Red Hat Satellite 6.15

オン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のあるネットワークに Satellite Server をインストールして設定する

Last Updated: 2024-04-26

Red Hat Satellite 6.15 オン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のあるネットワークに 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

概要

本書では、オンラインネットワークから 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

多様性を受け入れるオープンソースの強化 4

RED HAT ドキュメントへのフィードバック (英語のみ) 5

第1章 インストールのための環境準備 6
1.1. システム要件 6
1.2. ストレージ要件 7
1.3. ストレージのガイドライン 8
1.4. サポート対象オペレーティングシステム 9
1.5. サポート対象のブラウザー 9
1.6. ポートとファイアウォールの要件 10
1.7. クライアントから SATELLITE SERVER への接続の有効化 15
1.8. DNS 解決の検証 16
1.9. 事前定義済みプロファイルを使用した SATELLITE SERVER の調整 16

第2章 IPV6 ネットワークでの SATELLITE インストール環境の準備 19
2.1. IPV6 ネットワークでの SATELLITE インストールの制限事項 19
2.2. IPV6 ネットワークでの SATELLITE インストールの要件 19

第3章 SATELLITE SERVER のインストール 20
3.1. RED HAT CDN に接続するための HTTP プロキシーの設定 20
3.2. RED HAT サブスクリプション管理への登録 21
3.3. SATELLITE INFRASTRUCTURE サブスクリプションの割り当て 22
3.4. リポジトリーの設定 23
3.5. オプション: SATELLITE SERVER での FAPOLICYD の使用 24
3.6. SATELLITE SERVER パッケージのインストール 25
3.7. CHRONYD によるシステムクロックの同期 25
3.8. 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 25
3.9. SATELLITE SERVER の設定 25
3.10. 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 26

第4章 SATELLITE SERVER での追加設定の実行 28
4.1. SATELLITE SERVER での RED HAT INSIGHTS の使用 28
4.2. RED HAT INSIGHTS の登録の無効化 28
4.3. SATELLITE CLIENT 6 リポジトリーの有効化と同期 29
4.4. SATELLITE SERVER でプルクライアントのリモート実行を設定する 33
4.5. IPV6 ネットワークでの UEFI HTTP ブート向けの SATELLITE の設定 34
4.6. HTTP プロキシーを使用した SATELLITE SERVER の設定 34
4.7. ホストでの電源管理の有効化 37
4.8. DNS、DHCP、および TFTP の設定 37
4.9. SATELLITE SERVER での送信メールの設定 39
4.10. SATELLITE 向けの別の CNAME の設定 41
4.11. カスタムの SSL 証明書を使用した SATELLITE SERVER の設定 42
4.12. SATELLITE での外部データベースの使用 46

第5章 外部認証の設定 51
5.1. LDAP の使用 52
5.2. RED HAT IDENTITY MANAGEMENT の使用 57
5.3. ACTIVE DIRECTORY の使用 60
5.4. 外部ユーザーグループの設定 65
5.5. LDAP の外部ユーザーグループのリフレッシュ 65
5.6. RED HAT IDENTITY MANAGEMENT または AD の外部ユーザーグループの更新 66
5.7. RED HAT IDENTITY MANAGEMENT ユーザー認証を使用するための HAMMER CLI の設定 66

5.8. プロビジョニングされたホストの外部認証 67
5.9. RED HAT SINGLE SIGN-ON 認証を使用した SATELLITE の設定 70
5.10. TOTP を使用した RED HAT SINGLE SIGN ON 認証の設定 76
5.11. RED HAT SINGLE SIGN-ON 認証の無効化 83

第6章 外部サービスを使用した SATELLITE SERVER の設定 84
6.1. 外部 DNS を使用した SATELLITE SERVER の設定 84
6.2. 外部 DHCP を使用した SATELLITE SERVER の設定 85
6.3. 外部 TFTP を使用した SATELLITE SERVER の設定 89
6.4. 外部 IDM DNS を使用した SATELLITE SERVER の設定 90

付録A DNF モジュールのトラブルシューティング 98
A.1. RUBY 98
A.2. POSTGRESQL 98

付録B RED HAT SATELLITE へのカスタム設定の適用 99

付録C PUPPET 実行で上書きされた手動変更の復元 100

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。この取り組みは膨大な作業を要するため、これらの変更による更新は可能な範囲で段階的に行われます。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

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 リポジトリーが有効化され、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ー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、**セキュリティー強化** 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、今後に焦点を合わせたより厳格なセキュリティーレベルです。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、想定の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ー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ーに追加するときに必要な追加ストレージが少なくて済みます。ストレージの多くは、**/var/lib/pulp/** ディレクトリーにあります。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。

同期された RHEL ISO

RHEL コンテンツの ISO を Satellite に同期する予定の場合には、Red Hat Enterprise Linux のすべてのマイナーバージョンも同期することに注意してください。これに対応するため、Satellite に適切なストレージを設定するようにプランニングする必要があります。

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ー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

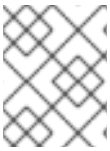
送信先ポート	プロトコル	サービス	ソース	用途	説明
53	TCP および UDP	DNS	DSN サーバーおよびクライアント	名前解決	DNS (オプション)
67	UDP	DHCP	クライアント	動的 IP	DHCP (オプション)
69	UDP	TFTP	クライアント	TFTP サーバー (オプション)	
443	TCP	HTTPS	Capsule	Red Hat Satellite API	Capsule からの通信

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	スマートプロキシー機能
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	スマートプロキシー機能
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ー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 (コンピュートリソース)	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。**LDAPAuthSource** 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 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	Capsule
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	console.redhat.com	Red Hat Cloud プラグイン API 呼び出し	

送信先ポート	プロトコル	サービス	宛先	用途	説明
443	TCP	HTTPS	cdn.redhat.com	コンテンツ同期	Red Hat CDN
443	TCP	HTTPS	api.access.redhat.com	SOS レポート	Red Hat カスタマーポータル を通じて提出されたサポートケースの支援 (オプション)
443	TCP	HTTPS	cert-api.access.redhat.com	Telemetry データのアップロードとレポート	
443	TCP	HTTPS	Capsule	コンテンツのミラーリング	開始
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

送信先ポート	プロトコル	サービス	宛先	用途	説明
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** ディレクトリーにあります。

--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/** ディレクトリーで定義され、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。

3. 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 IPV6 ネットワークでの SATELLITE インストール環境の準備

IPv6 ネットワークで Satellite をインストールして使用できます。IPv6 ネットワークで Satellite をインストールする前に、制限事項と、以下の要件を満たしていることを確認してください。

IPv6 ネットワークにホストをプロビジョニングするには、Satellite のインストール後に、UEFI HTTP ブートプロビジョニング用の Satellite も設定する必要があります。詳細は、「IPv6 ネットワークでの UEFI HTTP ブート向けの Satellite の設定」 を参照してください。

2.1. IPV6 ネットワークでの SATELLITE インストールの制限事項

IPv6 ネットワークでの Satellite のインストールには、次の制限があります。

- Satellite および Capsule は、IPv6 のみのシステムにインストールでき、デュアルスタックのインストールはサポートしていません。
- Satellite プロビジョニングテンプレートには、PXE と HTTP (iPXE) プロビジョニングでの IPv6 サポートがありますが、テスト済みかつ認定済みのプロビジョニングワークフローは UEFI HTTP ブートプロビジョニングです。この制約は、Satellite を使用してホストをプロビジョニングする場合にのみ適用されます。

2.2. IPV6 ネットワークでの SATELLITE インストールの要件

IPv6 ネットワークで Satellite をインストールする前に、以下の要件を満たしていることを確認してください。

- 外部の IPv6 サーバーをマネージド外のサービスとして別にデプロイして GURB2 にクライアントをブートストラップしてから、DHCPv6 を使用するか、IPv6 アドレスを割り当てて IPv 6 ネットワークを設定する必要があります。これが必要なのは、Red Hat Enterprise Linux の DHCP サーバー (ISC DHCP) に IPv6 レコードを管理するための統合 API がなく、DHCP 管理を提供する Capsule DHCP プラグインが IPv4 サブネットに制限されるためです。
- IPv4 と IPv6 の両方をサポートする外部 HTTP プロキシーサーバーをデプロイする必要があります。これは、Red Hat Content Delivery Network が IPv4 ネットワーク上でのみコンテンツを配信するため、IPv6 ネットワーク上の Satellite にコンテンツを取り込むにはこのプロキシーを使用する必要があるためです。
- このデュアルスタック (IPv4 と IPv6 の両方の) HTTP プロキシーをデフォルトのプロキシーとして使用するように Satellite を設定する必要があります。詳細は、デフォルトの HTTP プロキシーの Satellite への追加 を参照してください。

第3章 SATELLITE SERVER のインストール

オンラインネットワークから Satellite Server をインストールする場合は、Red Hat コンテンツ配信ネットワークから直接パッケージと更新を取得できます。

注記

Satellite Server に自己登録することはできません。

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。サブスクリプション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。

Satellite インストールスクリプトは Puppet をベースとするので、インストールスクリプトを複数回実行すると、手動での設定変更を上書きする可能性がある点に注意してください。これを回避し、今後どの変更を適用するか判断するには、インストールスクリプトの実行時に **--noop** の引数を使用します。この引数では、実際の変更は加えられません。今後変更される可能性のある内容は **/var/log/foreman-installer/satellite.log** に書き込まれます。

ファイルは常にバックアップされるため、不要な変更を元に戻すことができます。たとえば、foreman-installer ログで Filebucket に関する以下のようなエントリーが確認できます。

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

以前のファイルは以下のように復元できます。

```
# puppet filebucket -l \
restore /etc/dhcp/dhcpd.conf 622d9820b8e764ab124367c68f5fa3a1
```

3.1. RED HAT CDN に接続するための HTTP プロキシーの設定

前提条件

ネットワークゲートウェイと HTTP プロキシーは、次のホストへのアクセスを許可する必要があります。

ホスト名	ポート	プロトコル
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS
cert.console.redhat.com (Red Hat Insights を使用している場合)	443	HTTPS
api.access.redhat.com (Red Hat Insights を使用している場合)	443	HTTPS

ホスト名	ポート	プロトコル
cert-api.access.redhat.com (Red Hat Insights を使用している場合)	443	HTTPS

Satellite Server は、SSL を使用して Red Hat CDN との通信のセキュリティーを確保します。この通信は、SSL インターセプトプロキシーによって妨害されます。上記のホストは、HTTP プロキシーで許可リストに登録されている必要があります。

Red Hat CDN (cdn.redhat.com) で使用されている IP アドレスのリストは、Red Hat カスタマーポータルのナレッジベース記事 Red Hat が公開している CIDR のリスト を参照してください。

HTTP プロキシーを使用して Subscription Manager を設定するには、以下の手順に従います。

手順

1. Satellite Server の **/etc/rhsm/rhsm.conf** ファイルで、以下の詳細を記入します。

```
# an http proxy server to use (enter server FQDN)
proxy_hostname = myproxy.example.com

# port for http proxy server
proxy_port = 8080

# user name for authenticating to an http proxy, if needed
proxy_user =

# password for basic http proxy auth, if needed
proxy_password =
```

3.2. RED HAT サブスクリプション管理への登録

Red Hat サブスクリプション管理にホストを登録すると、ユーザーが利用可能なサブスクリプションにホストを登録して、サブスクリプションのコンテンツを使用できるようになります。これには、Red Hat Enterprise Linux、Red Hat Satellite などのコンテンツが含まれます。

手順

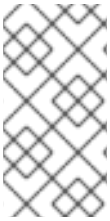
- Red Hat コンテンツ配信ネットワークにシステムを登録します。プロンプトが表示されたら、カスタマーポータルのユーザー名とパスワードを入力します。

```
# subscription-manager register
```

このコマンドを実行すると、以下のような出力が表示されます。

```
# subscription-manager register
Username: user_name
Password:
The system has been registered with ID: 541084ff2-44cab-4eb1-9fa1-7683431bcf9a
```

3.3. SATELLITE INFRASTRUCTURE サブスクリプションの割り当て



注記

Red Hat カスタマーポータルで SCA を有効にしている場合は、この手順をスキップしてください。subscription-manager を使用して Red Hat Satellite Infrastructure Subscription サブスクリプションを Satellite Server にアタッチする必要はありません。SCA の詳細は、Simple Content Access を参照してください。

Satellite Server の登録後に、サブスクリプションプール ID を特定して、利用可能なサブスクリプションを割り当てる必要があります。Red Hat Satellite Infrastructure サブスクリプションは、Red Hat Satellite および Red Hat Enterprise Linux コンテンツにアクセスできるようになります。

Red Hat Satellite Infrastructure は、Satellite (以前は Smart Management と呼ばれていました) を提供するすべてのサブスクリプションに含まれています。詳細は、**Red Hat ナレッジベース** の Satellite Infrastructure サブスクリプション MCT3718 MCT3719 を参照してください。

サブスクリプションがシステムに割り当てられていない場合には、利用可能として分類されます。利用可能な Satellite サブスクリプションが見つからない場合は、Red Hat ナレッジベースソリューション Red Hat Subscription Manager に登録されているクライアントが使用したサブスクリプションを把握するにはどうすればよいですか? を参照してスクリプトを実行し、サブスクリプションが別のシステムで使用されているかどうかを確認します。

手順

1. Satellite Infrastructure サブスクリプションのプール ID を特定します。

```
# subscription-manager list --all --available --matches 'Red Hat Satellite Infrastructure Subscription'
```

このコマンドを実行すると、以下のような出力が表示されます。

```
Subscription Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Satellite
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Satellite Capsule
                     Red Hat Ansible Engine
                     Red Hat Satellite with Embedded Oracle
                     Red Hat Satellite 5 Managed DB
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat Beta
                     Red Hat Software Collections Beta (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Satellite Proxy
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Discovery
SKU:                 MCT3718
Contract:
Pool ID:             8aca43dd771bf31101771c0231f906a5
Provides Management: Yes
Available:           10
Suggested:           1
```

```
Service Type:       L1-L3
Roles:
Service Level:      Premium
Usage:
Add-ons:
Subscription Type:  Standard
Starts:             11/11/2020
Ends:               11/11/2023
Entitlement Type:   Physical
```

2. サブスクリプションプール ID を書き留めます。上記の例と、実際のサブスクリプションプール ID は異なります。

3. Satellite Server が実行されているベースオペレーティングシステムに、Satellite Infrastructure サブスクリプションを割り当てます。SCA が Satellite Server で有効になっている場合は、この手順をスキップできます。

```
# subscription-manager attach --pool=pool_id
```

このコマンドを実行すると、以下のような出力が表示されます。

```
Successfully attached a subscription for: Red Hat Satellite Infrastructure Subscription
```

4. オプション: Satellite Infrastructure サブスクリプションが割り当てられていることを確認します。

```
# subscription-manager list --consumed
```

3.4. リポジトリーの設定

これらの手順を使用して、Satellite Server のインストールに必要なリポジトリーを有効にします。

1. すべてのリポジトリーを無効にします。

```
# subscription-manager repos --disable "*"
```

2. 以下のリポジトリーを有効にします。

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

3. DNF モジュールを有効にします。

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

注記

satellite:el8 モジュールを有効にする際に Ruby または PostgreSQL との競合に関する警告が表示される場合は、付録A *DNF モジュールのトラブルシューティング*を参照してください。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。

3.5. 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、セキュリティー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。

3.5.1. 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。

3.6. SATELLITE SERVER パッケージのインストール

手順

1. すべてのパッケージを更新します。

```
# dnf update
```

2. Satellite Server パッケージをインストールします。

```
# dnf install satellite
```

3.7. 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
```

3.8. 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の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
```

3.9. 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。

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。

3.9.1. 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、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。

3.10. 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。

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"
```

リポジトリーを有効にし、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4章 SATELLITE SERVER での追加設定の実行

4.1. SATELLITE SERVER での RED HAT INSIGHTS の使用

Red Hat Insights を使用すると、セキュリティー違反、パフォーマンスの低下、および安定性の消失に関連するシステムとダウンタイムを診断できます。ダッシュボードを使用して、安定性、セキュリティー、およびパフォーマンスの主要なリスクを素早く特定できます。また、カテゴリー別に分類したり、影響度および解決方法の詳細を表示したり、影響を受けたシステムを調べたりすることができます。

サブスクリプションマニフェストに Red Hat Insights のエンタイトルメントを追加する必要がない点に注意してください。Satellite および Red Hat Insights の詳細は、Red Hat Insights on Satellite Red Hat Enterprise Linux (RHEL) を参照してください。

Satellite Server を保守し、Satellite で発生する可能性のある問題を監視および診断する能力を向上させるには、Satellite Server に Red Hat Insights をインストールし、Satellite Server を Red Hat Insights に登録します。

insights-client のスケジューリング

Satellite に **insights-client.timer** を設定することで、デフォルトの **insights-client** 実行スケジュールを変更できる点に留意してください。詳細は、**Red Hat Insights のクライアント設定ガイド** の insights-client スケジュールの変更 を参照してください。

手順

1. Satellite Server で Red Hat Insights をインストールするには、以下のコマンドを入力します。

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

2. Satellite Server を Red Hat Insights に登録するには、以下のコマンドを入力します。

```
# satellite-installer --register-with-insights
```

4.2. RED HAT INSIGHTS の登録の無効化

Satellite のインストールまたはアップグレード後に、必要に応じて Red Hat Insights の登録または登録解除を選択できます。たとえば、オフライン環境で Satellite を使用する必要がある場合は、Satellite Server から **insights-client** の登録を解除できます。

前提条件

1. Satellite を Red Hat カスタマーポータルに登録している。

手順

1. オプション: Satellite Server から Red Hat Insights の登録を解除するには、以下のコマンドを入力します。

```
# insights-client --unregister
```

2. オプション: Satellite Server を Red Hat Insights に登録するには、以下のコマンドを入力します。

```
# satellite-installer --register-with-insights
```

4.3. SATELLITE CLIENT 6 リポジトリーの有効化と同期

Satellite Client 6 リポジトリーは、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 にリポジトリーを定期的に同期し、ホスト上でリポジトリーを有効にする必要があります。

4.3.1. Red Hat Enterprise Linux 9 と Red Hat Enterprise Linux 8 の Satellite Client 6 リポジトリーの同期

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 リポジトリーを同期します。

```
# 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

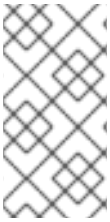
- Satellite Client 6 リポジトリーを同期します。

```
# 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。

4.3.2. Red Hat Enterprise Linux 7 と Red Hat Enterprise Linux 6 の Satellite Client 6 リポジトリーの同期



注記

Red Hat Enterprise Linux 6 のリポジトリーを同期するには、**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 リポジトリーを同期します。

```
# 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 リポジトリーを同期します。

```
# 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。

4.3.3. Red Hat Enterprise Linux 9 と Red Hat Enterprise Linux 8 の Satellite Client 6 リポジトリーの有効化

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

手順

1. Satellite Web UI で、**コンテンツ** > **Red Hat リポジトリー** に移動します。
2. Available Repositories ペインで、**Recommended Repositories** を有効にして、リポジトリー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、リポジトリー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ーを有効にします。**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ーを追加し、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 リポジトリーを有効にします。Red Hat リポジトリーの有効後に、このリポジトリー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 リポジトリーを有効にします。

```
# 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

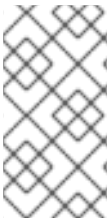
- Satellite Client 6 リポジトリーを有効にします。

```
# 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。

4.3.4. Red Hat Enterprise Linux 7 と Red Hat Enterprise Linux 6 の Satellite Client 6 リポジトリーの有効化



注記

Red Hat Enterprise Linux 6 のリポジトリーを有効にするには、**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

手順

1. Satellite Web UI で、**コンテンツ** > **Red Hat リポジトリー** に移動します。
2. **Available Repositories** ペインで、**Recommended Repositories** を有効にして、リポジトリー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ー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ーを追加し、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。
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ーを有効にします。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 リポジトリーを有効にします。Red Hat リポジトリーの有効後に、このリポジトリー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 リポジトリーを有効にします。

```
# 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 リポジトリーを有効にします。

```
# 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。

4.4. 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設定されます。

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。

4.5. IPV6 ネットワークでの UEFI HTTP ブート向けの SATELLITE の設定

以下の手順を使用して、UEFI HTTP ブートプロビジョニングで IPv6 ネットワークのホストをプロビジョニングするように Satellite を設定します。

前提条件

- クライアントが DHCP および HTTP サーバーにアクセスできることを確認します。
- クライアントが DHCP の要求と応答を送受信できるように、クライアントが UDP ポート 67 および 68 にアクセス可能であることを確認します。
- Satellite および Capsule からファイルおよびキックスタートテンプレートをダウンロードできるように、クライアントに対して TCP ポート 8000 が解放してあることを確認します。
- ホストプロビジョニングインターフェイスサブネットに HTTP ブート Capsule、テンプレート Capsule セットがあることを確認します。詳細は、**ホストのプロビジョニング** の Satellite Server へのサブネットの追加 を参照してください。
- Satellite Web UI で、**Administer** > **Settings** > **Provisioning** に移動し、**Token duration** の設定が **0** に設定されていないことを確認します。Satellite は、DHCPv6 サービスがマネージド外であるため、リモートの IPv6 アドレスでネットワークから起動するクライアントを特定できないので、プロビジョニングトークンは有効化しておく必要があります。

手順

1. インストーラーでの DHCP 管理を無効にするか、使用しないようにします。
2. IPv6 サブネットが Satellite で作成されている場合にはすべて、**DHCP Capsule** を空白に設定します。
3. オプション: ホストおよび DHCP サーバーがルーターで隔てられている場合は、DHCP リレーエージェントを設定し、DHCP サーバーを指定しておく。
4. プロビジョニング元の Satellite または Capsule で、**grub2-efi** パッケージを最新版に更新します。

```
# satellite-maintain packages update grub2-efi
```

5. Red Hat Enterprise Linux 8 キックスタートリポジトリーを同期します。

4.6. HTTP プロキシーを使用した SATELLITE SERVER の設定

以下の手順を使用して、HTTP プロキシーで Satellite を設定します。

4.6.1. デフォルトの HTTP プロキシーを Satellite に追加する

ネットワークで HTTP プロキシーを使用している場合は、Red Hat コンテンツ配信ネットワーク (CDN) または別のコンテンツソースへの要求送信に HTTP プロキシーを使用するように Satellite Server を設定できます。ネットワークの変更が原因で接続が失われるのを回避するために、可能な限り IP の代わりに FQDN を使用します。

以下の手順では、Satellite のコンテンツダウンロード専用のプロキシーを設定します。Satellite Web UI の代わりに CLI を使用する場合は、CLI 手順 を参照してください。

手順

1. Satellite Web UI で、**Infrastructure** > **HTTP Proxies** に移動します。
2. **新しい HTTP プロキシー** をクリックします。
3. **名前** フィールドで、HTTP プロキシーの名前を入力します。
4. **Url** フィールドで、**https://proxy.example.com:8080** の形式で HTTP プロキシーの URL を入力します。
5. オプション: 認証が必要な場合には、**Username** フィールドに認証に使用するユーザー名を入力します。
6. オプション: 認証が必要な場合には、**Password** フィールドに認証に使用するパスワードを入力します。
7. プロキシーへの接続をテストするには、**Test Connection** をクリックします。
8. **Submit** をクリックします。
9. Satellite Web UI で、**Administer** > **Settings** に移動して、**Content** タブをクリックします。
10. 作成した HTTP プロキシーに **Default HTTP Proxy** 設定を指定します。

CLI 手順

1. **http_proxy**、**https_proxy** および **no_proxy** 変数が設定されていないことを確認します。

```
# unset http_proxy
# unset https_proxy
# unset no_proxy
```

2. HTTP プロキシーエントリーを Satellite に追加します。

```
# hammer http-proxy create --name=myproxy \
--url http://myproxy.example.com:8080  \
--username=proxy_username \
--password=proxy_password
```

3. Satellite がデフォルトでこの HTTP プロキシーを使用するように設定します。

```
# hammer settings set --name=content_default_http_proxy --value=myproxy
```

4.6.2. カスタムポートで Satellite にアクセスできるように SELinux を設定する

SELinux は、Red Hat Satellite および Subscription Manager のアクセスを特定のポートに限定します。HTTP キャッシュの場合には、TCP ポートは 8080、8118、8123、および 10001-10010 を使用できます。SELinux タイプが **http_cache_port_t** のポートを使用する場合には、以下の手順を実行してください。

手順

1. Satellite で以下のコマンドを実行して、SELinux で HTTP キャッシュに許可されているポートを確認します。

```
# semanage port -l | grep http_cache
http_cache_port_t       tcp    8080, 8118, 8123, 10001-10010
[output truncated]
```

2. 以下のコマンドを実行して、SELinux が HTTP キャッシュにポート (たとえば、8088) を許可するよう設定します。

```
# semanage port -a -t http_cache_port_t -p tcp 8088
```

4.6.3. すべての Satellite HTTP 要求に HTTP プロキシーを使用する

Satellite Server は、HTTP および HTTPS をブロックするファイアウォールの内側に設定する必要がある場合に、コンピュートリソースなどの外部システムとの通信に使用するプロキシーを設定してください。

プロビジョニングにコンピュートリソースを使用し、コンピュートリソースと、異なる HTTP プロキシーを併用する場合には、コンピュートリソースに設定したプロキシーではなく、Satellite 通信すべてに設定したプロキシーが優先されます。

手順

1. Satellite Web UI で、**Administer** > **Settings** に移動します。
2. **HTTP(S) プロキシー** 行で、隣接する **Value** 列を選択し、プロキシー URL を入力します。
3. チェックのアイコンをクリックして変更を保存します。

CLI 手順

- 以下のコマンドを入力します。

```
# hammer settings set --name=http_proxy --value=Proxy_URL
```

4.6.4. プロキシー化された要求を受信しないようにホストを除外する

すべての Satellite HTTP または HTTPS 要求に HTTP プロキシーを使用する場合、特定のホストがプロキシー経由で通信するのを防ぐことができます。

手順

1. Satellite Web UI で、**Administer** > **Settings** に移動します。
2. **HTTP(S) proxy except hosts**の行で、隣接する **Value** の列を選択して、プロキシー要求から除外する、1 つまたは複数のホストの名前を入力します。
3. チェックのアイコンをクリックして変更を保存します。

CLI 手順

- 以下のコマンドを入力します。

```
# hammer settings set --name=http_proxy_except_list --value=[hostname1.hostname2...]
```

4.6.5. HTTP プロキシーのリセット

現在の HTTP プロキシーの設定をリセットする場合には、**Default HTTP Proxy** 設定を解除します。

手順

1. Satellite Web UI で、**Administer** > **Settings** に移動して、**Content** タブをクリックします。
2. **Default HTTP Proxy** の設定を **no global default** に指定します。

CLI 手順

- **content_default_http_proxy** の設定を空の文字列に設定します。

```
# hammer settings set --name=content_default_http_proxy --value=""
```

4.7. 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"
```

4.8. 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。

4.8.1. 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。

4.8.2. 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、想定のレ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。

4.8.3. 関連情報

- DNS、DHCP、TFTP を外部で設定する方法の詳細は、*6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。

4.9. 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

- スパム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4.1 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
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4.2 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4.3 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

セキュリティー上の理由から、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、**sendmail 1** 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。

4.10. SATELLITE 向けの別の CNAME の設定

Satellite 向けに別の CNAME を設定できます。これは、Satellite に接続するクライアントシステムとは別のドメイン名で、Satellite Web インターフェイスをデプロイする場合に便利です。新規証明書をホストにもう一度デプロイしなくてもいいように、Capsule をインストールして Satellite にホストを登録する前に、別の CNAME 設定を事前に計画しておく必要があります。

4.10.1. 別の CNAME を使用した Satellite の設定

以下の手順を使用して、別の CNAME で Satellite を設定します。デフォルトの Satellite 証明書のユーザーとカスタム証明書のユーザーでは、手順が異なることに留意してください。

デフォルトの Satellite 証明書ユーザーの場合

- デフォルトの Satellite 証明書で Satellite をインストールし、別の CNAME で Satellite を設定する場合には、Satellite で以下のコマンドを入力して、追加の CNAME で新たにデフォルトの Satellite SSL 証明書を生成します。

```
# satellite-installer --certs-cname alternate_fqdn --certs-update-server
```

- Satellite をインストールしていない場合には、**satellite-installer** コマンドに **--certs-cname alternate_fqdn** オプションを追加して Satellite を別の CNAME でインストールしてください。

カスタム証明書ユーザーの場合

カスタム証明書で Satellite を使用する場合は、カスタム証明書の作成時に、別の CNAME レコードをカスタム証明書に追加します。詳細は、Satellite Server 向けのカスタム SSL 証明書の作成 を参照してください。

4.10.2. コンテンツ管理に別の Satellite CNAME を使用するようにホストを設定する

Satellite が別の CNAME で設定されている場合には、コンテンツ管理にもう 1 つの Satellite CNAME を使用するようにホストを設定できます。これには、ホストがもう 1 つの Satellite CNAME を参照するように設定してから、Satellite に登録する必要があります。この設定は、ブートストラップスクリプトを使用するか、手動で実行できます。

ブートストラップスクリプトを使用したホストの設定

ホストで **--server alternate_fqdn.example.com** オプションを指定してブートストラップスクリプトを実行し、ホストを別の Satellite CNAME に登録します。

```
# ./bootstrap.py --server alternate_fqdn.example.com
```

ホストの手動設定

ホストで **/etc/rhsm/rhsm.conf** ファイルを編集して、以下のように別のホスト名を参照するように **hostname** および **baseurl** 設定を更新します。

```
[server]
# Server hostname:
hostname = alternate_fqdn.example.com

content omitted

[rhsm]
# Content base URL:
baseurl=https://alternate_fqdn.example.com/pulp/content/
```

これで、**subscription-manager** でホストを登録できました。

4.11. 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 証明書であってはなりません。
- 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。

4.11.1. 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ーを作成します。

```
# 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ことになります。

4.11.2. 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、証明書のコモンネーム (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。

4.11.3. 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
```

4.12. 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。

4.12.1. 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別してください。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になります。
- 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。

4.12.2. 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ーを無効にし、以下のリポジトリー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。

4.12.3. 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-evr 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
```

7. 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
```

8. 外部 PostgreSQL サーバーで **postgresql** ポートを開きます。

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

9. 変更を永続化します。

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

10. **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
```

11. 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;
```

12. 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".
```

13. **hstore** エクステンションを作成します。

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

14. **postgres** ユーザーをログアウトします。

```
# \q
```

15. 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"
```

4.12.4. 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5章 外部認証の設定

外部認証を使用して、外部 ID プロバイダーのユーザーグループメンバーシップからユーザーとユーザーグループのパーミッションを派生させることができます。外部認証を使用する場合には、このようなユーザーを作成したり、グループメンバーシップを Satellite Server で手動で保守したりする必要はありません。外部ソースでメールが提供されない場合は、Satellite Web UI で最初のログイン時に要求されます。

重要なユーザーおよびグループアカウント情報

ユーザーおよびグループアカウントはすべて、ローカルアカウントである必要があります。これにより、Satellite Server 上のローカルアカウントと Active Directory ドメイン内のアカウントによる認証競合が避けられます。

ユーザーおよびグループアカウントが **/etc/passwd** と **/etc/group** ファイルの両方に存在すれば、この競合によってシステムが影響を受けることはありません。たとえば、**puppet**、**apache**、**foreman** および **foreman-proxy** グループのエントリーが **/etc/passwd** と **/etc/group** の両ファイルに存在することを確認するには、以下のコマンドを実行します。

```
# cat /etc/passwd | grep 'puppet\|apache\|foreman\|foreman-proxy'
# cat /etc/group | grep 'puppet\|apache\|foreman\|foreman-proxy'
```

外部認証の設定シナリオ

Red Hat Satellite では、外部認証の設定において以下の一般的なシナリオがサポートされます。

- **Lightweight Directory Access Protocol** (LDAP) サーバーを外部 ID プロバイダーとして使用するシナリオ。LDAP は、一元的に保存された情報にネットワークを介してアクセスするために使用されるオープンプロトコルセットです。Satellite では、Satellite Web UI を介して LDAP 全体を管理できます。詳細は、「LDAP の使用」 を参照してください。LDAP を使用して Red Hat Identity Management または AD サーバーに接続できますが、セットアップでは、Satellite の Web UI でのサーバー検出、フォレスト間信頼、または Kerberos を使用したシングルサインオンはサポートされません。

- Red Hat Identity Management サーバーを外部 ID プロバイダーとして使用するシナリオ。Red Hat Identity Management は、ネットワーク環境で使用される個別 ID、認証情報、および権限を管理します。Red Hat Identity Management を使用した設定は、Satellite Web UI のみを使用して完了できず、CLI との対話が必要です。詳細は、「Red Hat Identity Management の使用」を参照してください。

- フォレスト間 Kerberos 信頼を介して Red Hat Identity Management に統合された **Active Directory** (AD) を外部 ID プロバイダーとして使用するシナリオ。詳細は、「フォレスト間信頼を使用する Active Directory」 を参照してください。

- Red Hat Single Sign On を Satellite への外部認証用の OpenID プロバイダーとして使用するシナリオ。詳細は、「Red Hat Single Sign-On 認証を使用した Satellite の設定」 を参照してください。

- TOTP を使用した Satellite への外部認証に Red Hat Single Sign-On を OpenID プロバイダーとして使用するシナリオ。詳細は、「TOTP を使用した Red Hat Single Sign On 認証の設定」 を参照してください。

Satellite でプロビジョニングしたホストは、Satellite Server にアクセスできるだけでなく、Red Hat Identity Management レルムと統合することもできます。Red Hat Satellite には、レルムまたはドメインプロバイダーに登録されたシステムのライフサイクルを自動的に管理するレルム機能があります。詳細は、「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。

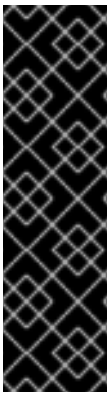
表5.1 認証の概要

型	認証	ユーザーグループ
Red Hat Identity Management	Kerberos または LDAP	あり
Active Directory	Kerberos または LDAP	あり
POSIX	LDAP	あり

5.1. LDAP の使用

Satellite は、1 つまたは複数の LDAP ディレクトリーを使用した LDAP 認証をサポートします。

Red Hat Satellite で **TLS** を使用してセキュアな LDAP 接続 (LDAPS) を確立する必要がある場合は、まず、接続先の LDAP サーバーで使用する証明書を取得して、以下の説明のように Satellite Server のベースオペレーティングシステムでこの証明書を信頼済みとしてマークします。LDAP サーバーで中間認証局との証明書チェーンを使用する場合は、すべての証明書が取得されるように、チェーン内のすべてのルートおよび中間証明書が信頼済みである必要があります。この時点でセキュアな LDAP を必要としない場合は、「Red Hat Satellite で LDAP を使用する設定」 に進みます。



重要

ユーザーは、Red Hat Identity Management と LDAP の両方を認証方法として使用することはできません。ユーザーが 1 つの方法を使用して認証されると、他の方法を使用することはできません。

ユーザーの認証方法を変更するには、自動的に作成されたユーザーを Satellite から削除する必要があります。

Red Hat Identity Management を認証方法として使用する方法の詳細については、「Red Hat Identity Management の使用」 を参照してください。

5.1.1. セキュア LDAP 向けの TLS の設定

Satellite CLI を使用して、セキュア LDAP (LDAPS) 向けに TLS を設定します。

手順

1. LDAP サーバーから証明書を取得します。
   a. Active Directory 証明書サービスを使用する場合は、ベース 64 エンコード X.509 形式を使用してエンタープライズ PKI CA 証明書をエクスポートします。Active Directory サーバーでの CA 証明書の作成およびエクスポートについては、How to configure Active Directory authentication with **TLS** on Satellite を参照してください。
   b. LDAP サーバー証明書を一時的な場所の Satellite Server にダウンロードし、終了したら削除します。
   たとえば、**/tmp/example.crt** です。ファイル名の拡張子を **.cer** と **.crt** にすることが唯一の規則であり、この拡張子は、DER バイナリーまたは PEM ASCII の形式の証明書を参照できます。
2. LDAP サーバーからの証明書を信頼します。

Satellite Server では、LDAP 認証用の CA 証明書は **/etc/pki/tls/certs/** ディレクトリー内の個別ファイルである必要があります。

a. **install** コマンドを使用して適切なパーミッションでインポート済みの証明書を **/etc/pki/tls/certs/** ディレクトリーにインストールします。

```
# install /tmp/example.crt /etc/pki/tls/certs/
```

b. **root** で以下のコマンドを実行して、LDAP サーバーから取得した **example.crt** 証明書を信頼します。

```
# ln -s example.crt /etc/pki/tls/certs/$(openssl \
x509 -noout -hash -in \
/etc/pki/tls/certs/example.crt).0
```

c. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

5.1.2. Red Hat Satellite で LDAP を使用する設定

Satellite Web UI で、LDAP を使用するように Satellite を設定します。

Satellite web UI で Kerberos を使用したシングルサインオン機能が必要な場合は、代わりに Red Hat Identity Management および AD 外部認証を使用する必要があることに注意してください。詳細は以下を参照してください。

- 「Red Hat Identity Management の使用」
- 「Active Directory の使用」

手順

1. Network Information System (NIS) サービスのブール値を true に設定して SELinux により LDAP の送信接続がブロックされないようにします。

```
# setsebool -P nis_enabled on
```

2. Satellite Web UI で、**Administer** > **Authentication Sources** に移動します。
3. **Create LDAP Authentication Source**をクリックします。
4. **LDAP サーバー**タブで LDAP サーバーの名前、ホスト名、ポート、およびサーバータイプを入力します。デフォルトポートは 389、デフォルトサーバータイプは POSIX です (認証サーバーのタイプに応じて FreeIPA または Active Directory を選択することもできます)。**TLS** 暗号化接続に対しては、**LDAPS** チェックボックスを選択して暗号化を有効にします。ポートは LDAPS のデフォルト値である 636 に変更されるはずです。
5. **アカウント** タブで、アカウント情報とドメイン名の詳細を入力します。説明と例については、「LDAP 設定の説明」 を参照してください。
6. **属性マッピング** タブで、LDAP 属性を Satellite 属性にマッピングします。ログイン名、名、姓、メールアドレス、および写真の属性をマッピングできます。サンプルについては、「LDAP 接続の設定例」 を参照してください。

7. **ロケーション** タブで、左側の表からロケーションを選択します。選択したロケーションは、LDAP 認証ソースから作成されたユーザーに割り当てられ、初回ログイン以降、利用可能となります。
8. **組織** タブで、左側の表から組織を選択します。選択した組織は、LDAP 認証ソースから作成されたユーザーに割り当てられ、初回ログイン以降、利用可能となります。
9. **Submit** をクリックします。
10. LDAP ユーザーの新しいアカウントを設定します。
    - **Automatically Create Accounts In Satellite**チェックボックスを選択しなかった場合は、**Red Hat Satellite の管理** の ユーザーの作成 を参照して、ユーザーアカウントを手動で作成します。
    - **Satellite でアカウントを自動作成する** のチェックボックスを選択した場合は、LDAP ユーザーは LDAP アカウントおよびパスワードを使用して Satellite にログインできます。初回ログイン後に、Satellite 管理者はロールを手動で割り当てる必要があります。Satellite でユーザーアカウントに適切なロールを割り当てる方法の詳細は、**Red Hat Satellite の管理** の ユーザーへのロールの割り当て を参照してください。

5.1.3. LDAP 設定の説明

次の表では、**Account** タブの各設定について説明します。

表5.2 Account タブの設定

設定	説明
アカウント	LDAP サーバーへの読み取りアクセス権のある LDAP アカウントのユーザー名。ユーザー名は、サーバーで匿名の読み取りが許可されている場合は必要ありません。以下に例を示します。 `uid=$login,cn=users,cn=accounts,dc=example,dc=com` **$login** 変数には、ログインページで入力されたユーザー名がリテラル文字列として格納されます。この値は、変数がデプロイメントされたときにアクセスされます。 この変数は、LDAP ソースからの外部ユーザーグループとは使用できません。ユーザーがログインしていない場合、Satellite はグループリストを取得する必要があります。匿名または専用サービスユーザーを使用してください。
アカウントパスワード	**アカウント** フィールドで定義されたユーザーの LDAP パスワード。**アカウント** が **$login** 変数を使用している場合は、このフィールドを空白にすることができます。
ベース DN	LDAP ディレクトリーの最上位のドメイン名。
グローバルベース DN	グループが含まれる LDAP ディレクトリーツリーの最上位のドメイン名。
LDAP フィルター	LDAP クエリーを制限するフィルター。

設定	説明
Satellite でアカウントを自動作成する	このチェックボックスを選択した場合には、LDAP ユーザーが Satellite に最初にログインしたときに、Satellite によりユーザーアカウントが作成されます。初回ログイン後に、Satellite 管理者はロールを手動で割り当てる必要があります。**Red Hat Satellite の管理** の ユーザーへのロールの割り当て を参照して、Satellite でユーザーアカウントに適切なロールを割り当てます。
ユーザーグループの同期	このオプションが選択された場合は、ユーザーのログイン時にユーザーのユーザーグループメンバーシップが自動的に同期され、メンバーシップは常に最新の状態になります。このオプションが選択されていない場合は、Satellite で cron ジョブを使用してグループメンバーシップを定期的 (デフォルトでは 30 分ごと) に同期します。詳細は、「外部ユーザーグループの設定」 を参照してください。

5.1.4. LDAP 接続の設定例

以下の表は、異なる種類の LDAP 接続の設定例を示しています。以下の例では、ユーザーおよびグループのエントリーに対してバインド、読み取り、および検索のパーミッションを持つ **redhat** という名前の専用サービスアカウントを使用します。LDAP 属性名は、大文字と小文字が区別されることに注意してください。

表5.3 Active Directory、Free IPA または Red Hat Identity Management、POSIX LDAP 接続の設定例

設定	Active Directory	FreeIPA または Red Hat Identity Management	POSIX (OpenLDAP)
アカウント	DOMAIN\redhat	uid=redhat,cn=users,cn=accounts,dc=example,dc=com	uid=redhat,ou=users,dc=example,dc=com
アカウントパスワード	P@ssword	-	-
ベース DN	DC=example,DC=COM	dc=example,dc=com	dc=example,dc=com
グループベース DN	CN=Users,DC=example,DC=com	cn=groups,cn=accounts,dc=example,dc=com	cn=employee,ou=userclass,dc=example,dc=com
ログイン名属性	userPrincipalName	uid	uid
名属性	givenName	givenName	givenName
姓属性	sn	sn	sn
メールアドレス属性	mail	mail	mail

設定	Active Directory	FreeIPA または Red Hat Identity Management	POSIX (OpenLDAP)
写真属性	thumbnailPhoto	-	-

注記

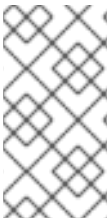
userPrincipalName では、ユーザー名に空白文字を使用できます。ログイン名属性 **sAMAccountName** (上記の表にはリストされていない) は、レガシーの Microsoft システムとの後方互換性を提供します。**sAMAccountName** では、ユーザー名に空白文字を使用できません。

5.1.5. LDAP フィルターの例

管理者は LDAP フィルターを作成することで、特定のユーザーの Satellite へのアクセスを制限することができます。

表5.4 特定ユーザーのログインを許可するフィルターの例

ユーザー	フィルター
User1	(distinguishedName=cn=User1,cn=Users,dc=domain,dc=example)
User1、User3	(memberOf=cn=Group1,cn=Users,dc=domain,dc=example)
User2、User3	(memberOf=cn=Group2,cn=Users,dc=domain,dc=example)
User1、User2、User3	(|(memberOf=cn=Group1,cn=Users,dc=domain,dc=example) (memberOf=cn=Group2,cn=Users,dc=domain,dc=example))
User1、User2、User3	(memberOf:1.2.840.113556.1.4.1941:=cn=Users,dc=domain,dc=example)



注記

グループの **Users** は、グループ **Group1** と **Group2** を含むネストされたグループです。ネストされたグループからすべてのユーザーをフィルターする場合は、ネストされたグループ名の前に **memberOf:1.2.840.113556.1.4.1941:=** を追加する必要があります。上の表の最後の例を参照してください。

LDAP ディレクトリー構造

上記の例のフィルターで使用される LDAP ディレクトリー構造

```
DC=Domain,DC=Example
   |
   |----- CN=Users
          |
          |----- CN=Group1
          |----- CN=Group2
```

```
|----- CN=User1
|----- CN=User2
|----- CN=User3
```

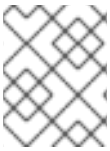
LDAP グループメンバーシップ

上記の例のフィルターで使用されるグループメンバーシップ

グループ	メンバー
Group1	User1、User3
Group2	User2、User3

5.2. RED HAT IDENTITY MANAGEMENT の使用

本項では、Satellite Server と Red Hat Identity Management サーバーを統合する方法とホストベースアクセス制御を有効にする方法を示します。



注記

Red Hat Identity Management は、外部認証ソースとして、シングルサインオンサポートなしで接続できます。詳細は、「LDAP の使用」 を参照してください。

重要

ユーザーは、Red Hat Identity Management と LDAP の両方を認証方法として使用することはできません。ユーザーが 1 つの方法を使用して認証されると、他の方法を使用することはできません。

ユーザーの認証方法を変更するには、自動的に作成されたユーザーを Satellite から削除する必要があります。

前提条件

- Satellite Server のベースオペレーティングシステムが、組織の Red Hat Identity Management 管理者によって Red Hat Identity Management ドメインに登録されていること。

この章の例では、Red Hat Identity Management と Satellite の設定が分離されていることを前提としています。ただし、両方のサーバーの管理者権限を持っている場合は、Red Hat Enterprise Linux 8 Installing Identity Management Guide の説明に従って Red Hat Identity Management を設定できます。

5.2.1. Satellite Server での Red Hat Identity Management 認証の設定

Satellite CLI で、まず Red Hat Identity Management サーバーにホストエントリーを作成して、Red Hat Identity Management 認証を設定します。

手順

1. Red Hat Identity Management サーバーで、次のコマンドを入力し、プロンプトが表示されたら、パスワードを入力して、認証します。

```
# kinit admin
```

2. 認証されたことを確認するには、次のコマンドを入力します。

```
# klist
```

3. 以下のように、Red Hat Identity Management サーバー上で Satellite Server のホストエントリーを作成し、ワンタイムパスワードを生成します。

```
# ipa host-add --random hostname
```

注記

Red Hat Identity Management 登録を完了するには、生成されたワンタイムパスワードをクライアントで使用する必要があります。

ホスト設定プロパティーの詳細は、**Identity Management の設定と管理** の IdM LDAP のホストエントリー を参照してください。

4. 以下のように、Satellite Server 向けの HTTP サービスを作成します。

```
# ipa service-add HTTP/hostname
```

サービスの管理に関する詳細は、Red Hat Enterprise Linux 8 Identity Management サービスへのアクセス を参照してください。

5. Satellite Server で、IPA クライアントをインストールします。

警告

このコマンドは、パッケージのインストール中に Satellite サービスを再起動する可能性があります。Satellite でのパッケージのインストールおよび更新に関する詳細は、**Red Hat Satellite の管理** の Satellite Server または Capsule Server のベースオペレーティングシステムでのパッケージの管理 を参照してください。

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

6. Satellite Server で、以下のコマンドを root として入力し、Red Hat Identity Management 登録を設定します。

```
# ipa-client-install --password OTP
```

OTP を、Red Hat Identity Management 管理者により提供されたワンタイムパスワードに置き換えます。

7. 次のコマンドのいずれかを使用して、Red Hat Identity Management を認証プロバイダーとして設定します。

- Satellite API ではなく、Satellite Web UI へのアクセスのみを有効にする場合は、次のように入力します。

```
# satellite-installer \
--foreman-ipa-authentication=true
```

- Satellite Web UI と Satellite API の両方へのアクセスを有効にする場合は、次のように入力します。

```
# satellite-installer \
--foreman-ipa-authentication-api=true \
--foreman-ipa-authentication=true
```

警告

Satellite API と Satellite Web UI の両方へのアクセスを有効にすると、セキュリティー上の問題が発生する可能性があります。IdM ユーザーが **kinit user_name** を入力して Kerberos Ticket-Granting Ticket (TGT) を受け取った後、攻撃者は API セッションを取得できます。ユーザーが以前にブラウザーなど、どこにも Satellite ログイン認証情報を入力していなかった場合でも、攻撃は可能です。

8. Satellite サービスを再起動します。

```
# satellite-maintain service restart
```

この時点で、外部ユーザーは Red Hat Identity Management 認証情報を使用して Satellite にログインできます。ユーザー名とパスワードを使用して直接 Satellite Server にログインするか、設定済みの Kerberos シングルサインオンを活用してクライアントマシンでチケットを取得し、自動的にログインするかを選択できます。また、ワンタイムパスワードを使用した二要素認証 (2FA OTP) もサポートされます。

5.2.2. ホストベースの認証制御の設定

HBAC ルールでは、Red Hat Identity Management ユーザーがドメイン内のどのマシンにアクセスできるかを定義します。一部のユーザーが Satellite Server にアクセスできないように、Red Hat Identity Management サーバーで HBAC を設定できます。この方法では、ログインが許可されていないユーザーのデータベースエントリーを、Satellite で作成できないようにします。HBAC の詳細は、Managing IdM Users, Groups, Hosts, and Access Control Rules Guide を参照してください。

Red Hat Identity Management サーバーで、ホストベースの認証制御 (HBAC) を設定します。

手順

1. Red Hat Identity Management サーバーで、次のコマンドを入力し、プロンプトが表示されたら、パスワードを入力して、認証します。

```
# kinit admin
```

2. 認証されたことを確認するには、次のコマンドを入力します。

```
# klist
```

3. HBAC サービスおよびルールを Red Hat Identity Management サーバーで作成し、リンクします。以下の例では、**satellite-prod** という PAM サービス名を使用しています。Red Hat Identity Management サーバー上で以下のコマンドを実行してください。

```
# ipa hbacsvc-add satellite-prod
# ipa hbacrule-add allow_satellite_prod
# ipa hbacrule-add-service allow_satellite_prod --hbacsvcs=satellite-prod
```

4. satellite-prod サービスへのアクセス権があるユーザーと Satellite Server のホスト名を追加します。

```
# ipa hbacrule-add-user allow_satellite_prod --user=username
# ipa hbacrule-add-host allow_satellite_prod --hosts=satellite.example.com
```

   または、**allow_satellite_prod** ルールにホストグループとユーザーグループを追加できます。

5. ルールのステータスを確認するために、以下のコマンドを実行します。

```
# ipa hbacrule-find satellite-prod
# ipa hbactest --user=username --host=satellite.example.com --service=satellite-prod
```

6. Red Hat Identity Management サーバーで allow_all ルールが無効であることを確認します。他のサービスに影響を与えずにこのルールを無効にする方法については、Red Hat カスタマーポータルのアーティクル How to configure HBAC rules in IdM to allow specific users to login to clients via ssh を参照してください。

7. 「Satellite Server での Red Hat Identity Management 認証の設定」 で説明されているように、Satellite Server で Red Hat Identity Management 統合を設定します。Satellite Server で、root として PAM サービスを定義します。

```
# satellite-installer --foreman-pam-service=satellite-prod
```

5.3. ACTIVE DIRECTORY の使用

このセクションでは、Satellite Server 用の外部認証ソースとして直接 Active Directory (AD) を使用する方法を示します。

注記

シングルサインオンサポートなしで、Active Directory を外部認証ソースとして接続できます。詳細は、「LDAP の使用」 を参照してください。設定例については、How to configure Active Directory authentication with TLS on Satellite を参照してください。

直接 AD 統合では、ID が保存されている AD ドメインに Satellite Server が直接参加します。推奨の設定には、以下の 2 つの手順が含まれます。

- 「AD サーバーへの Satellite Server の登録」 の説明に従って、Active Directory サーバーに Satellite Server を登録します。

- 「GSS-proxy を使用した直接 AD 統合の設定」 の説明に従って、GSS-proxy との直接 Active Directory 統合を設定します。

5.3.1. GSS-Proxy

Apache での Kerberos 認証の従来のプロセスでは、Apache プロセスが keytab ファイルへの読み取りアクセスを持っている必要があります。GSS-Proxy を使用すると、Kerberos 認証機能を保持しつつ keytab ファイルへのアクセスを削除することにより Apache サーバーに対してより厳密な権限の分離を実行できます。AD を Satellite の外部認証ソースとして使用する場合は、keytab ファイルのキーがホストキーと同じであるため、GSS-proxy を実装することが推奨されます。

Satellite Server のベースオペレーティングシステムとして動作する Red Hat Enterprise Linux で以下の手順を実行します。本セクションの例では、**EXAMPLE.ORG** が AD ドメインの Kerberos レルムです。手順を完了すると、EXAMPLE.ORG レルムに属するユーザーは Satellite Server にログインできます。

5.3.2. AD サーバーへの Satellite Server の登録

Satellite CLI で、Active Directory サーバーに Satellite Server を登録します。

前提条件

- GSS-proxy と nfs-utils がインストールされていること。
  GSS-proxy と nfs-utils をインストールします。

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

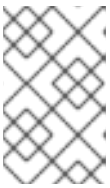
手順

1. 必要なパッケージをインストールします。

```
# satellite-maintain packages install sssd adcli realmd ipa-python-compat krb5-workstation
samba-common-tools
```

2. Satellite Server を AD サーバーに登録します。以下のコマンドを実行するには、管理者パーミッションが必要な場合があります。

```
# realm join -v EXAMPLE.ORG --membership-software=samba -U Administrator
```



注記

「GSS-proxy を使用した直接 AD 統合の設定」 で HTTP キータブを作成するには、Samba クライアントソフトウェアを使用して AD サーバーに登録する必要があります。

5.3.3. GSS-proxy を使用した直接 AD 統合の設定

Satellite CLI で、GSS-proxy を使用する直接 Active Directory 統合を設定します。

前提条件

- Satellite が、Active Directory サーバーに登録されていること。詳細は、「AD サーバーへの Satellite Server の登録」 を参照してください。

手順

1. **/etc/ipa/** ディレクトリーと **default.conf** ファイルを作成します。

```
# mkdir /etc/ipa
# touch /etc/ipa/default.conf
```

2. **default.conf** ファイルに以下のコンテンツを追加します。

```
[global]
server = unused
realm = EXAMPLE.ORG
```

3. 以下の内容で **/etc/net-keytab.conf** ファイルを作成します。

```
[global]
workgroup = EXAMPLE
realm = EXAMPLE.ORG
kerberos method = system keytab
security = ads
```

4. Apache ユーザーの有効なユーザー ID を特定します。

```
# id apache
```

Apache ユーザーには keytab ファイルへのアクセス権を割り当てないでください。

5. 以下の内容で **/etc/gssproxy/00-http.conf** ファイルを作成します。

```
[service/HTTP]
mechs = krb5
cred_store = keytab:/etc/httpd/conf/http.keytab
cred_store = ccache:/var/lib/gssproxy/clients/krb5cc_%U
euid = ID_of_Apache_User
```

6. keytab エントリーを作成します。

```
# KRB5_KTNAME=FILE:/etc/httpd/conf/http.keytab net ads keytab add HTTP -U
administrator -d3 -s /etc/net-keytab.conf
# chown root.apache /etc/httpd/conf/http.keytab
# chmod 640 /etc/httpd/conf/http.keytab
```

7. Satellite で IPA 認証を有効にします。

```
# satellite-installer --foreman-ipa-authentication=true
```

8. **gssproxy** サービスを起動して、有効にします。

```
# systemctl restart gssproxy
# systemctl enable --now gssproxy
```

9. Apache サーバーが **gssproxy** サービスを使用するように設定するには、**systemd** ドロップインファイルを作成し、以下の内容を追加します。

```
# mkdir -p /etc/systemd/system/httpd.service.d/
# vi /etc/systemd/system/httpd.service.d/gssproxy.conf
[Service]
Environment=GSS_USE_PROXY=1
```

10. 変更をサービスに適用します。

```
# systemctl daemon-reload
```

11. **httpd** サービスを起動して、有効にします。

```
# systemctl restart httpd
```

重要

直接 AD 統合では、Red Hat Identity Management を介した HBAC は利用できません。代わりに、管理者が AD 環境でポリシーを一元管理することを可能にする Group Policy Objects (GPO) を使用できます。GPO から PAM サービスへのマッピングが正しいことを確認するには、以下の SSSD 設定を **/etc/sssd/sssd.conf** に追加します。

```
access_provider = ad
ad_gpo_access_control = enforcing
ad_gpo_map_service = +foreman
```

ここでは、**foreman** は PAM サービスの名前です。GPO の詳細は、**RHEL システムと Windows Active Directory を直接統合** の SSSD が GPO アクセス制御ルールを解釈する方法 を参照してください。

検証

SSO が想定どおりに動作していることを確認します。

Apache サーバーが実行中であり、クライアントに有効な Kerberos チケットがある場合、サーバーに対して HTTP 要求を行うユーザーは認証されます。

1. 次のコマンドを使用して、LDAP ユーザーの Kerberos チケットを取得します。

```
# kinit ldapuser
```

2. 以下のコマンドを使用して、Kerberos チケットを表示します。

```
# klist
```

3. 以下のコマンドを使用して、SSO 認証に成功時の出力を表示します。

```
# curl -k -u : --negotiate https://satellite.example.com/users/extlogin
```

これにより、以下の応答が返されます。

```
<html><body>You are being <a href="https://satellite.example.com/users/4-ldapuserexample-com/edit">redirected</a>.</body></html>
```

5.3.4. Web ブラウザーでの Kerberos の設定

Firefox の設定は、**Red Hat Enterprise Linux RHEL での認証および認可の設定**ガイドの Single Sign-On に Kerberos を使用するための Firefox の設定 を参照してください。

Internet Explorer ブラウザーを使用している場合は、Satellite Server をローカルイントラネットまたは信頼済みサイトのリストに追加し、**統合 Windows 認証を使用する** の設定にチェックを入れます。詳細については、Internet Explorer のマニュアルを参照してください。

5.3.5. フォレスト間信頼を使用する Active Directory

Kerberos では、**cross-forest trust** を作成して、2 つの異なるドメインフォレスト間の関係を定義できます。ドメインフォレストとは、ドメインの階層構造のことで、AD と Red Hat Identity Management の両方でフォレストが形成されます。AD と Red Hat Identity Management との間での有効な信頼関係により、AD のユーザーは一連の認証情報を使用して Linux ホストおよびサービスにアクセスできます。フォレスト間の信頼の詳細については、**Red Hat Enterprise Linux Identity Management の計画** の IdM と AD との間のフォレスト間の信頼の計画 を参照してください。

Satellite 側から見ると、設定プロセスは、フォレスト間の信頼を設定せずに Red Hat Identity Management サーバーと統合する場合と同じです。Satellite Server は IdM ドメインに登録し、「Red Hat Identity Management の使用」 で説明されているように統合する必要があります。

5.3.6. フォレスト間信頼を使用するための Red Hat Identity Management サーバーの設定

Red Hat Identity Management サーバーで、**cross-forest trust** を使用するようにサーバーを設定します。

手順

1. HBAC を有効にします。
   a. 外部グループを作成して、この外部グループに AD グループを追加します。
   b. 新しい外部グループを POSIX グループに追加します。
   c. HBAC ルールで POSIX グループを使用します。
2. AD ユーザーの追加属性を転送するよう sssd を設定します。
   - この AD ユーザー属性を **/etc/sssd/sssd.conf** の **nss** セクションと **domain** セクションに追加します。以下に例を示します。

```
[nss]
user_attributes=+mail, +sn, +givenname
[domain/EXAMPLE.com]
...
krb5_store_password_if_offline = True
ldap_user_extra_attrs=email:mail, lastname:sn, firstname:givenname

[ifp]
allowed_uids = ipaapi, root
user_attributes=+email, +firstname, +lastname
```

   - AD 属性値を確認します。

```
# dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe
/org/freedesktop/sssd/infopipe org.freedesktop.sssd.infopipe.GetUserAttr string:ad-
user@ad-domain array:string:email,firstname,lastname
```

5.4. 外部ユーザーグループの設定

Satellite は、自動的に、外部ユーザーグループに外部ユーザーを関連付けることはありません。Satellite 上の外部ソースと同じ名前のユーザーグループを作成する必要があります。こうすることで、外部ユーザーグループのメンバーは、自動的に Satellite ユーザーグループのメンバーになり、関連するパーミッションが付与されます。

外部ユーザーグループの設定は、外部認証の種類によって異なります。

外部ユーザーに追加のパーミッションを割り当てるには、外部マッピングが指定されていない内部ユーザーグループに、このユーザーを追加します。次に、このグループに必要なロールを割り当てます。

前提条件

- LDAP サーバーを使用する場合は、Satellite が LDAP 認証を使用するように設定する。詳細は、「LDAP の使用」 を参照してください。
  LDAP ソースから外部ユーザーグループを使用する場合は、アカウントユーザー名の代わりとして **$login** 変数を使用できず、匿名または専用サービスユーザーを使用する。
- Red Hat Identity Management または AD サーバーを使用する場合は、Satellite が Red Hat Identity Management または AD 認証を使用するように設定する。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 外部認証の設定 を参照してください。
- 少なくとも 1 人の外部ユーザーが初回認証されることを確認する。
- 使用する外部グループ名をメモする。外部ユーザーのグループメンバーシップを確認するには、以下のコマンドを入力します。

```
# id username
```

手順

1. Satellite Web UI で、**管理** > **ユーザーグループ** に移動して、**ユーザーグループの作成** をクリックします。
2. 新規ユーザーグループの名前を指定します。外部ユーザーグループのリフレッシュ時にユーザーが自動的に追加されるのを避けるため、ユーザーを選択しないでください。
3. **ロール** タブをクリックし、ユーザーグループに割り当てるロールを選択します。または、**管理者** のチェックボックスを選択して、利用可能なすべてのパーミッションを割り当てます。
4. **外部グループ** タブで、**外部ユーザーグループの追加** をクリックして、**認証ソース** ドロップダウンメニューから認証ソースを選択します。
   名前 フィールドに外部グループの名前を指定します。
5. **Submit** をクリックします。

5.5. LDAP の外部ユーザーグループのリフレッシュ

ユーザーのログイン時にユーザーのグループメンバーシップを自動的に同期するように LDAP ソースを

設定するには、**認証ソース** ページで、**ユーザーグループの同期** オプションを選択します。このオプションが選択されていない場合、デフォルトで、LDAP ユーザーグループは、 30 分ごとに LDAP 認証ソースを同期するようにスケジュールされた cron ジョブで自動的にリフレッシュされます。

LDAP 認証ソースのユーザーグループが、次回のスケジュールタスクが実行されるまでの間に変更された場合に、ユーザーが不正な外部ユーザーグループに割り当てられることがあります。これはスケジュールされたタスクの実行時に、自動的に修正されます。

以下の手順を使用して、LDAP ソースを手動でリフレッシュします。

手順

1. Satellite web UI で、**Administer** > **Usergroups** に移動し、ユーザーグループを選択します。
2. **External Groups** タブで、必要なユーザーグループの右側にある **Refresh** をクリックします。

CLI 手順

- 以下のコマンドを入力します。

```
# foreman-rake ldap:refresh_usergroups
```

5.6. RED HAT IDENTITY MANAGEMENT または AD の外部ユーザーグループの更新

Red Hat Identity Management または AD ベースの外部ユーザーグループは、グループメンバーが Satellite にログインしたときのみリフレッシュされます。Satellite Web UI で、外部ユーザーグループのユーザーメンバーシップを変更することはできず、このような変更がされた場合には、次のグループリフレッシュ時に上書きされます。

5.7. RED HAT IDENTITY MANAGEMENT ユーザー認証を使用するための HAMMER CLI の設定

このセクションでは、Red Hat Identity Management (IdM) を使用してユーザーを認証するように Satellite Hammer コマンドラインインターフェイス (CLI) ツールを設定する方法について説明します。

前提条件

- Hammer を使用して Satellite にアクセスするホストにログインしている。

手順

1. **~/.hammer/cli.modules.d/foreman.yml** Hammer 設定ファイルでセッションを有効にするには、**foreman** パラメーターに **:use_sessions: true** 行を追加します。

```
:foreman:
  :use_sessions: true
```

この行を追加すると、Hammer で強制的にセッションが使用されます。つまり、Hammer が **Hammer** コマンドごとに認証要求を実行するのではなく、認証要求を 1 回だけ実行するようになります。

2. オプション: ~/**.hammer/cli.modules.d/foreman.yml** Hammer 設定ファイルでネゴシエート認証を有効にするには、**:default_auth_type: 'Negotiate_Auth'** 行を **foreman** パラメーターに追加します。

```
:foreman:
  :default_auth_type: 'Negotiate_Auth'
  :use_sessions: true
```

この行を追加すると、最初の **Hammer** コマンドを入力したときに認証がネゴシエートされることになります。このエントリーが存在する場合、Hammer はネゴシエーションプロトコルを使用して Satellite Server との通信を試みます。

5.8. プロビジョニングされたホストの外部認証

以下のセクションを使用して、Red Hat Identity Management レルムサポート用の Satellite Server または Capsule Server を設定します。続いて、Red Hat Identity Management レルムグループにホストを追加します。

前提条件

- Satellite Server をコンテンツ配信ネットワークに登録しておくか、外部の Capsule Server を Satellite Server に登録しておく。
- Red Hat Identity Management などのレルムまたはドメインプロバイダーがデプロイされていること。

Satellite Server または Capsule Server に Identity Management パッケージをインストールして設定するには:

プロビジョニングされたホストに Identity Management を使用するには、次の手順を実行して、Satellite Server または Capsule Server に Identity Management パッケージをインストールおよび設定します。

1. Satellite Server または Capsule Server に **ipa-client** パッケージをインストールします。

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

2. サーバーを Red Hat Identity Management クライアントとして設定します。

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

3. Red Hat Identity Management でレルムプロキシーユーザー **realm-capsule** と、関連のロールを作成します。

```
# foreman-prepare-realm admin realm-capsule
```

以下の手順で必要となるので、返されたプリンシパル名と、Red Hat Identity Management サーバーの設定情報をメモします。

Red Hat Identity Management レルムのサポートのために Satellite Server または Capsule Server を設定する方法:

Satellite および使用するすべての Capsule で次の手順を実行します。

1. 同じプリンシパルおよびレルムに追加する Capsule Server に、**/root/freeipa.keytab** ファイルをコピーします。

```
# scp /root/freeipa.keytab root@capsule.example.com:/etc/foreman-proxy/freeipa.keytab
```

2. **/root/freeipa.keytab** ファイルを **/etc/foreman-proxy** ディレクトリーに移動して、所有者を **foreman-proxy** ユーザーに設定します。

```
# mv /root/freeipa.keytab /etc/foreman-proxy
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/freeipa.keytab
```

3. レルムに追加する全 Capsule で、以下のコマンドを入力します。Satelllite に 統合された Capsule を使用する場合には、Satellite Server でこのコマンドを入力します。

```
# satellite-installer --foreman-proxy-realm true \
--foreman-proxy-realm-keytab /etc/foreman-proxy/freeipa.keytab \
--foreman-proxy-realm-principal realm-capsule@EXAMPLE.COM \
--foreman-proxy-realm-provider freeipa
```

これらのオプションは、Satellite Server を初めて設定する場合にも使用できます。

4. ca-certificates パッケージの最新バージョンがインストールされていることを確認し、Red Hat Identity Management 認証局を信頼します。

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

5. オプション: 既存の Satellite Server または Capsule Server で Red Hat Identity Management を設定する場合には、以下の手順を実行して、設定の変更が適用されていることを確認します。

   a. **foreman-proxy** サービスを再起動します。

   ```
   # systemctl restart foreman-proxy
   ```

   b. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。

   c. Red Hat Identity Management 用に設定した Capsule の場所を特定して、**アクション** コラムのリストから **リフレッシュ** を選択します。

Red Hat Identity Management 対応の Capsule のレルムの作成方法

統合型または外部の Capsule に Red Hat Identity Management を設定した後に、レルムを作成して、Red Hat Identity Management が設定された Capsule をレルムに追加する必要があります。

手順

1. Satellite Web UI で、**インフラストラクチャー** > **レルム** に移動して、**レルムの作成** をクリックします。

2. **名前** フィールドには、レルムの名前を入力します。

3. **レルムのタイプ** リストから、レルムのタイプを選択します。

4. **Realm Capsule** リストから、Red Hat Identity Management を設定した Capsule Server を選択します。

5. **ロケーション** タブをクリックして、**ロケーション** リストから、新しいレルムを追加するロケーションを選択します。

6. **組織** タブをクリックして、**組織** リストから、新規レルムを追加する組織を選択します。

7. **Submit** をクリックします。

レルム情報によるホストグループの更新

使用するホストグループを、新しいレルム情報で更新する必要があります。

1. Satellite web UI で、**Configure** > **Host Groups** に移動し、更新するホストグループを選択して、**Network** タブをクリックします。

2. **レルム** リストから、この手順の一部で作成するレルムを選択して **送信** をクリックします。

Red Hat Identity Management ホストグループへのホストの追加

Red Hat Identity Management では、システムの属性に基づいて自動メンバーシップルールをセットアップできます。Red Hat Satellite のレルム機能は、管理者に対し、Red Hat Satellite ホストグループを Red Hat Identity Management パラメーター **userclass** にマップする機能を提供します。これにより、管理者は automembership を設定することができます。

ネスト化されたホストグループが使用される場合、それらは Red Hat Satellite ユーザーインターフェイスに表示され、Red Hat Identity Management サーバーに送信されます。たとえば、"Parent/Child/Child" のように表示されます。

Satellite Server または Capsule Server は更新を Red Hat Identity Management サーバーに送信しますが、automembership のルールは、初回登録時にのみ適用されます。

Red Hat Identity Management ホストグループにホストを追加する方法:

1. Red Hat Identity Management サーバーで、ホストグループを作成します。

```
# ipa hostgroup-add hostgroup_name --desc=hostgroup_description
```

2. **automembership** ルールを作成します。

```
# ipa automember-add --type=hostgroup hostgroup_name automember_rule
```

以下のオプションを使用できる場所:

- **automember-add** は automember グループとしてグループにフラグを立てます。
- **--type=hostgroup** は、ターゲットグループがユーザーグループではなく、ホストグループであることを特定します。
- **automember_rule** は、automember ルールの識別に使用する名前を追加します。

3. **userclass** 属性に基づいて automembership の条件を定義します。

```
# ipa automember-add-condition --key=userclass --type=hostgroup --inclusive-
regex=^webserver hostgroup_name
---------------------------------
```

```
Added condition(s) to "hostgroup_name"
----------------------------------
Automember Rule: automember_rule
Inclusive Regex: userclass=^webserver
----------------------------
Number of conditions added 1
----------------------------
```

以下のオプションを使用できる場所:

- **automember-add-condition** では、グループメンバーを識別する正規表現の条件を追加します。
- **--key=userclass** はキー属性を **userclass** に指定します。
- **--type=hostgroup** は、ターゲットグループがユーザーグループではなく、ホストグループであることを特定します。
- **--inclusive-regex= ^webserver** は、正規表現パターンで一致する値を識別します。
- **hostgroup_name**: ターゲットホストグループの名前を識別します。

システムが Satellite Server の **hostgroup_name** ホストグループに追加されると、そのシステムは、Red Hat Identity Management サーバーの "**hostgroup_name**" ホストグループに自動的に追加されます。Red Hat Identity Management ホストグループは、HBAC (ホストベースアクセス制御)、sudo ポリシー、およびその他の Red Hat Identity Management 機能を許可します。

5.9. RED HAT SINGLE SIGN-ON 認証を使用した SATELLITE の設定

Red Hat Single Sign On を外部認証用の OpenID プロバイダーとして使用するように Satellite を設定するには、以下のセクションを使用します。

5.9.1. Red Hat Single Sign-On 認証を使用して Satellite を設定するための前提条件

Red Hat Single Sign-On 外部認証を使用して Satellite を設定する前に、以下の要件を満たすようにしてください。

- HTTP ではなく、HTTPS を使用する Red Hat Single Sign On サーバーを正常にインストールしている。
- 管理者権限を持つ Red Hat Single Sign-On アカウント。
- Red Hat Single Sign-On で作成した Satellite ユーザーアカウントのレルム。
- 証明書または CA が自己署名されている場合は、それらがエンドユーザー証明書トラストストアに追加されていることを確認する。
- ユーザーが Red Hat Single Sign-On にインポートまたは追加されている。
  LDAP や Kerberos などの既存のユーザーデータベースが設定されている場合は、ユーザーのフェデレーションを設定することでユーザーをインポートできます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の ユーザーストレージフェデレーション を参照してください。

  既存のユーザーデータベースが設定されていない場合は、Red Hat Single Sign-On でユーザーを手作業で作成できます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の 新規ユーザーの作成 を参照してください。

5.9.2. Satellite を Red Hat Single Sign-On クライアントとして登録する

以下の手順を使用して、Satellite をクライアントとして Red Hat Single Sign-On に登録し、認証ソースとして Red Hat Single Sign-On を使用するように Satellite を設定します。

Satellite と Red Hat Single Sign-On は、2 つの異なる認証方法で設定できます。

1. Satellite Web UI を使用した Satellite への認証。
2. Satellite CLI を使用した Satellite への認証。

どちらの方法でも、異なる Satellite クライアントを Red Hat Single Sign-On に登録して設定する必要があるため、ユーザーの認証方法を事前に決定する必要があります。この手順では、Red Hat Single Sign-On の Satellite クライアントの登録および設定方法が区別されています。

両認証方法を使用して、どちらのクライアントも適宜設定する場合には、Red Hat Single Sign-On に異なる Satellite クライアントを 2 つ登録することも可能です。

手順

1. Satellite Server で、以下のパッケージをインストールします。

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install python3-lxml
```

2. Satellite をクライアントとして Red Hat Single Sign-On に登録します。Web UI と CLI とでログインの登録プロセスが異なる点に注意してください。Red Hat Single Sign-On に 2 つの Satellite クライアントを登録すると、Web UI と CLI から Satellite にログインできます。

   - Web UI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

     プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

     次に、認証ソースとして Red Hat Single Sign On を使用するように Satellite を設定します。

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

   - CLI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name hammer-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
```

```
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

3. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

5.9.3. Red Hat Single Sign-On での Satellite クライアントの設定

以下の手順を使用して、Red Hat Single Sign-On Web UI で Satellite クライアントを設定し、Satellite クライアントのグループおよびオーディエンスマッパーを作成します。

手順

1. Red Hat Single Sign-On Web UI で、**クライアント** に移動し、Satellite クライアントをクリックします。
2. アクセスタイプを設定します。
   - Satellite web UI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **機密** を選択します。
   - CLI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **公開** を選択します。
3. **有効なリダイレクト URI** フィールドに有効なリダイレクト URI を追加します。
   - Satellite web UI を使用して Satellite への認証を行うには、**https://satellite.example.com/users/extlogin** の形式で URI を入力します。Satellite FQDN の後に **/users/extlogin** の文字列を追加する必要があります。
     この手順の完了後に、Satellite クライアントが Satellite web UI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     https://satellite.example.com/users/extlogin
     ```

   - CLI を使用してユーザーが Satellite への認証を行うには、既存の URI の下の空白フィールドに **urn:ietf:wg:wg:oauth:2.0:oob** を入力します。
     この手順の完了後に、Satellite クライアントが CLI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

     ```
     https://satellite.example.com/users/extlogin/redirect_uri
     urn:ietf:wg:oauth:2.0:oob
     ```

4. **Save** をクリックします。
5. **マッパー** タブ、**作成** の順にクリックし、オーディエンスマッパーを追加します。
6. **名前** フィールドに、オーディエンスマッパーの名前を入力します。
7. **マッパータイプ** リストから、**オーディエンス** を選択します。

8. **組み込み済みクライアントオーディエンス** リストから、Satellite クライアントを選択します。
9. **Save** をクリックします。
10. **作成** をクリックして、グループメンバーシップをもとに Satellite の認証を指定できるようにグループマッパーを追加します。
11. **名前** フィールドにグループマッパーの名前を入力します。
12. **マッパータイプ** リストから、**グループメンバーシップ** を選択します。
13. **トークンクレーム名** に **groups** と入力します。
14. **フルグループパス** のトグルを OFF に設定します。
15. **Save** をクリックします。

5.9.4. Red Hat Single Sign-On 認証用の Satellite の設定

このセクションでは、Satellite Web UI または CLI を使用して Red Hat Single Sign-On 認証用に Satellite を設定します。

5.9.4.1. Web UI を使用した Red Hat Single Sign-On 認証用の Satellite の設定

以下の手順に従って、Satellite Web UI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration** の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **機密** に設定されていることを確認します。

手順

1. Satellite Web UI で、**管理** > **設定** に移動して、**認証** タブをクリックします。
2. **ログイン委任の認証** の行を見つけ、**値** コラムで **Yes** に値を設定します。
3. **Authorize login delegation auth source user autocreate**行を見つけ、**値** コラムで **External** に値を設定します。
4. **Login delegation logout URL** の行を見つけ、**Value** 列で、値を **https://satellite.example.com/users/extlogout** に設定します。
5. **OIDC アルゴリズム** の行を見つけ、**値** コラムで、Red Hat Single Sign On のエンコーティングのアルゴリズムを設定します (例: **RS256**)。
6. **OIDC オーディエンス** 行を見つけ、**値** コラムで、値を Red Hat Single Sign On のクライアント ID に設定します。
7. **OIDC Issuer** 行を見つけ、**Value** 列で、値を **https://RHSSO.example.com/auth/realms/Satellite_Realm** に設定します。

8. **OIDC JWKs URL** 行を見つけ、**Value** 列で、値を **https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs** に設定します。
9. Satellite Web UI で、**Administer** > **Authentication Sources** に移動し、**External** カードの垂直省略記号をクリックして、**Edit** を選択します。
10. **場所** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる場所を追加します。
11. **組織** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる組織を追加します。
12. **Submit** をクリックします。

5.9.4.2. CLI を使用した Red Hat Single Sign-On 認証用の Satellite の設定

以下の手順に従って、Satellite CLI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration** の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **公開** に設定されていることを確認します。

手順

1. Satellite で、ログイン委任を **true** に設定し、ユーザーが Open IDC プロトコルを使用して認証できるようにします。

```
# hammer settings set --name authorize_login_delegation --value true
```

2. ログイン委任のログアウト URL を以下のように設定します。

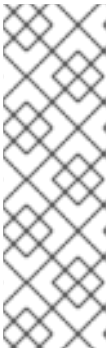
```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

3. Red Hat Single Sign On のエンコーティングのアルゴリズムを設定します (例: **RS256**)。

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

4. URL **RHSSO.example.com/auth/realms/RHSSO_REALM/.well-known/openid-configuration** を開いて値をメモし、以下のステップのオプションに入力します。
5. Open IDC オーディエンスに Hammer クライアントの値を追加します。

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-hammer-openidc']"
```



注記

複数の Red Hat Single Sign-On クライアントを Satellite に登録する場合は、以下のように、アレイに全オーディエンスを必ず追加してください。以下に例を示します。

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-foreman-openidc', 'satellite.example.com-hammer-openidc']"
```

6. Open IDC 発行者の値を設定します。

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

7. Open IDC Java Web Token (JWT) の値を設定します。

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

8. Red Hat Single Sign-On 認証ソースの ID を取得します。

```
# hammer auth-source external list
```

9. ロケーションと組織を設定します。

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

5.9.5. Red Hat Single Sign-On を使用した Satellite Web UI へのログイン

以下の手順に従って、Red Hat Single Sign-On を使用して Satellite Web UI にログインします。

手順

- ブラウザーで Satellite にログインし、認証情報を入力します。

5.9.6. Red Hat Single Sign-On を使用した Satellite CLI へのログイン

以下の手順に従って、コード付与タイプを使用して Satellite CLI への認証を行います。

手順

1. コード付与タイプを使用して Satellite CLI への認証を行うには、以下のコマンドを入力します。

```
# hammer auth login oauth \
--two-factor \
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-connect/token' \
```

```
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \
--oidc-client-id 'satellite.example.com-foreman-openidc' \
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

このコマンドは、サクセスコードの入力を要求します。

2. サクセスコードを取得するには、コマンドが返す URL に移動し、必要な情報を提供します。
3. Web UI が返すサクセスコードをコピーします。
4. **hammer auth login oauth** のコマンドプロンプトに、サクセスコードを入力して Satellite CLI に対して認証を行います。

5.9.7. Red Hat Single Sign-On 認証用のグループマッピングの設定

必要に応じて、ロールベースのアクセス制御 (RBAC) を実装するには、Satellite でグループを作成し、このグループにロールを割り当ててから Active Directory グループを Satellite グループにマッピングします。これにより、Red Hat Single Sign-On の指定のグループに所属する場合には、該当する Satellite グループでログインします。この例では、Active Directory の Satellite-admin ユーザーグループのユーザーを設定し、Satellite で管理者権限を持つユーザーとして認証されるようにします。

手順

1. Satellite Web UI で、**Administer** > **User Groups** に移動します。
2. **Create User Group** をクリックします。
3. **名前** フィールドにユーザーグループの名前を入力します。名前は Active Directory と同じにしないでください。
4. 右側の列には、ユーザーおよびユーザーグループを追加しないでください。**ロール** タブをクリックします。
5. **管理** チェックボックスを選択します。
6. **外部グループ** タブをクリックします。
7. **外部ユーザーグループの追加** をクリックします。
8. **名前** フィールドに、Active Directory の名前を入力します。
9. リストから **外部** を選択します。

5.10. TOTP を使用した RED HAT SINGLE SIGN ON 認証の設定

TOTP カードを使用した外部認証用の OpenID プロバイダーとして Red Hat Single Sign-On を使用するように Satellite を設定するには、以下のセクションを使用します。

5.10.1. Red Hat Single Sign-On 認証を使用して Satellite を設定するための前提条件

Red Hat Single Sign-On 外部認証を使用して Satellite を設定する前に、以下の要件を満たすようにしてください。

- HTTP ではなく、HTTPS を使用する Red Hat Single Sign On サーバーを正常にインストールしている。

- 管理者権限を持つ Red Hat Single Sign-On アカウント。
- Red Hat Single Sign-On で作成した Satellite ユーザーアカウントのレルム。
- 証明書または CA が自己署名されている場合は、それらがエンドユーザー証明書トラストストアに追加されていることを確認する。
- ユーザーが Red Hat Single Sign-On にインポートまたは追加されている。
  LDAP や Kerberos などの既存のユーザーデータベースが設定されている場合は、ユーザーのフェデレーションを設定することでユーザーをインポートできます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の ユーザーストレージフェデレーション を参照してください。

  既存のユーザーデータベースが設定されていない場合は、Red Hat Single Sign-On でユーザーを手作業で作成できます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の 新規ユーザーの作成 を参照してください。

5.10.2. Satellite を Red Hat Single Sign-On クライアントとして登録する

以下の手順を使用して、Satellite をクライアントとして Red Hat Single Sign-On に登録し、認証ソースとして Red Hat Single Sign-On を使用するように Satellite を設定します。

Satellite と Red Hat Single Sign-On は、2 つの異なる認証方法で設定できます。

1. Satellite Web UI を使用した Satellite への認証。
2. Satellite CLI を使用した Satellite への認証。

どちらの方法でも、異なる Satellite クライアントを Red Hat Single Sign-On に登録して設定する必要があるため、ユーザーの認証方法を事前に決定する必要があります。この手順では、Red Hat Single Sign-On の Satellite クライアントの登録および設定方法が区別されています。

両認証方法を使用して、どちらのクライアントも適宜設定する場合には、Red Hat Single Sign-On に異なる Satellite クライアントを 2 つ登録することも可能です。

手順

1. Satellite Server で、以下のパッケージをインストールします。

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install python3-lxml
```

2. Satellite をクライアントとして Red Hat Single Sign-On に登録します。Web UI と CLI とでログインの登録プロセスが異なる点に注意してください。Red Hat Single Sign-On に 2 つの Satellite クライアントを登録すると、Web UI と CLI から Satellite にログインできます。
   - Web UI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

次に、認証ソースとして Red Hat Single Sign On を使用するように Satellite を設定します。

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

- CLI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name hammer-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

3. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

5.10.3. Red Hat Single Sign-On での Satellite クライアントの設定

以下の手順を使用して、Red Hat Single Sign-On Web UI で Satellite クライアントを設定し、Satellite クライアントのグループおよびオーディエンスマッパーを作成します。

手順

1. Red Hat Single Sign-On Web UI で、**クライアント** に移動し、Satellite クライアントをクリックします。
2. アクセスタイプを設定します。
   - Satellite web UI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **機密** を選択します。
   - CLI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **公開** を選択します。
3. **有効なリダイレクト URI** フィールドに有効なリダイレクト URI を追加します。
   - Satellite web UI を使用して Satellite への認証を行うには、**https://satellite.example.com/users/extlogin** の形式で URI を入力します。Satellite FQDN の後に **/users/extlogin** の文字列を追加する必要があります。
     この手順の完了後に、Satellite クライアントが Satellite web UI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri
https://satellite.example.com/users/extlogin
```

- CLI を使用してユーザーが Satellite への認証を行うには、既存の URI の下の空白フィールドに **urn:ietf:wg:wg:oauth:2.0:oob** を入力します。
  この手順の完了後に、Satellite クライアントが CLI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri
urn:ietf:wg:oauth:2.0:oob
```

4. **Save** をクリックします。
5. **マッパー** タブ、**作成** の順にクリックし、オーディエンスマッパーを追加します。
6. **名前** フィールドに、オーディエンスマッパーの名前を入力します。
7. **マッパータイプ** リストから、**オーディエンス** を選択します。
8. **組み込み済みクライアントオーディエンス** リストから、Satellite クライアントを選択します。
9. **Save** をクリックします。
10. **作成** をクリックして、グループメンバーシップをもとに Satellite の認証を指定できるようにグループマッパーを追加します。
11. **名前** フィールドにグループマッパーの名前を入力します。
12. **マッパータイプ** リストから、**グループメンバーシップ** を選択します。
13. **トークンクレーム名** に **groups** と入力します。
14. **フルグループパス** のトグルを OFF に設定します。
15. **Save** をクリックします。

5.10.4. Red Hat Single Sign-On 認証用の Satellite の設定

このセクションでは、Satellite Web UI または CLI を使用して Red Hat Single Sign-On 認証用に Satellite を設定します。

5.10.4.1. Web UI を使用した Red Hat Single Sign-On 認証用の Satellite の設定

以下の手順に従って、Satellite Web UI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration** の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **機密** に設定されていることを確認します。

手順

1. Satellite Web UI で、**管理** > **設定** に移動して、**認証** タブをクリックします。

2. **ログイン委任の認証** の行を見つけ、**値** コラムで **Yes** に値を設定します。
3. **Authorize login delegation auth source user autocreate**行を見つけ、**値** コラムで **External** に値を設定します。
4. **Login delegation logout URL** の行を見つけ、**Value** 列で、値を **https://satellite.example.com/users/extlogout** に設定します。
5. **OIDC アルゴリズム** の行を見つけ、**値** コラムで、Red Hat Single Sign On のエンコーティングのアルゴリズムを設定します (例: **RS256**)。
6. **OIDC オーディエンス** 行を見つけ、**値** コラムで、値を Red Hat Single Sign On のクライアント ID に設定します。
7. **OIDC Issuer** 行を見つけ、**Value** 列で、値を **https://RHSSO.example.com/auth/realms/Satellite_Realm** に設定します。
8. **OIDC JWKs URL** 行を見つけ、**Value** 列で、値を **https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs** に設定します。
9. Satellite Web UI で、**Administer** > **Authentication Sources** に移動し、**External** カードの垂直省略記号をクリックして、**Edit** を選択します。
10. **場所** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる場所を追加します。
11. **組織** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる組織を追加します。
12. **Submit** をクリックします。

5.10.4.2. CLI を使用した Red Hat Single Sign-On 認証用の Satellite の設定

以下の手順に従って、Satellite CLI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の **https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration** の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **公開** に設定されていることを確認します。

手順

1. Satellite で、ログイン委任を **true** に設定し、ユーザーが Open IDC プロトコルを使用して認証できるようにします。

```
# hammer settings set --name authorize_login_delegation --value true
```

2. ログイン委任のログアウト URL を以下のように設定します。

```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

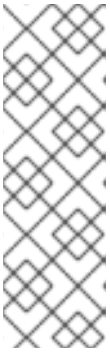
3. Red Hat Single Sign On のエンコーティングのアルゴリズムを設定します (例: **RS256**)。

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

4. URL **RHSSO.example.com/auth/realms/RHSSO_REALM/.well-known/openid-configuration** を開いて値をメモし、以下のステップのオプションに入力します。

5. Open IDC オーディエンスに Hammer クライアントの値を追加します。

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-hammer-openidc']"
```



注記

複数の Red Hat Single Sign-On クライアントを Satellite に登録する場合は、以下のように、アレイに全オーディエンスを必ず追加してください。以下に例を示します。

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-foreman-openidc', 'satellite.example.com-hammer-openidc']"
```

6. Open IDC 発行者の値を設定します。

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

7. Open IDC Java Web Token (JWT) の値を設定します。

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

8. Red Hat Single Sign-On 認証ソースの ID を取得します。

```
# hammer auth-source external list
```

9. ロケーションと組織を設定します。

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

5.10.5. Red Hat Single Sign-On による TOTP 認証用に Satellite を設定する

Time-based One-time Password (TOTP) を使用した外部認証用の OpenID プロバイダーとして Red Hat Single Sign-On を使用するように Satellite を設定するには、以下のセクションを使用します。

手順

1. Red Hat Single Sign-On Web UI で、Satellite レルムに移動します。

2. **Authentication** に移動し、**OTP Policy** タブをクリックします。
3. **サポートされるアプリケーション** フィールドに FreeOTP または Google Authenticator が含まれていることを確認します。
4. 要件に合わせて OTP を設定します。
5. 必要に応じて、すべてのユーザーのデフォルト認証方法として TOTP 認証を使用する場合は、**Flows** タブをクリックして **OTP Form** 設定の右側にある **REQUIRED** を選択します。
6. **Required Actions** タブをクリックします。
7. **Configure OTP** 行の右側にある **Default Action** チェックボックスを選択します。

5.10.6. Red Hat Single Sign-On TOTP 認証を使用した Satellite Web UI へのログイン

以下の手順に従って、Red Hat Single Sign-On TOTP 認証で Satellite Web UI にログインします。

手順

1. Satellite にログインすると、Satellite は Red Hat Single Sign-On のログイン画面にリダイレクトします。
2. ユーザー名とパスワードを入力し、**ログイン** をクリックします。
3. 初回ログインの場合には、Red Hat Single Sign-On により、バーコードをスキャンし、表示された暗証番号を入力してクライアントを設定するように求められます。
4. クライアントを設定して有効な暗証番号を入力すると、Red Hat Single Sign-On で Satellite にリダイレクト後にログインされます。

5.10.7. Red Hat Single Sign-On を使用した Satellite CLI へのログイン

以下の手順に従って、コード付与タイプを使用して Satellite CLI への認証を行います。

手順

1. コード付与タイプを使用して Satellite CLI への認証を行うには、以下のコマンドを入力します。

```
# hammer auth login oauth \
--two-factor \
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-connect/token' \
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \
--oidc-client-id 'satellite.example.com-foreman-openidc' \
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

   このコマンドは、サクセスコードの入力を要求します。

2. サクセスコードを取得するには、コマンドが返す URL に移動し、必要な情報を提供します。
3. Web UI が返すサクセスコードをコピーします。

4. **hammer auth login oauth** のコマンドプロンプトに、サクセスコードを入力して Satellite CLI に対して認証を行います。

5.10.8. Red Hat Single Sign-On 認証用のグループマッピングの設定

必要に応じて、ロールベースのアクセス制御 (RBAC) を実装するには、Satellite でグループを作成し、このグループにロールを割り当ててから Active Directory グループを Satellite グループにマッピングします。これにより、Red Hat Single Sign-On の指定のグループに所属する場合には、該当する Satellite グループでログインします。この例では、Active Directory の Satellite-admin ユーザーグループのユーザーを設定し、Satellite で管理者権限を持つユーザーとして認証されるようにします。

手順

1. Satellite Web UI で、**Administer** > **User Groups** に移動します。
2. **Create User Group** をクリックします。
3. **名前** フィールドにユーザーグループの名前を入力します。名前は Active Directory と同じにしないでください。
4. 右側の列には、ユーザーおよびユーザーグループを追加しないでください。**ロール** タブをクリックします。
5. **管理** チェックボックスを選択します。
6. **外部グループ** タブをクリックします。
7. **外部ユーザーグループの追加** をクリックします。
8. **名前** フィールドに、Active Directory の名前を入力します。
9. リストから **外部** を選択します。

5.11. RED HAT SINGLE SIGN-ON 認証の無効化

Satellite で Red Hat Single Sign On 認証を無効化するには、以下の手順を実行します。

手順

- Red Hat Single Sign On 認証を無効化するには、以下のコマンドを入力します。

```
# satellite-installer --reset-foreman-keycloak
```

第6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。

6.1. 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。

6.2. 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」

6.2.1. 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 設定とリースフィアルを 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
```

6.2.2. 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。

6.3. 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:tftpdir_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。

6.4. 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。レルム登録の設定の詳細は、「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。

6.4.1. 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。

6.4.2. 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る。
- 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
```

7. 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

1. 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==";
};
```

2. 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
```

3. 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
```

4. 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。

5. 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
```

6. 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。

6.4.3. 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 DNF モジュールのトラブルシューティング

DNF モジュールを有効にできない場合は、間違ったモジュールが有効になっている可能性があります。その場合は、次のように依存関係を手動で解決する必要があります。有効なモジュールをリストします。

```
# dnf module list --enabled
```

A.1. RUBY

Ruby モジュールを有効にできない場合は、間違ったモジュールが有効になっている可能性があります。その場合は、次のように依存関係を手動で解決する必要があります。

有効なモジュールをリストします。

```
# dnf module list --enabled
```

Ruby 2.5 モジュールがすでに有効になっている場合は、モジュールのリセットを実行します。

```
# dnf module reset ruby
```

A.2. POSTGRESQL

PostgreSQL モジュールを有効にできない場合は、間違ったモジュールが有効になっている可能性があります。その場合は、次のように依存関係を手動で解決する必要があります。

有効なモジュールをリストします。

```
# dnf module list --enabled
```

PostgreSQL 10 モジュールがすでに有効になっている場合は、モジュールのリセットを実行します。

```
# dnf module reset postgresql
```

データベースが以前に PostgreSQL 10 を使用して作成されていた場合は、アップグレードを実行します。

1. DNF モジュールを有効にします。

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

2. PostgreSQL アップグレードパッケージをインストールします。

   ```
   # dnf install postgresql-upgrade
   ```

3. アップグレードを実行します。

   ```
   # postgresql-setup --upgrade
   ```

付録B 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、以下のようにします。

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

- Apache サーバー署名をオフにするには、以下のようにします。

```
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C 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。