.com -p 5432 -U pulp -
d pulpcore -c "SELECT 1 as ping"
```

3.12.4. 外部データベースを使用するための Satellite Server の設定

satellite-installer コマンドを使用して Satellite が外部の PostgreSQL データベースに接続するように設定します。

前提条件

- Red Hat Enterprise Linux サーバーに PostgreSQL データベースをインストールおよび設定していること。

手順

1. Satellite の外部データベースを設定するには以下のコマンドを入力します。

```
# satellite-installer --scenario satellite \  
--foreman-db-database foreman \  
--foreman-db-host postgres.example.com \  
--foreman-db-manage false \  
--foreman-db-password Foreman_Password \  
--foreman-proxy-content-pulpcore-manage-postgresql false \  
--foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \  
--foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \  
--foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password \  
--foreman-proxy-content-pulpcore-postgresql-user pulp \  
--katello-candlepin-db-host postgres.example.com \  
--katello-candlepin-db-name candlepin \  
--katello-candlepin-db-password Candlepin_Password \  
--katello-candlepin-manage-db false
```

これらの外部データベースに対して Secure Sockets Layer (SSL) プロトコルを有効にするには、次のオプションを追加します。

```
--foreman-db-root-cert <path_to_CA>  
--foreman-db-sslmode verify-full  
--foreman-proxy-content-pulpcore-postgresql-ssl true  
--foreman-proxy-content-pulpcore-postgresql-ssl-root-ca <path_to_CA>  
--katello-candlepin-db-ssl true  
--katello-candlepin-db-ssl-ca <path_to_CA>  
--katello-candlepin-db-ssl-verify true
```

第4章 外部サービスを使用した SATELLITE SERVER の設定

Satellite Server で DNS、DHCP、および TFTP サービスを設定しない場合は、このセクションを使用して、Satellite Server が外部 DNS、DHCP、および TFTP サービスと連携するように設定します。

4.1. 外部 DNS を使用した SATELLITE SERVER の設定

外部 DNS を使用して Satellite Server を設定できます。Satellite Server は **nsupdate** ユーティリティー-を使用して、リモートサーバーで DNS レコードを更新します。

変更を永続的に保存するには、お使いの環境に適したオプションを指定して、**satellite-installer** コマンドを入力する必要があります。

前提条件

- 外部 DNS サーバーが設定されている必要がある。
- このガイドは、既存のインストールがあることを前提としています。

手順

1. 外部 DNS サーバーの **/etc/rndc.key** ファイルを Satellite Server にコピーします。

```
# scp root@dns.example.com:/etc/rndc.key /etc/foreman-proxy/rndc.key
```

2. 所有者、パーミッション、SELinux コンテキストを設定します。

```
# restorecon -v /etc/foreman-proxy/rndc.key
# chown -v root:foreman-proxy /etc/foreman-proxy/rndc.key
# chmod -v 640 /etc/foreman-proxy/rndc.key
```

3. **nsupdate** ユーティリティーをテストするには、ホストをリモートで追加します。

```
# echo -e "server DNS_IP_Address\n \
update add aaa.example.com 3600 IN A Host_IP_Address\n \
send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
# nslookup aaa.example.com DNS_IP_Address
# echo -e "server DNS_IP_Address\n \
update delete aaa.example.com 3600 IN A Host_IP_Address\n \
send\n" | nsupdate -k /etc/foreman-proxy/rndc.key
```

4. **satellite-installer** コマンドを入力して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/dns.yml** ファイルに加えます。

```
# satellite-installer --foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="DNS_IP_Address" \
--foreman-proxy-keyfile=/etc/foreman-proxy/rndc.key
```

5. Satellite Web UI で、**Infrastructure > Capsules** に移動します。
6. Satellite Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。

7. DNS サービスに適切なサブネットとドメインを関連付けます。

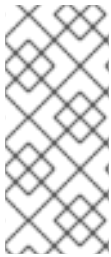
4.2. 外部 DHCP を使用した SATELLITE SERVER の設定

外部の DHCP で Satellite Server を設定するには、以下の手順を実行します。

1. 「[Satellite Server で使用する外部 DHCP サーバーの設定](#)」
2. 「[外部 DHCP サーバーを使用した Satellite Server の設定](#)」

4.2.1. Satellite Server で使用する外部 DHCP サーバーの設定

Red Hat Enterprise Linux を実行する外部の DHCP サーバーを Satellite Server で使用できるように設定するには、ISC DHCP Service と Berkeley Internet Name Domain (BIND) ユーティリティーパッケージをインストールする必要があります。また、DHCP 設定とリースファイルを Satellite Server と共有する必要があります。この手順の例では、分散型の Network File System (NFS) プロトコルを使用して DHCP 設定とリースファイルを共有します。



注記

外部の DHCP サーバーとして dnsmasq を使用する場合には、**dhcp-no-override** の設定を有効にします。Satellite は **grub2/** サブディレクトリーの配下にある TFTP サーバーに設定ファイルを作成するので、この設定を必ず有効にしてください。**dhcp-no-override** 設定が無効になっている場合、ホストがブートローダーとその設定をルートディレクトリーから取得するため、エラーが発生する可能性があります。

手順

1. Red Hat Enterprise Linux ホストに、ISC DHCP Service と Berkeley Internet Name Domain (BIND) ユーティリティーパッケージをインストールします。

```
# dnf install dhcp-server bind-utils
```

2. セキュリティートークンを生成します。

```
# dnssec-keygen -a HMAC-MD5 -b 512 -n HOST omapi_key
```

上記のコマンドを実行すると、2つのファイルで設定されるキーペアが現在のディレクトリーに作成されます。

3. キーからシークレットハッシュをコピーします。

```
# grep ^Key Komapi_key.+.private | cut -d ' ' -f2
```

4. すべてのサブネットの **dhcpd** 設定ファイルを編集し、キーを追加します。以下に例を示します。

```
# cat /etc/dhcp/dhcpd.conf
default-lease-time 604800;
max-lease-time 2592000;
log-facility local7;

subnet 192.168.38.0 netmask 255.255.255.0 {
  range 192.168.38.10 192.168.38.100;
```

```
option routers 192.168.38.1;  
option subnet-mask 255.255.255.0;  
option domain-search "virtual.lan";  
option domain-name "virtual.lan";  
option domain-name-servers 8.8.8.8;  
}
```

```
omapi-port 7911;  
key omapi_key {  
  algorithm HMAC-MD5;  
  secret "My_Secret";  
};  
omapi-key omapi_key;
```

option routers の値は、外部 DHCP サービスで使用する Satellite Server または Capsule Server の IP アドレスであることに注意してください。

5. キーファイルが作成されたディレクトリーから、2つのキーファイルを削除します。
6. Satellite Server で各サブネットを定義します。定義済みのサブネットに DHCP Capsule は設定しないでください。
競合を回避するには、リースと予約範囲を別に設定します。たとえば、リース範囲を 192.168.38.10 から 192.168.38.100 に設定した場合には、Satellite Web UI で予約範囲を 192.168.38.101 から 192.168.38.250 に設定します。
7. DHCP サーバーに外部アクセスできるように、ファイアウォールを設定します。

```
# firewall-cmd --add-service dhcp
```

8. 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

9. Satellite Server で **foreman** ユーザーの UID と GID を指定します。

```
# id -u foreman  
993  
# id -g foreman  
990
```

10. DHCP サーバーで、1つ前の手順で定義した ID と同じ **foreman** ユーザーとグループを作成します。

```
# groupadd -g 990 foreman  
# useradd -u 993 -g 990 -s /sbin/nologin foreman
```

11. 設定ファイルにアクセスできるように、読み取りおよび実行フラグを復元します。

```
# chmod o+rx /etc/dhcp/  
# chmod o+r /etc/dhcp/dhcpd.conf  
# chattr +i /etc/dhcp/ /etc/dhcp/dhcpd.conf
```

12. DHCP サービスを有効にして開始します。


```
# systemctl enable --now dhcpd
```

13. NFS を使用して DHCP 設定ファイルおよびリースファイルをエクスポートします。

```
# dnf install nfs-utils  
# systemctl enable --now nfs-server
```

14. NFS を使用してエクスポートする DHCP 設定ファイルとリースファイルのディレクトリーを作成します。

```
# mkdir -p /exports/var/lib/dhcpd /exports/etc/dhcp
```

15. 作成したディレクトリーにマウントポイントを作成するには、以下の行を **/etc/fstab** ファイルに追加します。

```
/var/lib/dhcpd /exports/var/lib/dhcpd none bind,auto 0 0  
/etc/dhcp /exports/etc/dhcp none bind,auto 0 0
```

16. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

17. **/etc/exports** に以下の行があることを確認します。

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)  
/exports/etc/dhcp 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)  
/exports/var/lib/dhcpd 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

入力する IP アドレスは、外部 DHCP サービスで使用する Satellite または Capsule IP アドレスを指定する点に注意してください。

18. NFS サーバーをリロードします。

```
# exportfs -rva
```

19. ファイアウォールで DHCP omapi ポート 7911 を設定します。

```
# firewall-cmd --add-port=7911/tcp
```

20. オプション: NFS に外部からアクセスできるようにファイアウォールを設定します。クライアントは NFSv3 を使用して設定します。

```
# firewall-cmd \  
--add-service mountd \  
--add-service nfs \  
--add-service rpc-bind \  
--zone public
```

21. 変更を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

4.2.2. 外部 DHCP サーバーを使用した Satellite Server の設定

外部 DHCP サーバーを使用した Satellite Server を設定できます。

前提条件

- 外部の DHCP サーバーを設定し、Satellite Server と DHCP 設定ファイルとリースファイルを共有していることを確認する。詳細は、「[Satellite Server で使用する外部 DHCP サーバーの設定](#)」を参照してください。

手順

1. **nfs-utils** パッケージをインストールします。

```
# satellite-maintain packages install nfs-utils
```

2. NFS 用の DHCP ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

3. ファイルの所有者を変更します。

```
# chown -R foreman-proxy /mnt/nfs
```

4. NFS サーバーとの通信とリモートプロシージャコール (RPC: Remote Procedure Call) 通信パスを検証します。

```
# showmount -e DHCP_Server_FQDN  
# rpcinfo -p DHCP_Server_FQDN
```

5. **/etc/fstab** ファイルに以下の行を追加します。

```
DHCP_Server_FQDN:/exports/etc/dhcp /mnt/nfs/etc/dhcp nfs  
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcp_etc_t:s0" 0 0  
  
DHCP_Server_FQDN:/exports/var/lib/dhcpd /mnt/nfs/var/lib/dhcpd nfs  
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcpd_state_t:s0" 0 0
```

6. **/etc/fstab** でファイルシステムをマウントします。

```
# mount -a
```

7. **foreman-proxy** ユーザーがネットワークで共有したファイルにアクセスできることを確認するには、DHCP 設定ファイルとリースファイルを表示します。

```
# su foreman-proxy -s /bin/bash  
$ cat /mnt/nfs/etc/dhcp/dhcpd.conf  
$ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases  
$ exit
```

8. **satellite-installer** コマンドを入力して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/dhcp.yml** ファイルに加えます。

```
# satellite-installer \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-dhcp-server=My_DHCP_Server_FQDN \
--foreman-proxy-dhcp=true \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-secret=My_Secret \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911
```

9. DHCP サービスに適切なサブネットとドメインを関連付けます。

4.3. 外部 TFTP を使用した SATELLITE SERVER の設定

外部 TFTP サービスを使用して Satellite Server を設定できます。

手順

1. NFS 用に TFTP ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/var/lib/tftpboot
```

2. **/etc/fstab** ファイルで以下の行を追加します。

```
TFTP_Server_IP_Address:/exports/var/lib/tftpboot /mnt/nfs/var/lib/tftpboot nfs
rw,vers=3,auto,nosharecache,context="system_u:object_r:tftpd_dir_rw_t:s0" 0 0
```

3. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

4. **satellite-installer** コマンドを入力して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/tftp.yml** ファイルに加えます。

```
# satellite-installer \
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot \
--foreman-proxy-tftp=true
```

5. DHCP サービスとは異なるサーバーで TFTP サービスを実行している場合は、TFTP サービスを実行するサーバーの FQDN または IP アドレスに、**tftp_servername** 設定を更新します。

```
# satellite-installer --foreman-proxy-tftp-servername=TFTP_Server_FQDN
```

6. Satellite Web UI で、**Infrastructure > Capsules** に移動します。
7. Satellite Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。
8. TFTP サービスに適切なサブネットとドメインを関連付けます。

4.4. 外部 IDM DNS を使用した SATELLITE SERVER の設定

Satellite Server がホストの DNS レコードを追加する時には、まずどの Capsule が対象のドメインに DNS を提供しているかを判断します。次に、デプロイメントに使用する DNS サービスを提供するように設定された Capsule と通信し、レコードを追加します。ホストはこのプロセスには関与しません。そのため、IdM サーバーを使用して管理するドメインに DNS サービスを提供するように設定された Satellite または Capsule に IdM クライアントをインストールし、設定する必要があります。

Satellite Server は、Red Hat Identity Management (IdM) サーバーを使用して DNS サービスを提供するように設定できます。Red Hat Identity Management の詳細は、[Linux Domain Identity, Authentication, and Policy Guide](#) を参照してください。

Red Hat Identity Management (IdM) サーバーを使用して DNS サービスを提供するように Satellite Server を設定するには、以下の手順のいずれかを使用します。

- [「GSS-TSIG 認証を使用した動的 DNS 更新の設定」](#)
- [「TSIG 認証を使用した動的 DNS 更新の設定」](#)

内部 DNS サービスに戻すには、次の手順を使用します。

- [「内部 DNS サービスに戻す」](#)



注記

DNS の管理に、Satellite Server を使用する必要はありません。Satellite のレルム登録機能を使用しており、プロビジョニングされたホストが自動的に IdM に登録されている場合は、**ipa-client-install** スクリプトでクライアント用に DNS レコードが作成されます。外部の IdM DNS とレルム登録を同時に使用して、Satellite Server を設定することはできません。レルム登録の設定に関する詳細は、[オンラインネットワーク環境への Satellite Server のインストール](#) の [プロビジョニングされたホストの外部認証](#) を参照してください。

4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

[RFC3645](#) で定義されている秘密鍵トランザクション (GSS-TSIG) 技術の一般的なセキュリティーサービスアルゴリズムを使用するように IdM サーバーを設定できます。IdM サーバーが GSS-TSIG 技術を使用するように設定するには、Satellite Server のベースオペレーティングシステムに IdM クライアントをインストールする必要があります。

前提条件

- IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されている。詳細は、[Identity Management のインストールガイド](#) の [IdM のポート要件](#) を参照してください。
- IdM サーバーの管理者に問い合わせて、IdM サーバーでゾーンを作成するパーミッションが割り当てられた、IdM サーバーのアカウントを取得する。
- 応答ファイルのバックアップを作成する必要があります。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、[Satellite Server の設定](#) を参照してください。

手順

GSS-TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

IdM サーバーでの Kerberos プリンシパルの作成

1. IdM 管理者から取得したアカウントの Kerberos チケットを取得します。

```
# kinit idm_user
```

2. IdM サーバーでの認証に使用する Satellite Server の新しい Kerberos プリンシパルを作成します。

```
# ipa service-add capsule/satellite.example.com
```

IdM クライアントのインストールおよび設定

1. デプロイメントの DNS サービスを管理する Satellite または Capsule のベースオペレーティングシステムで **ipa-client** パッケージをインストールします。

```
# satellite-maintain packages install ipa-client
```

2. インストールスクリプトとそれに続くプロンプトを実行して、IdM クライアントを設定します。

```
# ipa-client-install
```

3. Kerberos チケットを取得します。

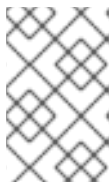
```
# kinit admin
```

4. 既存の **keytab** を削除します。

```
# rm /etc/foreman-proxy/dns.keytab
```

5. このシステムの **keytab** を取得します。

```
# ipa-getkeytab -p capsule/satellite.example.com@EXAMPLE.COM \  
-s idm1.example.com -k /etc/foreman-proxy/dns.keytab
```



注記

サービス中の元のシステムと同じホスト名を持つスタンバイシステムに keytab を追加する際には、**r** オプションを追加します。これにより、新規の認証情報が生成されることを防ぎ、元のシステムの認証情報が無効になります。

6. **dns.keytab** ファイルのグループと所有者を **foreman-proxy** に設定します。

```
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/dns.keytab
```

7. オプション: **keytab** ファイルが有効であることを確認するには、以下のコマンドを入力します。

```
# kinit -kt /etc/foreman-proxy/dns.keytab \  
capsule/satellite.example.com@EXAMPLE.COM
```

IdM Web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。
 - a. **Network Services** > **DNS** > **DNS Zones** に移動します。
 - b. **Add** を選択し、ゾーン名を入力します。(例: **example.com**)
 - c. **Add and Edit** をクリックします。
 - d. 設定タブをクリックして **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

```
grant capsule\047satellite.example.com@EXAMPLE.COM wildcard * ANY;
```

- e. **Dynamic update** を **True** に設定します。
 - f. **Allow PTR sync** を有効にします。
 - g. **Save** をクリックして、変更を保存します。
2. 逆引きゾーンを作成して設定します。
 - a. **Network Services** > **DNS** > **DNS Zones** に移動します。
 - b. **Add** をクリックします。
 - c. **Reverse zone IP network** を選択して、CIDR 形式でネットワークアドレスを追加し、逆引き参照を有効にします。
 - d. **Add and Edit** をクリックします。
 - e. **Settings** タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

```
grant capsule\047satellite.example.com@EXAMPLE.COM wildcard * ANY;
```

- f. **Dynamic update** を **True** に設定します。
 - g. **Save** をクリックして、変更を保存します。

ドメインの DNS サービスを管理する Satellite または Capsule Server の設定

1. **satellite-installer** コマンドを使用して、ドメインの DNS サービスを管理するように Satellite または Capsule を設定します。
 - Satellite で以下のコマンドを入力します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns=true
```

- Capsule で、以下のコマンドを実行します。

```
# satellite-installer --scenario capsule \  
--foreman-proxy-dns-managed=false \  
--foreman-proxy-dns-provider=nsupdate_gss \  
--foreman-proxy-dns-server="idm1.example.com" \  
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \  
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \  
--foreman-proxy-dns=true
```

satellite-installer コマンドを実行して Capsule 設定に変更を加えた後に、Satellite Web UI で変更のある Capsule ごとに設定を更新する必要があります。

Satellite Web UI での設定更新

1. Satellite Web UI で、**Infrastructure** > **Capsules** に移動し、Satellite Server を見つけて、**Actions** 列のリストから **Refresh** を選択します。
2. ドメインを設定します。
 - a. Satellite Web UI で、**Infrastructure** > **Domains** に移動し、ドメイン名を選択します。
 - b. **Domain** タブで、**DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
3. サブネットを設定します。
 - a. Satellite Web UI で、**Infrastructure** > **Subnets** に移動し、サブネット名を選択します。
 - b. **Subnet** タブで、**IPAM** を **None** に設定します。
 - c. **Domains** タブで、IdM サーバーを使用して管理するドメインを選択します。
 - d. **Capsules** タブで、**Reverse DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
 - e. **Submit** をクリックして変更を保存します。

4.4.2. TSIG 認証を使用した動的 DNS 更新の設定

IdM サーバーが DNS (TSIG) テクノロジーの秘密鍵トランザクション認証を使用するように設定できます。このテクノロジーは、認証に **rndc.key** キーファイルを使用します。TSIG プロトコルについては [RFC2845](#) に定義されています。

前提条件

- IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されている。詳細は [Linux Domain Identity, Authentication, and Policy Guide](#) の [Port Requirements](#) を参照してください。
- IdM サーバーで **root** 権限を取得する必要があります。
- デプロイメントに DNS サービスを提供するように Satellite Server または Capsule Server が設定されていることを確認する。

- デプロイメントの DNS サービスを管理する Satellite または Capsule のいずれかのベースオペレーティングシステムで DNS、DHCP および TFTP サービスを設定する必要があります。
- 応答ファイルのバックアップを作成しておく。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、[Satellite Server の設定](#) を参照してください。

手順

TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

IdM サーバーの DNS ゾーンに対する外部更新の有効化

1. IdM サーバーで、以下の内容を `/etc/named.conf` ファイルの先頭に追加します。

```
#####
include "/etc/rndc.key";
controls {
inet _IdM_Server_IP_Address_ port 953 allow { _Satellite_IP_Address_; } keys { "rndc-key";
};
};
#####
```

2. **named** サービスをリロードして、変更を有効にします。

```
# systemctl reload named
```

3. IdM Web UI で、**Network Services > DNS > DNS Zones** に移動して、ゾーンの名前をクリックします。**Settings** タブで、以下の変更を適用します。

- a. **BIND update policy** ボックスで以下の内容を追加します。

```
grant "rndc-key" zonesub ANY;
```

- b. **Dynamic update** を **True** に設定します。

- c. **Update** をクリックして変更を保存します。

4. IdM サーバーから Satellite Server のベースオペレーティングシステムに `/etc/rndc.key` ファイルをコピーします。以下のコマンドを入力します。

```
# scp /etc/rndc.key root@satellite.example.com:/etc/rndc.key
```

5. **rndc.key** ファイルに適切な所有者、パーミッション、SELinux コンテキストを設定するには、以下のコマンドを入力します。

```
# restorecon -v /etc/rndc.key
# chown -v root:named /etc/rndc.key
# chmod -v 640 /etc/rndc.key
```

6. **foreman-proxy** ユーザーは、手動で **named** グループに割り当てます。通常、`satellite-installer` は **foreman-proxy** ユーザーが **named** UNIX グループに所属させますが、今回のシナリオでは、Satellite でユーザーとグループを管理していないので、**foreman-proxy** ユーザーを **named** グループに手作業で割り当てる必要があります。


```
# usermod -a -G named foreman-proxy
```

- Satellite Server で以下の **satellite-installer** コマンドを入力して、Satellite が外部の DNS サーバーを使用するように設定します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="IdM_Server_IP_Address" \
--foreman-proxy-dns-ttl=86400 \
--foreman-proxy-dns=true \
--foreman-proxy-keyfile=/etc/rndc.key
```

IdM サーバーの DNS ゾーンに対する外部更新のテスト

- Satellite Server 上の **/etc/rndc.key** ファイルのキーが IdM サーバーで使用されているキーファイルと同じであることを確認します。

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "secret-key==";
};
```

- Satellite Server で、ホストのテスト DNS エントリーを作成します。(例: **192.168.25.1** の IdM サーバーに、**192.168.25.20** の A レコードを指定した **test.example.com** ホストなど)

```
# echo -e "server 192.168.25.1\n \
update add test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

- Satellite Server で、DNS エントリーをテストします。

```
# nslookup test.example.com 192.168.25.1
Server: 192.168.25.1
Address: 192.168.25.1#53

Name: test.example.com
Address: 192.168.25.20
```

- IdM Web UI でエントリーを参照するために、**Network Services > DNS > DNS Zones** に移動します。ゾーンの名前をクリックし、名前でホストを検索します。
- 正常に解決されたら、テスト DNS エントリーを削除します。

```
# echo -e "server 192.168.25.1\n \
update delete test.example.com 3600 IN A 192.168.25.20\n \
send\n" | nsupdate -k /etc/rndc.key
```

- DNS エントリーが削除されたことを確認します。

```
# nslookup test.example.com 192.168.25.1
```

レコードが正常に削除されている場合は、上記の **nslookup** コマンドが失敗し、**SERVFAIL** エラーメッセージを返します。

4.4.3. 内部 DNS サービスに戻す

Satellite Server および Capsule Server を DNS プロバイダーとして使用するように戻すことができます。外部の DNS を設定する前に作成した応答ファイルのバックアップを使用するか、応答ファイルのバックアップを作成します。応答ファイルの詳細は、[Satellite Server の設定](#) を参照してください。

手順

ドメインの DNS サーバーを管理するように設定する Satellite または Capsule Server で、以下の手順を実行します。

Satellite または Capsule を DNS サーバーとして設定する

- 外部の DNS を設定する前に応答ファイルのバックアップを作成済みの場合には、応答ファイルを復元して、**satellite-installer** コマンドを入力します。

```
# satellite-installer
```

- 応答ファイルの適切なバックアップがない場合には、ここで応答ファイルのバックアップを作成します。応答ファイルを使用せずに Satellite または Capsule を DNS サーバーとして設定するには、Satellite と Capsule で、以下の **satellite-installer** コマンドを入力します。

```
# satellite-installer \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1" \
--foreman-proxy-dns=true
```

詳細は、[Capsule Server での DNS、DHCP、および TFTP の設定](#) を参照してください。

satellite-installer コマンドを実行して Capsule 設定に変更を加えた後に、Satellite Web UI で変更のある Capsule ごとに設定を更新する必要があります。

Satellite Web UI での設定更新

1. Satellite Web UI で、**Infrastructure > Capsules** に移動します。
2. 更新する各 Capsule で、**Actions** リストから **Refresh** を選択します。
3. ドメインを設定します。
 - a. Satellite Web UI で、**Infrastructure > Domains** に移動し、設定するドメイン名をクリックします。
 - b. **Domain** タブで、**DNS Capsule** を、サブネットの接続先の Capsule に設定します。
4. サブネットを設定します。
 - a. Satellite Web UI で、**Infrastructure > Subnets** に移動し、サブネット名を選択します。
 - b. **Subnet** タブで、**IPAM** を **DHCP** または **Internal DB** に設定します。
 - c. **Domains** タブで、Satellite または Capsule で管理するドメインを選択します。

- d. **Capsules** タブで、**Reverse DNS Capsule** を、サブネットの接続先の Capsule に設定します。
- e. **Submit** をクリックして変更を保存します。

付録A RED HAT SATELLITE へのカスタム設定の適用

satellite-installer を使用して初めて Satellite をインストールし、設定する場合には、**--foreman-proxy-dns-managed=false** と **--foreman-proxy-dhcp-managed=false** のインストーラーフラグを使用して、DNS および DHCP 設定ファイルが Puppet で管理されないように指定してください。これらのフラグがインストーラーの初回実行時に指定されていない場合には、アップグレードの目的で再実行する場合など、インストーラーを再実行すると、手動で変更した内容がすべて上書きされます。変更が上書きされた場合には、復元の手順を実行して手動の変更を復元する必要があります。詳細は、[Puppet 実行で上書きされた手動変更の復元](#) を参照してください。

カスタム設定に利用可能なすべてのインストーラーフラグを表示するには、**satellite-installer --scenario satellite --full-help** を実行します。Puppet クラスには、Satellite インストーラーに公開されていないものもあります。これらのクラスを手動で管理して、インストーラーが値を上書きしないようにするには、設定ファイル **/etc/foreman-installer/custom-hiera.yaml** にエントリーを追加して設定値を指定します。この設定ファイルは YAML 形式で、**<puppet class>::<parameter name>: <value>** という形式を 1 行あたり 1 エントリーで記入します。このファイルで指定した設定値は、インストーラーを再起動しても維持されます。

一般的な例を示します。

- Apache で ServerTokens ディレクティブが製品名のみを返すように設定するには、以下のようになります。

```
apache::server_tokens: Prod
```

- Apache サーバー署名をオフにするには、以下のようになります。

```
apache::server_signature: Off
```

Satellite インストーラー用の Puppet モジュールは、**/usr/share/foreman-installer/modules** に保存されています。クラス、パラメーター、および値を調べるには、**.pp** ファイル (例: **moduleName/manifests/example.pp**) を確認してください。別の方法では、**grep** コマンドでキーワード検索を実行します。

値の設定によっては、Red Hat Satellite のパフォーマンスや機能に影響が出る意図しない結果がもたらされる場合があります。設定を適用する前に変更の影響を考慮して、実稼働以外の環境で最初に変更をテストしてください。実稼働以外の Satellite 環境がない場合は、Satellite インストーラーを **--noop** と **--verbose** のオプションを追加して実行します。変更によって問題が発生する場合は、該当箇所を **custom-hiera.yaml** から削除し、Satellite インストーラーを再実行します。特定の値を変更することが安全かどうかを確認する場合は、Red Hat サポートにお問い合わせください。

付録B PUPPET 実行で上書きされた手動変更の復元

Puppet 実行で手動による設定が上書きされた場合でも、ファイルを元の状態に戻すことができます。以下の例では、Puppet 実行で上書きされた DHCP 設定ファイルを復元します。

手順

1. 復元するファイルをコピーします。こうすることで、アップグレードに必要な変更があるか、ファイル間で比較できます。これは DNS や DHCP サービスでは一般的ではありません。

```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.backup
```

2. ログファイルを確認して、上書きされたファイルの md5sum をメモします。以下に例を示します。

```
# journalctl -xe
...
/Stage[main]/Dhcp/File[/etc/dhcp/dhcpd.conf]: Filebucketed /etc/dhcp/dhcpd.conf to puppet
with sum 622d9820b8e764ab124367c68f5fa3a1
...
```

3. 上書きされたファイルを復元します。

```
# puppet filebucket restore --local --bucket \
/var/lib/puppet/clientbucket /etc/dhcp/dhcpd.conf \ 622d9820b8e764ab124367c68f5fa3a1
```

4. バックアップしたファイルと復元されたファイルを比べます。復元されたファイルに、アップグレードに必要な変更を追加します。

付録C RED HAT CDN からコンテンツをダウンロードするように SATELLITE SERVER を戻す

お使いの環境がオフラインからオンラインに変更された場合は、Red Hat CDN から直接コンテンツをダウンロードするようにオフラインの Satellite Server を再設定することができます。

手順

1. Satellite Web UI で、**コンテンツ > サブスクリプション** に移動します。
2. **マニフェストの管理** をクリックします。
3. **CDN Configuration** タブに切り替えます。
4. **Red Hat** を選択します。
5. **URL** フィールドを編集して、Red Hat CDN URL をポイントします。
<https://cdn.redhat.com>
6. **Update** をクリックします。

これで Satellite Server は、次にリポジトリを同期するときに Red Hat CDN からコンテンツをダウンロードするように設定されました。

CLI 手順

1. SSH を使用して Satellite Server にログインします。
2. Hammer を使用して CDN を再設定します。

```
# hammer organization configure-cdn --name="My_Organization" --type=redhat_cdn
```