Red Hat Satellite 6.2

インストールガイド

Red Hat Satellite Server および Capsule Server のインストール

Last Updated: 2018-05-04

Red Hat Satellite 6.2 インストールガイド

Red Hat Satellite Server および Capsule Server のインストール

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

法律上の通知

概要

本ガイドでは、Red Hat Satellite Server および Capsule 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について説明します。

目次

第1章 SATELLITE SERVER と CAPSULE SERVER の機能

Red Hat Satellite は、物理環境、仮想環境、およびクラウド環境でシステムをデプロイ、設定、および保守することを可能にするシステム管理ソリューションです。Satellite は、一元化された単一のツールにより、プロビジョニング、リモート管理、および複数の Red Hat Enterprise Linux デプロイメントの監視を提供します。Red Hat Satellite Server は、Red Hat カスタマーポータルからのコンテンツを同期し、詳細なライフサイクル管理、ユーザーおよびグループロールベースアクセス制御、統合サブスクリプション管理、高度な GUI、CLI、および API アクセスを含む機能を提供します。

Red Hat Satellite Capsule Server は、さまざまな地理的な場所でのコンテンツフェデレーションを実現するために Red Hat Satellite Server からのコンテンツをミラーリングします。ホストシステムは中央 Satellite Server からではなく Capsule Server からコンテンツをプルできます。また、Capsule Server は Puppet Master、DHCP、DNS、TFTP などのローカライズされたサービスも提供します。Capsule Server を使用すると、管理対象システムの数が増えたときに Satellite 環境を簡単にスケーリングできます。

Capsule Server により中央サーバーの負荷が減少し、冗長性が増加し、帯域幅の使用率が低下します。詳細については、『**Red Hat Satellite Architecture Guide**』を参照してください。

第2章 インストールのために環境を準備

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

注記

Red Hat Satellite Server と Capsule Server のバージョンは一致する必要があります。たとえば、Satellite 6.1 Server は 6.2 Capsule Server 実行をできず、Satellite 6.2 Server は 6.1 Capsule Server を実行できません。Satellite Server と Capsule Server のバージョンが一致しないと、Capsule Server が警告なしで失敗します。

大量のコンテンツホストがある場合は、Large Deployment Considerations を参照して、環境が適切にセットアップされていることを確認します。

Capsule Server のスケーリングの詳細については、「Capsule Server のスケーラビリティーに関する考慮事項」を参照してください。

2.1. ストレージの要件と推奨事項

Satellite Server または Capsule Server をインストールする前に環境が最小要件を満たしていることを確認します。

異なるリポジトリーで重複するパッケージは、ディスク上で 1 回だけ格納されます。重複するパッケージを含む追加リポジトリーに必要な追加ストレージは少なくなります。大量のストレージは、**/var/lib/mongodb/** ディレクトリーおよび **/var/lib/pulp/** ディレクトリーに存在します。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防ぐ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。

/var/cache/pulp/ ディレクトリーは、同期中にコンテンツを一時的に保管するために使用されます。RPM 形式のコンテンツの場合、このディレクトリーには任意のときに最大 5 RPM ファイルが保管されます。各ファイルは、同期後に **/var/lib/pulp/** ディレクトリーに移動されます。デフォルトでは、最大 8 個の RPM コンテンツ同期タスクを同時に実行できます (それぞれでは最大 1 GB のメタデータが使用されます)。ISO 形式のコンテンツの場合、同期タスクあたりのすべての ISO ファイルは、タスクが完了するまで **/var/cache/pulp/** に格納されます (タスクの完了後は、**/var/lib/pulp/** ディレクトリーに移動されます)。たとえば、4 つの ISO ファイル (それぞれのサイズが 4 GB) を同期している場合は、**/var/cache/pulp/** ディレクトリーで合計 16 GB が必要です。同期する ISO ファイルの数を考慮してください (これらのファイルに必要な一時ディスク容量は通常 RPM コンテンツのサイズを超えます)。

/var/lib/qpidd/ ディレクトリーは、コンテンツホストあたり 2 MB を少し超える容量を使用します。たとえば、10 000 のコンテンツホストの場合、**/var/lib/qpidd/** に 20 GB のディスク容量が必要になります。

ストレージ要件

以下の表には、特定のディレクトリーの推奨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、期待されるユースケースシナリオに基づき、個別の環境に応じて異なることがあります。この表は Satellite Server と各 Capsule Server に適用されます。

表2.1 インストール向け最小ストレージ要件

フォルダー	インストールサイズ	同期された Red Hat Enterprise Linux 5、6、および 7 の実行時サイズ	留意事項
/var/cache/pulp/	1M バイト	10 GB (最小)	本項の概要にある記述を参照してください。
/var/lib/pulp/	1 MB	500 GB	• コンテンツが Satellite Server に追加されるときに継続して増加します。長期間の増加を計画してください。 • シンボリックリンクは使用できません。
/var/lib/mongodb/	3.5 GB	50 GB	• コンテンツが Satellite Server に追加されるときに継続して増加します。長期間の増加を計画してください。 • シンボリックリンクは使用できません。 • MongoDB では NFS は推奨されません。
/var/log/	10 MB	250 MB	なし
/var/lib/pgsql/	100 MB	10 GB	**`/var/lib/pgsql/`** に最小 2 GB の利用可能なストレージがあること。さらに、データストレージ要件の増加に伴ってこのディレクトリーを含むパーティションを拡張できること。PostgreSQL で NFS を使用することは推奨されません。
/usr	3 GB	適用外	なし

フォルダー	インストールサイズ	同期された Red Hat Enterprise Linux 5、6、および 7 の実行時サイズ	留意事項
/opt	500 MB (接続されたインストール)	適用外	ソフトウェアコレクションは、**/opt/rh/** ディレクトリーと **/opt/theforeman/** ディレクトリーにインストールされます。**/opt** ディレクトリーへのインストールには、root による書き込みおよび実行パーミッションが必要です。
/opt	2 GB (切断されたインストール)	適用外	• ソフトウェアコレクションは、**/opt/rh/** ディレクトリーと **/opt/theforeman/** ディレクトリーにインストールされます。**/opt** ディレクトリーへのインストールには、root による書き込みおよび実行パーミッションが必要です。 • インストールに使用されるリポジトリーのコピーは、このディレクトリーに格納されます。

ストレージの推奨事項

- ほとんどの Satellite Server データは **/var** ディレクトリーに格納されるため、システムがスケーラブルになるよう **/var** を LVM ストレージにマウントすることを強くお勧めします。
- Red Hat は、**/var/lib/pulp/** ディレクトリーと **/var/lib/mongodb/** ディレクトリーに高帯域幅で低レイテンシーのストレージの使用をお勧めします。Red Hat Satellite には I/O を大量に使用する多くの操作があるため、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することがあります。MongoDB はデータファイルにアクセスするために通常の I/O を使用せず、データファイルとジャーナルファイルが NFS でホストされ

た場合にパフォーマンスの問題が発生するため、MongoDB とともに NFS を使用することは推奨されません。NFS を使用する必要がある場合は、**/etc/fstab** ファイルで **bg**、**nolock**、および **noatime** のオプションを使用してボリュームをマウントします。

- 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。
- パフォーマンスを向上させるには、HDD (Hard Disk Drive) ではなく SSD (Solid State Drive) を使用します。
- XFS ファイルシステムは、**ext4** では存在する inode の制限がないため、Red Hat Satellite 6 に推奨されます。Satellite はたくさんのシンボリックリンクを使用するため、**ext4** とデフォルトの数の inode を使用する場合は、システムで inode が足りなくなる可能性が高くなります。Red Hat Enterprise Linux 6 を代わりに使用する場合は、このシステムで XFS を有効にすることについてアカウントチームにご連絡ください。また、Red Hat Enterprise Linux 6 上の Satellite 6 の長期サポートのライフスパンが短いため、将来バージョン 6 から 7 への移行が必要になることがあります。新しいインストールには Red Hat Enterprise Linux 7 が強く推奨されます。

2.2. 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 がサポートする他の任意の方法でインストールできます。Red Hat Satellite Server と Red Hat Satellite Capsule Server は、Satellite 6.2 がリリースされたときに利用可能な Red Hat Enterprise Linux 6 Server または 7 Server の最新バージョンでのみサポートされます。EUS または z-stream を含む Red Hat Enterprise Linux の以前のバージョンはサポートされません。

Red Hat Satellite Server および Red Hat Satellite Capsule 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、サーバーの直接的な運用に直接必要でないサードパーティーの構成やソフトウェアは含めないようにしてください。この制限には、機能強化や Red Hat 以外のセキュリティーソフトウェアが含まれます。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、完全に機能する Satellite Server 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。

Satellite Server は新しくプロビジョニングされたシステムであることが推奨されます。Satellite を実行する以外の目的でのシステムの使用はサポートされません。

以下のいずれかがシステムに存在する場合は、インストールする前にそれらを削除する必要があります。

- Java 仮想マシン
- Puppet RPM ファイル
- 本書でインストールのために明示的に必要とされた以外の追加の yum リポジトリー

2.3. ハードウェア要件

ネットワーク接続されたベースシステムには、以下の要件が適用されます。

- 64 ビットアーキテクチャー
- Red Hat Enterprise Linux 6 Server または 7 Server の最新バージョン
- 最低 2 CPU コア (4 CPU コアを推奨)
- Satellite Server が機能するには、最低 12 GB のメモリーが必要です。Satellite Server の各イン

スタンスには 16 GB 以上のメモリーが推奨されます。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されている 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順方向および逆方向の DNS 解決。

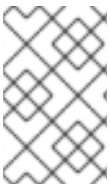
2.4. サポート対象ブラウザー

以下の Web ブラウザーは完全にサポートされます。

- Firefox バージョン 35 以降
- Chrome バージョン 28 以降

以下の Web ブラウザーは部分的にサポートされます。Satellite Web UI インターフェースは正常に機能しますが、特定のデザイン要素が期待どおりに表示されないことがあります。

- Firefox バージョン 38
- Chrome バージョン 27
- Internet Explorer バージョン 10 および 11



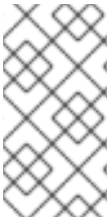
注記

Satellite Server の Web UI とコマンドラインインターフェースは、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。

2.5. 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が通信できるようにするには、特定のネットワークポートがベースオペレーティングシステムでオープンかつフリーの状態であり、ネットワークベースファイアウォールでオープンである必要があります。本項の表は、ポートの用途を説明しています。ホストベースのファイアウォール向けの対応するファイアウォールコマンドは、以下の項に記載されています。インストールが開始される前に、Satellite Server と Capsule Server 間のポートがオープンされない場合は、Capsule Server のインストールに失敗します。

以下の表は、ネットワークトラフィックの宛先ポートと方向を示しています。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ます。一部のクラウドソリューションでは、ネットワークベースのファイアウォールと同様にそれぞれのマシンが分断されるため、マシン間の通信を特別に許可するよう設定する必要があることに注意してください。



注記

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、以下の表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システムが含まれます。ネットワークベースのファイアウォール設定を計画している場合は、このことを考慮してください。

Capsule のクライアントであるシステム (内部 Capsule 以外) は、Satellite Server へのアクセスを必要としません。Satellite の詳細については、『Red Hat Satellite 6.2 Architecture Guide』の項「Capsule Networking」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。

表2.2 Red Hat CDN 通信に対する Satellite のポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	サブスクリプション管理サービス (access.redhat.com) と Red Hat CDN (cdn.redhat.com) への接続。

切断された Satellite のケースを除き、Satellite Server は Red Hat CDN へのアクセスを必要とします。

表2.3 Satellite へのブラウザーベースユーザーインターフェース向けポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Satellite へのブラウザーベース UI アクセス
80	TCP	HTTP	Satellite に Web UI アクセスするための HTTPS へのリダイレクション (オプション)

表2.4 Satellite に通信するクライアント向けポート

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Anaconda、yum、Katello 証明書およびテンプレートの取得向け、iPXE ファームウェアのダウンロード向け
443	TCP	HTTPS	サブスクリプション管理サービス、yum、Telemetry サービス、Katello エージェントへの接続向け
5647	TCP	amqp	Satellite の Qpid ディスパッチルータと通信する Katello エージェント
8000	TCP	HTTPS	キックスタートテンプレートをホストにダウンロードする Anaconda、iPXE ファームウェアのダウンロード向け
8140	TCP	HTTPS	マスター接続に対する Puppet エージェント

ポート	プロトコル	サービス	用途
9090	TCP	HTTPS	統合 Capsule のスマートプロキシーへの SCAP レポートの送信、プロビジョニング中の検出イメージ向け
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続

Satellite Server に直接接続された管理対象ホストは、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 が実行されているベースシステムが含まれます。

表2.5 Capsule に通信するクライアント向けポート

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Anaconda、yum、および Katello 証明書アップデートの取得向け
443	TCP	HTTPS	Anaconda、yum、Telemetry サービス、および Puppet
5647	TCP	amqp	Capsule の Qpid ディスパッチルータと通信する Katello エージェント
8000	TCP	HTTPS	キックスタートテンプレートをホストにダウンロードする Anaconda、iPXE ファームウェアのダウンロード向け
8140	TCP	HTTPS	マスター接続に対する Puppet エージェント
8443	TCP	HTTPS	サブスクリプション管理サービスおよび Telemetry サービス
9090	TCP	HTTPS	Capsule のスマートプロキシーへの SCAP レポートの送信、プロビジョニング中の検出イメージ向け
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続

表2.6 Satellite に通信する Capsule 向けポート

ポート	プロトコル	サービス	用途

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Katello、Foreman、Foreman API、および Pulp
5646	TCP	amqp	Capsule の Qpid ディスパッチルーターから Satellite の Qpid ディスパッチルーターへの通信
5647	TCP	amqp	Satellite の Qpid ディスパッチルーターと通信する Katello エージェント
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続

Capsule Server が実行されているベースシステムは Satellite Server に接続されたクライアントであることに注意してください。表「Satellite に通信するクライアント向けポート」を参照してください。

表2.7 Capsule に通信する Satellite 向けポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Capsule の Pulp サーバーへの接続
9090	TCP	HTTPS	Capsule のプロキシーへの接続
80	TCP	HTTP	bootdisk のダウンロード (オプション)

表2.8 オプションのネットワークポート

ポート	プロトコル	サービス	用途
53	TCP および UDP	DNS	Capsule の DNS サービスに Capsule DNS を問い合わせるクライアント
67	UDP	DHCP	Capsule ブロードキャストと、Capsule からプロビジョニングするクライアントに対する DHCP ブロードキャストを行うクライアント
68	UDP	DHCP	クライアントブロードキャストと、Capsule からプロビジョニングするクライアントに対する DHCP ブロードキャストを行うクライアント

ポート	プロトコル	サービス	用途
69	UDP	TFTP	プロビジョニングのために Capsule から PXE ブートイメージファイルをダウンロードするクライアント
8443	TCP	HTTP	プロビジョニング中に検出済みホストに送信する Capsule から Client への "reboot" コマンド
7911	TCP	DHCP	• DHCP レコードのオーケストレーションのための実行元が Capsule のコマンド (ローカルまたは外部) • DHCP が外部サービスにより提供された場合は、外部サーバーでポートを開く必要があります。
5000	TCP	HTTP	OpenStack のコンピュートリソースまたは実行中の Docker コンテナーに対する Satellite による通信
22, 16514	TCP	SSH、SSL/TLS	libvirt のコンピュートリソースに対する Satellite による通信
389、636	TCP	LDAP、LDAPS	LDAP およびセキュアな LDAP 認証ソースに対する Satellite による通信
5900～5930	TCP	SSL/TLS	ハイパーバイザー向け Web UI の NoVNC コンソールに対する Satellite による通信

2.6. クライアントから SATELLITE SERVER への接続を有効化

Satellite Server の内部 Capsule のクライアントであるシステムには、ホストおよび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が必要です。本項では、クライアントからの受信接続を有効にし、これらのルールをシステムの再起動後にも保持するために Satellite Server のベースシステムでホストベースファイアウォールを設定することについて説明します。使用されたポートの詳細については、「ポートとファイアウォールの要件」を参照してください。

Red Hat Enterprise Linux 6 でのファイアウォールの設定

1. Client と Satellite 間の通信に必要なポートを開きます。

```
# iptables -I INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
\
&& iptables -I INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 67 -j
```

```
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 69 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5647 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 -j
ACCEPT \
&& service iptables save
```

2. iptables サービスが起動され、有効であることを確認します。

```
# service iptables start
# chkconfig iptables on
```

Red Hat Enterprise Linux 7 でのファイアウォールの設定

1. Client と Satellite 間の通信に必要なポートを開きます。

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
 --add-port="67/udp" --add-port="69/udp" \
 --add-port="80/tcp"  --add-port="443/tcp" \
 --add-port="5647/tcp" \
 --add-port="8000/tcp" --add-port="8140/tcp"
```

2. **`--permanent`** オプションを追加してコマンドを繰り返し、設定を永続化します。

```
# firewall-cmd --permanent --add-port="53/udp" --add-port="53/tcp" \
 --add-port="67/udp" --add-port="69/udp" \
 --add-port="80/tcp"  --add-port="443/tcp" \
 --add-port="5647/tcp" \
 --add-port="8000/tcp" --add-port="8140/tcp"
```

2.7. CAPSULE SERVER から SATELLITE SERVER への接続を有効化

Capsule Server から Satellite Server への受信接続を有効にし、これらのルールを再起動後に保持するには、以下の手順に従います。外部の Capsule Server を使用しない場合は、この接続を有効にする必要はありません。

前提条件

Capsule Server のベースシステムは、Satellite Server のクライアントであるため、クライアントから Satellite Server への接続を有効化の手順を最初に完了する必要があります。この手順により、外部の Capsule Server が必要とする追加のポートが開きます。

使用されるポートの詳細については、「ポートとファイアウォールの要件」を参照してください。

Red Hat Enterprise Linux 6 でのファイアウォールの設定

1. iptables サービスを設定します。

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 5646 -j
ACCEPT \
&& service iptables save
```

2. iptables サービスを起動します。

```
# service iptables restart
# chkconfig iptables on
```

Red Hat Enterprise Linux 7 でのファイアウォールの設定

1. Satellite Server でファイアウォールを設定します。

```
# firewall-cmd --add-port="5646/tcp"
```

2. **--permanent** オプションを追加してコマンドを繰り返し、設定を永続化します。

```
# firewall-cmd --permanent --add-port="5646/tcp"
```

2.8. SATELLITE SERVER およびクライアントからの CAPSULE SERVER への接続を有効化

Satellite Server およびクライアントから Capsule Server への受信接続を有効にし、再起動後にこれらのルールが保持されるようにすることができます。外部の Capsule Server を使用しない場合は、この接続を有効にする必要はありません。

使用されるポートの詳細については、「ポートとファイアウォールの要件」を参照してください。

Red Hat Enterprise Linux 6 でのファイアウォールの設定

1. iptables サービスを設定します。

```
# iptables -I INPUT -m state --state NEW -p udp --dport 53 -j ACCEPT
\
&& iptables -I INPUT -m state --state NEW -p tcp --dport 53 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 67 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p udp --dport 69 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 80 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 443 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 5647 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8000 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8140 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 8443 -j
ACCEPT \
```

```
&& iptables -I INPUT -m state --state NEW -p tcp --dport 9090 -j
ACCEPT \
&& service iptables save
```

2. iptables サービスを起動します。

```
# service iptables restart
# chkconfig iptables on
```

Red Hat Enterprise Linux 7 でのファイアウォールの設定

1. Capsule Server でファイアウォールを設定します。

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
 --add-port="67/udp" \
 --add-port="69/udp" --add-port="80/tcp" \
 --add-port="443/tcp" --add-port="5647/tcp" \
 --add-port="8000/tcp" --add-port="8140/tcp" \
 --add-port="8443/tcp" --add-port="9090/tcp"
```

2. **--permanent** オプションを追加してコマンドを繰り返し、設定を永続化します。

```
# firewall-cmd --permanent --add-port="53/udp" --add-port="53/tcp" \
 --add-port="67/udp" \
 --add-port="69/udp" --add-port="80/tcp" \
 --add-port="443/tcp" --add-port="5647/tcp" \
 --add-port="8000/tcp" --add-port="8140/tcp" \
 --add-port="8443/tcp" --add-port="9090/tcp"
```

2.9. 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。

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
```

警告

Satellite 6 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、多くのオプションは失敗します。これらのオプションには、コンテンツ管理、サブスクリプション管理、およびプロビジョニングがあります。

2.10. デフォルトの SELINUX ポートの変更

Red Hat Satellite 6 では、事前定義されたポートセットが使用されます。Red Hat は、Satellite 6 システムの SELinux を Permissive または Enforcing に設定することを推奨します。いずれかのサービスのポートを変更する必要がある場合は、関連する SELinux ポートタイプを変更して、リソースへのアクセスを許可する必要があります。これらのポートは、標準以外のポートを使用する場合のみ、変更する必要があります。

たとえば、Satellite Web UI ポート (HTTP/HTTPS) を 8018/8019 に変更する場合は、これらのポート番号を httpd_port_t SELinux ポートタイプに追加する必要があります。

この変更は、ターゲットポートにも必要です (たとえば、Satellite 6 が Red Hat Virtualization や Red Hat OpenStack Platform などの外部ソースに接続する場合)。

デフォルトのポート割り当てには 1 度だけ変更を加える必要があります。Satellite をアップデートまたはアップグレードしても、これらの割り当てには影響ありません。割り当てが存在しない場合、アップグレードすると、デフォルトの SELinux ポートのみが追加されます。

作業を開始する前に

- Satellite をインストールする前に、SELinux を有効にし、Permissive または Enforcing モードで実行する必要があります。詳細については、『**Red Hat Enterprise 6 Security-Enhanced Linux User Guide**』または『**Red Hat Enterprise Linux 7 SELinux User's and Administrator's Guide**』を参照してください。

デフォルトのポートをユーザー指定のポートに変更

1. ポートをデフォルトのポートからユーザー指定のポートに変更するには、使用している環境に関連する値を使用してコマンドを実行します。以下の例では、デモのためにポート 99999 を使用しています。

デフォルトのポート	SELinux コマンド
80, 443, 8443	semanage port -a -t http_port_t -p tcp 99999
8080	semanage port -a -t http_cache_port_t -p tcp 99999

デフォルトのポート	SELinux コマンド
8140	semanage port -a -t puppet_port_t -p tcp 99999
9090	semanage port -a -t websm_port_t -p tcp 99999
69	semanage port -a -t tftp_port_t -p udp 99999
53 (TCP)	semanage port -a -t dns_port_t -p tcp 99999
53 (UDP)	semanage port -a -t dns_port_t -p udp 99999
67, 68	semanage port -a -t dhcpd_port_t -p udp 99999
5671	semanage port -a -t amqp_port_t -p tcp 99999
8000	semanage port -a -t soundd_port_t -p tcp 99999
7911	semanage port -a -t dhcpd_port_t -p tcp 99999
5000 (Red Hat Enterprise Linux 6 の場合)	semanage port -a -t commplex_port_t -p tcp 99999
5000 (Red Hat Enterprise Linux 7 の場合)	semanage port -a -t commplex_main_port_t -p tcp 99999
22	semanage port -a -t ssh_port_t -p tcp 99999
16514 (libvirt)	semanage port -a -t virt_port_t -p tcp 99999
389、636	semanage port -a -t ldap_port_t -p tcp 99999
5910～5930	semanage port -a -t vnc_port_t -p tcp 99999

2. 以前に使用されたポート番号とポートタイプの関連付けを解除します。

```
# semanage port -d -t virt_port_t -p tcp 99999
```

第3章 SATELLITE SERVER のインストール

Satellite Server のインストールには、接続と切断の 2 つの方法があります。接続されたインストールでは、Satellite Server をインストールするのに必要なパッケージを Red Hat Content Delivery Network (CDN) から直接インストールして取得できます。切断されたインストールでは、外部のコンピューターからパッケージの ISO イメージをダウンロードし、インストールのために Satellite Server にコピーします。

ネットワークに接続されているホストの場合、Red Hat は CDN から直接パッケージをインストールすることをお勧めします。ISO イメージには最新のアップデートが含まれないことがあるため、ISO イメージの使用は、切断された環境のホストにのみ推奨されます。

Satellite Server を正常にインストールするには、root アクセスが必要です。

3.1. 切断されたネットワークからの SATELLITE SERVER のインストール

切断されたネットワークから Satellite Server をインストールすると、Red Hat Content Delivery Network から直接パッケージとアップデートを取得できます。

Satellite 6 インストールプログラムは Puppet に基づいていることに注意してください。つまり、インストールプログラムを複数回実行すると、手動での設定の変更が上書きされることがあります。この問題を回避する場合は、インストールプログラムを実行するときに **--noop** 引数を使用して、適用する変更を特定します。この引数により、実際の変更は行われません。潜在的な変更は **/var/log/katello-installer.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、不要な変更は復元することができます。たとえば、katello-installer ログには、Filebucket に関する以下のようなエントリーが示されます。

```
/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
```

3.1.1. Red Hat Subscription Management 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、ホストはユーザーが利用可能なサブスクリプションに対するコンテンツをサブスクライブし、消費できます。これには、Red Hat Enterprise Linux、Red Hat Software Collection (RHSCL)、Red Hat Satellite などのコンテンツが含まれます。

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
```

3.1.2. Satellite サブスクリプションを識別してホストに割り当てる

ホストの登録後に、利用可能な Satellite サブスクリプションを識別し、割り当てる必要があります。Satellite サブスクリプションは、Satellite コンテンツ、Red Hat Enterprise Linux、Red Hat Software Collections (RHSCL)、および Red Hat Satellite へのアクセスを提供します。これは、必要な唯一のサブスクリプションです。各 Red Hat サブスクリプションはプール ID によって識別されます。

1. Satellite サブスクリプションの識別
   Red Hat Enterprise Linux 6.7 (以上) または 7.1 (以上) では、文字列 **`Red Hat Satellite`** を含む利用可能なすべてのサブスクリプションを検索できます。Red Hat Enterprise Linux の古いバージョンでは、**すべての**利用可能なサブスクリプションをリストし、適切なサブスクリプションの出力を手動でチェックする必要があります。

   a. Red Hat Enterprise Linux 6.7 (以上) または 7.1 (以上) では、以下のコマンドを実行します。

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

   このコマンドは、利用可能なすべてのサブスクリプションのフィールドに対して小文字と大文字を区別しない検索を実行します (**`Subscription Name`** と **`Provides`** を含み、**`Red Hat Satellite`** のすべてのインスタンスに一致)。サブスクリプションは、システムにすでに割り当てられていない場合に利用可能として分類されます。検索文字列には、単一の文字またはゼロ以上の文字にそれぞれ一致するワイルドカード **`?`** または **`*`** を含めることもできます。ワイルドカード文字はバックスラッシュでエスケープして、リテラルの疑問符またはアスタリスクを表すことができます。

   利用可能な Satellite サブスクリプションを見つけることができない場合は、Red Hat ナレッジベースソリューション How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager? 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によって消費されているかどうかを確認できます。

   b. Red Hat Enterprise Linux の他のバージョンで、以下のコマンドを実行します。

   ```
   # subscription-manager list --all --available
   ```

   出力が長すぎる場合は、**`less`** や **`more`** などのページャーユーティリティーにパイプして、一度に 1 画面ずつ出力を確認できるようにします。

   c. 実行される **`subscription-manager`** コマンドの形式に関係なく、出力は以下のようになります。

   ```
   Subscription Name: Red Hat Satellite
   Provides:          Oracle Java (for RHEL Server)
                      Red Hat Satellite 6
                      Red Hat Enterprise Linux Server
                      Red Hat Satellite
                      Red Hat Enterprise Linux Load Balancer (for
   RHEL Server)
   SKU:               MCT0370
   Pool ID:           8a85f9874152663c0541943739717d11
   ```

```
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2014
System Type:       Physical
```

2. Satellite ホストに割り当てることができるように、プール ID をメモします。使用するプール ID は、ここで提示した例とは異なります。

3. Satellite サーバーにサブスクリプションを割り当てるには、お使いのプール ID を指定して以下のコマンドを実行します。

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

出力は以下のようになります。

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

4. サブスクリプションが正しく割り当てられたことを確認するには、以下のコマンドを実行します。

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

この出力では、以下のような内容が表示されます。

```
+-------------------------------------------+
   Consumed Subscriptions
+-------------------------------------------+
Subscription Name: Red Hat Satellite
Provides:          Red Hat Satellite
                   Red Hat Enterprise Linux Server
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Satellite
                   Red Hat Satellite 6
                   Red Hat Software Collections  (for RHEL Server)
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
                   Red Hat Satellite with Embedded Oracle
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux High Availability (for
RHEL Server)
SKU:               MCT0370
Contract:          10293569
Account:           5361051
Serial:            1653856191250699363
Pool ID:           8a85f9874152663c0541943739717d11
Active:            True
Quantity Used:     1
Service Level:     Premium
Service Type:      L1-L3
Status Details:
```

```
Starts:            10/08/2013
Ends:              10/07/2014
System Type:       Physical
```

3.1.3. リポジトリーの設定

1. すべての既存のリポジトリーを無効にします。

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

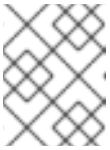
2. Red Hat Satellite および Red Hat Enterprise Linux、Red Hat Software Collections リポジトリーを有効にします。
Red Hat Enterprise Linux リポジトリーが、使用してる特定のバージョンに一致することを確認します。

   a. Red Hat Enterprise Linux 6 を使用している場合は、次のコマンドを実行します。

```
# subscription-manager repos --enable=rhel-6-server-rpms \
--enable=rhel-server-rhscl-6-rpms \
--enable=rhel-6-server-satellite-6.2-rpms
```

   b. Red Hat Enterprise Linux 7 を使用している場合は、次のコマンドを実行します。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-6.2-rpms
```



注記

Red Hat Enterprise Linux の異なるバージョンを使用する場合は、お使いのバージョンに基づいてリポジトリーを変更してください。

3. Red Hat 以外の yum リポジトリーから残されたすべてのメタデータを消去します。

```
# yum clean all
```

4. リポジトリーが有効になっていることを確認します。

```
# yum repolist enabled
```

以下のような出力が表示されます。

```
Loaded plugins: product-id, subscription-manager
repo id                                          repo name
status
!rhel-7-server-rpms/x86_64                       Red Hat Enterprise
Linux 7 Server (RPMs)                                     9,889
!rhel-7-server-satellite-6.2-rpms/x86_64    Red Hat Satellite 6.2
(for RHEL 7 Server) (RPMs)                            545
!rhel-server-rhscl-7-rpms/x86_64                 Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server     4,279
repolist: 14,713
```

3.1.4. Satellite サーバーパッケージのインストール

Satellite サーバーパッケージをインストールする前に、すべてのパッケージを更新する必要があります。インストール後に、サーバー証明書の設定、ユーザー名、パスワード、デフォルトの組織および場所の設定を含む Satellite サーバーの初期設定を実行する必要があります。

1. すべてのパッケージを更新します。

```
# yum update
```

2. インストールパッケージをインストールします。

```
# yum install satellite
```

3. 初期設定の実行に移動して、インストーラープログラムを実行し、Satellite サーバーの初期設定を行います。

3.1.5. マニフェストの作成

Satellite サーバーのカスタマーポータルページでは、サブスクリプションのグループを収集し、管理対象システムに配布するためにサーバーに割り当てます。これを行うには、Satellite サーバー向けのサブスクリプションマニフェストを作成します。マニフェストを作成するには、以下の手順を実行します。

1. Red Hat カスタマーポータルに移動し、ログインします。
2. **サブスクリプション** をクリックします。
3. **Red Hat サブスクリプション管理** セクションで **Satellite Organizations** をクリックします。
4. **Satellite の登録** をクリックします。
5. **名前** フィールドに、**Satellite_Server_example** と入力します。
6. バージョンに **Satellite 6.2** を選択して **登録** をクリックします。
7. **サブスクリプションのアタッチ** をクリックします。
8. 割り当てるサブスクリプションのチェックボックスをそれぞれ選択して、サブスクリプション数を指定します。
9. **選択項目のアタッチ** をクリックします。
   すべてのサブスクリプションが割り当てられるまで数分かかることがあります。
10. **マニフェストのダウンロード** をクリックして、マニフェストファイルを所定の場所に保存します。

3.1.6. マニフェストの Satellite サーバーへのアップロード

Red Hat Satellite 6 Web UI と CLI は、マニフェストをインポートする手段を提供します。

Web UI を使用したマニフェストのアップロード

1. 正しい組織内にいることを確認します。
2. **コンテンツ** > **Red Hat サブスクリプション** をクリックします。

3. **マニフェストの管理** をクリックしてサブスクリプションページを開きます。
4. **参照** をクリックして、作成したマニフェストファイルを選択し、**開く** をクリックします。
5. **アップロード** をクリックして、マニフェストを Satellite サーバーにアップロードします。

Hammer CLI を使用したマニフェストのアップロード

1. Satellite サーバーにマニフェストをアップロードします。

```
hammer subscription upload --organization-label org_label --file
path_to_manifest
```

3.1.7. 自己登録 Satellite の設定

Red Hat Satellite サーバーは通常 Red Hat カスタマーポータルに登録され、Satellite サーバーとしてアクティベートされ、Red Hat カスタマーポータルから新しいコンテンツを取得します。自己登録 Red Hat Satellite 6 サーバーは、Red Hat カスタマーポータルではなくそれ自体に登録されます。

Red Hat Satellite 6 サーバーがインストールされた場合、クライアントとして自己登録するには複数の利点があります。

- 同じライフサイクル管理の手順は、残りの管理対象に適用された Satellite 6 サーバー自体に適用できます。
- Satellite 6 サーバーを独自のコンテンツビューにサブスクライブすることにより、残りの管理対象ホストと同じスケジュールで同じアップデートを受け取ります。
- 自己登録 Satellite Server を使用して virt-who を使用できますが、自己登録なしで virt-who をインストールおよび設定することもできます。詳細については、『Red Hat Satellite Virtual Instances Guide』を参照してください。

自己登録 Satellite Server には複数の制限があります。

- 自己登録 Satellite Server は、ライフサイクル環境を使用してパッケージアップデートをテストできません。未テストのパッケージにアップグレードする前に、自己登録 Satellite Server の完全バックアップを作成することが重要です。
- すべての puppet モジュールが自己登録 Satellite Server でサポートされるわけではありません。puppet モジュールを自己登録 Satellite Server に適用する場合は、未サポートの設定が作成されないようにしてください。

Satellite の自己登録

自己登録 Satellite がそれ自体からアップデートを取得するよう設定する前に、Satellite サブスクリプションを Satellite のマニフェストに追加する必要があります。サブスクリプションがマニフェストに含まれる場合、適切な Satellite リポジトリーと Satellite を同期できます。

Satellite の自己登録手順:

1. Satellite がすでに Red Hat カスタマーポータルに登録されている場合は、以下のコマンドを使用して Red Hat カスタマーポータルから Satellite を登録解除します。

```
# subscription-manager remove --all
# subscription-manager unregister
```

2. Red Hat カスタマーポータルで Satellite サブスクリプションが利用可能になりました。サブスクリプションは、Satellite のマニフェストに移行できます。マニフェストの詳細については、『**Content Management Guide**』の「Managing Subscriptions」を参照してください。

   a. https://access.redhat.com に移動し、ページ上部のメインメニューにある **サブスクリプション** をクリックします。

   b. **Red Hat サブスクリプション管理** セクションまで下にスクロールし、**サブスクリプション管理アプリケーション** 下の **Satellite** をクリックします。

   c. 表でホスト名をクリックして、必要な Satellite Server を選択します。

   d. **サブスクリプションのアタッチ** をクリックし、割り当てるサブスクリプションを選択します。各サブスクリプションの数を指定し、ボタン **選択項目のアタッチ** をクリックします。

3. Satellite Server のマニフェストを更新します。

   a. **Satellite** サーバーにログインします。

   b. 正しい組織が選択されていることを確認します。

   c. **コンテンツ** > **Red Hat サブスクリプション** をクリックし、次にページの右上にある **マニフェストの管理** をクリックします。

   d. **サブスクリプションマニフェスト** セクションで、**アクション** をクリックし、**サブスクリプションマニフェスト** サブセクションで、**マニフェストの更新** をクリックします。

4. Satellite Web UI またはコマンドラインインターフェースを使用して Red Hat リポジトリーを有効にします。

   - **Satellite Web UI の使用:**

     a. **コンテンツ** > **Red Hat リポジトリー** をクリックします。

     b. 必要なリポジトリーに移動します。リポジトリーを選択する各リポジトリーセットをクリックし、必要な各リポジトリーのチェックボックスをオンにします。リポジトリーは自動的に有効になります。

        - **`Red Hat Enterprise Linux 6`** の場合、有効にする必要があるリポジトリーは以下のとおりです。

          - Red Hat Enterprise Linux 6 Server RPM x86_64 6Server
          - Red Hat Satellite 6.2 for RHEL 6 Server RPMs x86_64
          - Red Hat Software Collections RPMs for Red Hat Enterprise Linux 6 Server x86_64 6Server
          - Red Hat Enterprise Linux 6 Server - Satellite Tools 6.2 RPMs x86_64 リポジトリー

        - **`Red Hat Enterprise Linux 7`** の場合、有効にする必要があるリポジトリーは以下のとおりです。

          - Red Hat Enterprise Linux 7 Server RPMs x86_64 6Server
          - Red Hat Satellite 6.2 for RHEL 7 Server RPMs x86_64

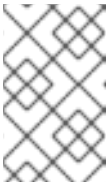
- Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64 6Server
- Red Hat Enterprise Linux 7 Server - Satellite Tools 6.2 RPMs x86_64 リポジトリー

- **Hammer CLI Tool の使用:**
  Satellite Server に必要なリポジトリーを有効にするには、以下の形式のコマンドを使用します。

```
subscription-manager repos --enable=repository-to-be-enabled
```

  a. **Red Hat Enterprise Linux 6** の場合、有効にする必要があるリポジトリーは以下のとおりです。
     i. rhel-6-server-satellite-6.2-rpms
     ii. rhel-6-server-satellite-tools-6.2-rpms
  b. **Red Hat Enterprise Linux 7** の場合、有効にする必要があるリポジトリーは以下のとおりです。
     i. rhel-7-server-satellite-6.2-rpms
     ii. rhel-7-server-satellite-tools-6.2-rpms

5. Satellite Server を同期します。
   a. **Content (コンテンツ)** > **Sync Status (同期ステータス)** に移動します。有効なサブスクリプションとリポジトリーに基づいて、同期できる製品リポジトリーのリストが表示されます。
   b. 製品名の横にある矢印をクリックして使用可能なコンテンツを表示します。
   c. 同期するコンテンツを選択します。
   d. **Synchronize Now (今すぐ同期)** をクリックして同期を開始します。同期プロセスのステータスは、**Result (結果)** 列に表示されます。同期が成功したら、**Sync complete (同期が完了)** が **Result (結果)** 列に表示されます。同期が失敗した場合は、**Error syncing (同期エラー)** が表示されます。



注記

コンテンツの同期には時間がかかることがあります。同期に要する時間は、ディスクドライブの速度やネットワーク接続の速度、同期対象として選択されたコンテンツの量によって異なります。

6. オプションで、Satellite Server を表すコンテンツビューを作成します。これにより、Satellite ではサーバー上の残りのコンテンツと同じライフサイクル管理手順を実行できます。コンテンツビューの詳細については、『Red Hat Satellite 6.2 Host Configuration Guide』の章「Using Content Views」を参照してください。
   a. コンテンツビューを作成するには、以下の手順を実行します。
      i. Satellite 管理者として Web UI ログインします。
      ii. **Content (コンテンツ)** > **Content Views (コンテンツビュー)** をクリックします。

iii. **Create New View (新規ビューの作成)** をクリックします。

iv. コンテンツビューの **Name (名前)** を指定します。**Name (名前)** フィールドに値が入力されると、**Label (ラベル)** フィールドに値が自動的に入力されます。オプションで、コンテンツビューの説明を提供できます。

v. **Save (保存)** をクリックします。

b. コンテンツビューを編集して Red Hat Enterprise Linux サーバーと Satellite リポジトリーを追加します。

i. **Content (コンテンツ)** > **Content Views (コンテンツビュー)** をクリックし、リポジトリーを追加するコンテンツビューを選択します。

ii. **Yum Content (Yum コンテンツ)** をクリックし、ドロップダウンメニューから **Repositories (リポジトリー)** を選択します。サブメニューから、**Add (追加)** をクリックします。

iii. 追加する必要があるリポジトリーを選択し、Add Repositories (リポジトリーの追加) をクリックします。自己登録 Satellite に必要なリポジトリーは、Satellite 向けのすべてのリポジトリー、すべてのサポートリポジトリー、および Base OS 向けリポジトリーです。自己登録 Satellite に必要なリポジトリーはこの手順の手順 4 にリストされています。

7. 以下のコマンドを実行して必要な証明書をダウンロードおよびインストールします。

```
# rpm -Uvh /var/www/html/pub/katello-ca-consumer-latest.noarch.rpm
```

8. Satellite Server を登録し、適切なエンタイトルメントを割り当てます。Satellite Server を登録する場合は、サーバーが属する組織とライフサイクル環境を指定する必要があります。利用可能な組織とライフサイクル環境を確認するには、Satellite Web UI で、**Hosts （ホスト)** > **New host (新規ホスト)** に移動し、これらの値に対してドロップダウンリストを選択します。

```
# subscription-manager register --org=organization  --
environment=environment
```

例

```
# subscription-manager register --org=ExampleCompany  --
environment=Library
```

Red Hat Satellite ユーザー名およびパスワードを入力するよう求められます。Satellite Server 管理者は、新規ユーザーを設定できます。詳細については、『Red Hat Satellite Server Administration Guide』を参照してください。

9. 以下のコマンドを実行して Satellite と Red Hat Enterprise Linux のプール ID を見つけます。

```
# subscription-manager list --available
```

10. 以下のコマンドを実行してエンタイトルメントを割り当てます。

```
# subscription-manager attach --pool Red_Hat_Satellite_Pool_ID --
pool Red_Hat_Enterprise_Linux_ID
```

この時点で、Satellite Server の内部の Satellite Server に対してコンテンツホストが作成されます。

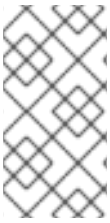
11. Satellite Web UI からエラータ管理とパッケージインストールを許可するために Katello Agent パッケージをインストールします。**katello-agent** パッケージは、goferd サービスを提供する gofer パッケージに依存します。Red Hat Satellite Server または Capsule Server がコンテンツホストに適用可能なエラータに関する情報を提供できるように goferd サービスを有効にする必要があります。
katello-agent をインストールするには、以下のコマンドを実行します。

```
# yum install katello-agent
```

katello-agent が正常にインストールされた後に、goferd サービスが起動し、有効になります。

3.2. 切断されたネットワークからのダウンロードおよびインストール

Red Hat Satellite Server に対するホストが切断された環境に存在する場合は、ISO イメージを使用して Satellite Server をインストールできます。ISO イメージには最新のアップデート、バグフィックス、および機能が含まれないことがあるため、この方法は他のすべての状況に推奨されません。



注記

ベースシステムが Red Hat CDN から更新されなかった場合、パッケージの依存エラーが発生することがあります。必要なパッケージの最新バージョンは手動でダウンロードおよびインストールする必要があります。詳細については、「パッケージを手動でダウンロード」 を参照してください。

作業を開始する前に

- インストールで使用されたリポジトリーのコピーは **/opt/** ディレクトリーに格納されます。このファイルシステムとディレクトリーのために最低 2GB の領域を確保してください。

3.2.1. バイナリー DVD イメージのダウンロード

1. Red Hat カスタマーポータルに移動し、ログインします。
2. **ダウンロード**をクリックします。
3. **Red Hat Enterprise Linux** を選択します。
4. 製品とバージョンがご使用の環境に適切であることを確認します。
   - **Product Variant (製品のバリアント)** は **Red Hat Enterprise Linux Server** に設定されます。
   - **Version (バージョン)** は、ベースシステムとして使用する予定の製品の最新マイナーバージョンに設定されます。
   - **Architecture (アーキテクチャー)** は 64 ビットバージョンに設定されます。
5. **Product Software (製品ソフトウェア)** タブで、最新の Red Hat Enterprise Linux Server バージョン向けのバイナリー DVD イメージをダウンロードします。
6. **ダウンロード**をクリックし、**Red Hat Satellite** を選択します。

7. 製品とバージョンがご使用の環境に適切であることを確認します。
   - **Product Variant (製品のバリアント)** は **Red Hat Satellite** に設定されます。
   - **Version (バージョン)** は、ベースシステムとして使用する予定の製品の最新マイナーバージョンに設定されます。
   - **Architecture (アーキテクチャー)** は 64 ビットバージョンに設定されます。
8. **Product Software (製品ソフトウェア)** タブで、最新の Red Hat Satellite バージョン向けのバイナリー DVD イメージをダウンロードします。
9. ISO ファイルを Satellite ベースシステムまたは他のアクセス可能なストレージデバイスにコピーします。

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

3.2.2. オフラインリポジトリーでベースシステムを設定

1. ベースシステムのバージョンに対応する ISO ファイルのマウントポイントとして使用するディレクトリーを作成します。

```
# mkdir /media/rhelX-server
```

   ここで、**X** は、使用している Red Hat Enterprise Linux のメジャーバージョンです。

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

```
mount -o loop rhelX-Server-DVD.iso /media/rhelX-server
```

   以下の例は、Red Hat Enterprise Linux 7.2 のマウントを示しています。

```
# mount -o loop RHEL-7.2-20151030.0-Server-x86_64-dvd1.iso
/media/rhel7-server
mount: /dev/loop0 is write-protected, mounting read-only
```

3. ISO ファイルのリポジトリーデータファイルをコピーします。

```
# cp /media/rhelX-server/media.repo /etc/yum.repos.d/rhelX-
server.repo
```

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

```
baseurl=file:///media/rhelX-server/
```

   以下の例は、Red Hat Enterprise Linux 7.2 を使用した場合のリポジトリーデータファイルを示しています。

```
# vi /etc/yum.repos.d/rhel7-server.repo
[InstallMedia]
name=Red Hat Enterprise Linux 7.2
mediaid=1446216863.790260
metadata_expire=-1
gpgcheck=0
```

```
cost=500
baseurl=file:///media/rhel7-server/
enabled=1
```

5. リポジトリーが設定されたことを確認します。

```
# yum repolist
Loaded plugins: product-id, search-disabled-repos, subscription-
manager
This system is not registered to Red Hat Subscription Management.
You can use subscription-manager to register.
repo id          repo name                          status
InstallMedia     Red Hat Enterprise Linux 7.2    4,620
```

6. ベースシステムのバージョンに対応する ISO ファイルのマウントポイントとして使用するディレクトリーを作成します。

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

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

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

以下の例は、Red Hat Enterprise Linux 7 向け Red Hat Satellite 6.2.1 を使用した ISO のマウントを示しています。

```
# mount -o loop satellite-6.2.1-rhel-7-x86_64-dvd.iso /media/sat6
mount: /dev/loop1 is write-protected, mounting read-only
```

3.2.3. オフラインリポジトリーからのインストール

1. Red Hat GPG キーをインポートします。

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

2. バイナリー DVD イメージを使用してベースシステムを最新の状態にします。

```
# yum update
```

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

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

4. マウントされたディレクトリーでインストーラースクリプトを実行します。

```
# ./install_packages
 This script will install the foreman packages on the current
machine.
   - Ensuring we are in an expected directory.
   - Copying installation files.
   - Creating a Repository File
   - Creating RHSCL Repository File
   - Checking to see if Foreman is already installed.
```

```
   - Importing the gpg key.
   - Foreman is not yet installed, installing it.
   - Installation repository will remain configured for future
package installs.
   - Installation media can now be safely unmounted.

Install is complete. Please run satellite-installer.
```

パッケージが不明であるか、古いため、スクリプトが失敗した場合は、これらをダウンロードし、個別にインストールする必要があります。手順については、「パッケージを手動でダウンロード」 を参照してください。

5. ISO ファイルをアンマウントします。

```
# unmount /media/sat6
# unmount /media/rhelX-server
```

3.2.4. パッケージを手動でダウンロード

パッケージを手動でダウンロードする必要がある場合は、以下の手順を実行します。

1. Red Hat カスタマーポータルに移動し、ログインします。
2. **ダウンロード**をクリックします。
3. **Red Hat Satellite** を選択します。
4. 製品とバージョンがご使用の環境に適切であることを確認します。
   - **Product Variant (製品のバリアント)** は **Red Hat Satellite** に設定されます。
   - **Version (バージョン)** は、ベースシステムとして使用する製品の最新マイナーバージョンに設定されます。
   - **Architecture (アーキテクチャー)** は 64 ビットバージョンに設定されます。
5. **Packages (パッケージ)** タブで、Search (検索) ボックスに必要なパッケージの名前を入力します。
6. 必要なパッケージの横にある **Download Latest (最新版のダウンロード)** をクリックします。

3.3. 初期設定の実行

初期設定の一部として、カスタムサーバー証明書を設定し、手動で Satellite を設定したり、回答ファイルを使用して Satellite を自動的に設定したりできます。

- 手動設定 - Satellite Server には、サーバーで使用できるデフォルトの初期設定オプションがあります。これらの設定は、使用している環境の要件に応じて上書きできます。必要なオプションを設定するために、コマンドは、必要に応じて実行できます。
- 自動設定 - 回答ファイルを使用することにより、ほとんどのインストールと設定を自動化できます。

注記

Satellite インストーラーの実行時に使用するオプションによっては、設定が完了するのに数分かかることがあります。

作業を進める前に、使用している環境に適切なマニフェストまたはパッケージを確認します。詳細については、『Content Management Guide』を参照してください。

3.3.1. 時間の同期

時刻の誤差を最小化するには、ホストオペレーティングシステムで時刻シンクロナイザーを起動し、有効にする必要があります。システムの時刻が正しくない場合は、証明書の検証に失敗することがあります。

NTP と **chrony** の 2 つの時刻シンクロナイザーが利用可能です。各シンクロナイザーにはそれぞれの利点があります。**chrony** は、頻繁に一時停止するシステムと、ネットワークから断続的に切断され、ネットワーク接続が再確立されるシステム (モバイルシステムや仮想システムなど) に推奨されます。**NTP** は、実行状態を維持し、中断なしでネットワークに接続することが期待されるシステムに推奨されます。

NTP と **chrony** の違いに関する詳細については、『**Red Hat Enterprise Linux 7 System Administrators Guide**』または『**Red Hat Enterprise Linux 6 Deployment Guide**』を参照してください。

NTP を使用した時刻の同期

1. ntp をインストールします。

```
# yum install ntp
```

2. NTP サーバーが利用可能であることを確認します。

```
# ntpdate -q ntp_server_address
```

3. システム時刻を設定します。

```
# ntpdate ntp_server_address
```

4. ntpd サービスを起動して、有効にします。

```
# chkconfig ntpd on
```

chronyd を使用した時刻の同期

1. chronyd をインストールします。

```
# yum install chrony
```

2. chrony サービスを起動し、有効にします。

```
# systemctl start chronyd
# systemctl enable chronyd
```

3.3.2. ホスト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

ホストオペレーティングシステムには **sos** パッケージをインストールする必要があります。**sos** パッケージを使用すると、Red Hat Enterprise Linux システムから設定と診断情報を収集できます。また、Red Hat テクニカルサポートでサービス要求をオープンするときに必要な初期システム分析を提供することもできます。sos の使用の詳細については、「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。

1. sos パッケージをインストールします。

```
# yum install sos
```

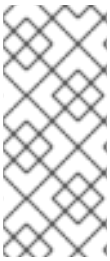
3.3.3. 初期設定を手動で実行

初期設定では、組織、場所、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場所を作成できます。

インストールプロセスが完了するには何十分も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 **screen** などのユーティリティーの使用を検討してください。Red Hat ナレッジベースの記事 How do I use the screen command? には **screen** のインストールについて記載されています。また、詳細については、**screen** man ページを参照してください。インストールコマンドが実行中のシェルへの接続が失われた場合は、**/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。

Satellite Server の手動設定

値を指定しない場合は、デフォルト値が使用されます。**satellite-installer --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。



注記

オプション **--foreman-initial-organization** に意味のある値を指定します。これには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、このラベルはあとで簡単に変更できません。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。

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-admin-username admin-username \
--foreman-admin-password admin-password
```

スクリプトが正常に完了すると、以下の出力が表示されます。

```
Installing             Done
   [100%] [.......................................]
   Success!
   * Satellite is running at https://satellite.example.com
       Default credentials are 'admin / changeme'
   * Capsule is running at https://satellite.example.com:9090
   * To install additional capsule on separate machine continue by
running:
```

```
   capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
"~/$CAPSULE-certs.tar"

   The full log is at /var/log/foreman-installer/satellite.log
```

3.3.4. Answer ファイルを使用した Red Hat Satellite の設定

回答ファイルを使用すると、カスタマイズされたオプションでインストールを自動化できます。最初の回答ファイルには、部分的に情報が入力されます。回答ファイルには、**satellite-installer** の初回実行後に、インストール向けの標準的なパラメーター値が入力されます。

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 を使用する必要があります。

1. デフォルトの回答ファイル **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** をローカルファイルシステムの場所にコピーします。

```
# cp /etc/foreman-installer/scenarios.d/satellite-answers.yaml
/etc/foreman-installer/scenarios.d/my-answer-file.yaml
```

2. 設定可能なすべてのオプションを表示するには、**satellite-installer --help** コマンドを実行します。
3. 回答ファイルのコピーを開き、ご使用の環境に適した値を編集し、ファイルを保存します。
4. **/etc/foreman-installer/scenarios.d/satellite.yaml** ファイルを開き、カスタム回答ファイルを参照する回答ファイルエントリーを編集します。

```
:answer_file: /etc/foreman-installer/scenarios.d/my-answer-file.yaml
```

5. **satellite-installer** コマンドを実行します。

```
# satellite-installer --scenario satellite
```

3.4. 追加設定の実行

3.4.1. Satellite Tools リポジトリーのインストール

Satellite Tools リポジトリーは、Satellite Server に登録されたクライアント向けの **katello-agent** パッケージと **puppet** パッケージを提供します。クライアントのリモートアップデートを許可するために、katello エージェントをインストールすることが推奨されます。自己登録 Satellite Server または Capsule Server のベースシステムは Satellite Server のクライアントであるため、katello エージェントもインストールする必要があります。

Satellite Tools リポジトリーのインストール手順:

1. Satellite Web UI で、**Content** > **Red Hat Repositories** に移動し、**RPM** タブを選択します。
2. Red Hat Enterprise Linux Server 項目を見つけ、展開します。
3. Red Hat Satellite Tools 6.2 (RHEL **VERSION** Server 向け) (RPM) 項目を見つけ、展開します。Red Hat Satellite Tools 6.2 項目が非表示の場合は、その項目がカスタマーポータルから取得さ

れたサブスクリプションマニフェストに含まれないことが原因であることがあります。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、To correct that, log in to the Customer Portal, add these repositories, download the subscription manifest and import it into Satellite.

4. Satellite 6.2 Tools リポジトリーの名前の横にある **Enabled** チェックボックスをオンにします。

ホストで実行されている 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 Tools 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。

Satellite Tools リポジトリーの同期方法:

1. **Content** > **Sync Status** に移動します。
同期可能な製品リポジトリーのリストが表示されます。

2. 製品コンテンツ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. 同期するコンテンツを選択します。

4. **Synchronize Now** をクリックします。

3.4.2. HTTP プロキシーを使用して Satellite Server を設定

ネットワークで HTTP プロキシーを使用している場合は、HTTP プロキシーを有効にできます。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。

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

```
# export http_proxy=""
# export https_proxy=$http_proxy
# export no_proxy=$http_proxy
```

2. HTTP プロキシーオプションを使用して **`satellite-installer`** を実行します。

```
# satellite-installer --scenario satellite \
--katello-proxy-url=http://myproxy.example.com \
--katello-proxy-port=8080 \
--katello-proxy-username=proxy_username \
--katello-proxy-password=proxy_password
```

3. Satellite Server が Red Hat Content Delivery Network (CDN) に接続し、リポジトリーを同期できることを確認します。

   a. ネットワークゲートウェイと HTTP プロキシーで、以下のホスト名に対して TCP を有効にします。

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

subscription.rhn.redhat.com により現在使用されている IP アドレスについては、Red Hat 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 the IP address range for 'subscription.rhn.redhat.com'」を参照してください。

Red Hat CDN (cdn.redhat.com) により使用されている IP アドレスのリストについては、Red Hat カスタマーポータルのナレッジベース記事「Public CIDR Lists for Red Hat」を参照してください。

b. 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 = http_proxy.example.com

# port for http proxy server
proxy_port = 3128

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

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

3.4.3. Satellite Server で DNS、DHCP、および TFTP を設定

Satellite Server では、DNS、DHCP、および TFTP を設定できます。

外部サービスを設定する場合は、詳細について「Satellite Server での外部サービスの設定」を参照してください。

これらのサービスを手動で管理するために Satellite でサービスを無効にする場合は、詳細について「管理対象外ネットワークに対して DNS、DHCP、および TFTP を無効化」を参照してください。

設定可能なオプションの完全なリストを表示するには、**satellite-installer --help** コマンドを実行します。

作業を開始する前に

- 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します。
- 以下の情報を用意する必要があります。
  - 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。

注記

タスクの情報は例です。自分の環境に該当する情報を使用する必要があります。

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

1. 使用している環境に適切なオプションを使用して **satellite-installer** を実行します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-servername $(hostname)
```

インストールのステータスが表示されます。ユーザー名とパスワードはコマンド出力で参照できます。また、これらの情報は **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** ファイルの **admin_password** パラメーターからも取得できます。

```
Success!
  * Satellite is running at https://satellite.example.com
      Default credentials are 'admin:*******'
  * Capsule is running at https://satellite.example.com:9090
  * To install additional capsule on separate machine continue by
running:"

      capsule-certs-generate --capsule-fqdn "$CAPSULE" --certs-tar
"~/$CAPSULE-certs.tar"

  The full log is at /var/log/foreman-installer/satellite.log
```

注記

設定を変更するには、**satellite-installer** を再び実行する必要があります。スクリプトは複数回実行でき、すべての設定ファイルが変更された値で更新されます。

3.4.4. 管理対象外ネットワークに対して DNS、DHCP、および TFTP を無効化

Satellite 6 は、Satellite の内部または外部 Capsule で実行されている TFTP、DHCP、および DNS ネッ

トワークサービス向けの完全な管理機能を提供します。これらのサービスを手動で管理、または外部の手段を使用する場合、Satellite 6 はそれらと直接統合できません。Foreman Hooks を介してカスタム統合スクリプトを開発できる一方で (新しいホストの作成後の DNS レコードの作成など) 、DHCP と DNS の検証エラーを回避するためにこの統合 (オーケストレーションとも呼ばれます) は無効にする必要があります。

1. **Infrastructure** > **Subnets** に移動し、サブネットを選択します。
2. **Capsules** タブで、ドロップダウンリストを **None** に設定して、関連付けられた DHCP Capsule または TFTP Capsule がないことを確認します。
3. 正引きレコードオーケストレーションを無効にします。
   a. **Infrastructure** > **Domains** に移動し、ドメインを選択します。
   b. **Domain (ドメイン)** タブで、**DNS Capsule** ドロップダウンリストを **None (なし)** に設定します。
4. 逆引き (PTR) レコードオーケストレーションを無効にします。
   a. **Infrastructure** > **Subnets** に移動し、サブネットを選択します。
   b. **Capsules** タブで、**Reverse DNS Capsule** ドロップダウンリストを **None (なし)** に設定します。

注記

Satellite 6 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すると、期待されたレコードと設定ファイルが存在しない場合に、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、将来ホストの削除に失敗することを回避するために、既存の Satellite 6 管理対象ホストに対して必要な DHCP および DNS レコードと TFTP ファイルが所定の場所にあることを確認します。

3.4.5. Satellite Server で送信電子メールを設定

Satellite Server から電子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。

1. お好みの配信方法に合わせて設定ファイル **/etc/foreman/email.yaml** を編集します。以下の例は、SMTP サーバーを使用する場合の設定ファイルの内容を示しています。

```
production:
   delivery_method: :smtp
   smtp_settings:
      address: smtp.example.com
      port: 25
      domain: example.com
      authentication: :login
      user_name: satellite@example.com
      password: satellite
```

ここで、**user_name** ディレクティブと **password** ディレクティブは SMTP サーバーのログインクレデンシャルを指定します。デフォルトの **/etc/foreman/email.yaml** には **authentication: :none** が含まれます。

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。

```
production:
   delivery_method: :smtp
   smtp_settings:
      enable_starttls_auto: :true
      address: smtp.gmail.com
      port: 587
      domain: smtp.gmail.com
      authentication: :plain
      user_name: user@gmail.com
      password: password
```

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。

```
production:
   delivery_method: :sendmail
   sendmail_settings:
      arguments: "-i -t -G"
```

ここで、**arguments** ディレクティブは、コマンドラインオプションを **sendmail** に渡すために使用されます。**arguments** のデフォルト値は "-i -t" です。詳細については、**sendmail 1** の man ページを参照してください。

2. 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、以下のディレクティブを smtp_settings 下の **/etc/foreman/email.yaml** に追加します。

     ```
     enable_starttls_auto: :false
     ```

3. **/etc/foreman/email.yaml** ファイルの更新後に、Katello サービスを再起動して変更を適用します。

   ```
   # katello-service restart
   ```

4. 返信アドレスまたは件名プレフィックスなどの追加の電子メール設定は、Satellite Web UI (**General** タブ下で **Administer (管理)** > **Settings (設定)**) で行えます。

注記

個々のユーザーまたはユーザーグループに対する電子メール通知の設定については、『Red Hat Satellite Server Administration Guide』を参照してください。

3.4.6. カスタムサーバー証明書を使用した Satellite Server の設定

Red Hat Satellite 6 は、Satellite Server、Capsule Server、およびすべてのホスト間で暗号化された通信を有効にするためにデフォルトの SSL 証明書を提供します。必要に応じて、デフォルト証明書をカスタム証明書に置き換えることができます。たとえば、会社のセキュリティーポリシーで、特定の認証局から SSL 証明書を取得することが規定されている場合があります。

デフォルトの SSL 証明書を置き換えるには、Satellite Server とすべての外部 Capsule Server (存在する場合) 向けのカスタム SSL 証明書を取得し、個々のホストでインストールする必要があります。

注記

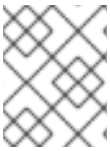
この手順を**実行する前**に、Satellite Server とすべての外部 Capsule Server 向けのカスタム SSL 証明書を取得します。

Satellite サーバーでカスタム証明書を使用するには、これらの手順を完了します。

1. 「Satellite Server 向けの SSL 証明書を取得」
2. 「Satellite Server の SSL 証明書の検証」
3. 「カスタム証明書パラメーターを使用した Satellite インストーラーの実行」
4. 「Satellite Server に接続されたすべてのホストに新しい証明書をインストール」

外部 Capsule サーバーがある場合は、「カスタムサーバー証明書を使用した Capsule Server の設定」の手順も完了する必要があります。

3.4.6.1. Satellite Server 向けの SSL 証明書を取得



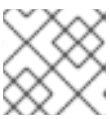
注記

Satellite Server 向けのカスタム SSL 証明書がすでにある場合は、この手順を省略します。

1. **root** ユーザーのみがアクセスできる、すべてのソース証明書ファイルを含むディレクトリーを作成します。
   これらの例では、ディレクトリーは **/root/sat_cert** です。

```
# mkdir /root/sat_cert
# cd /root/sat_cert
```

2. Certificate Signing Request (CSR) を署名する秘密鍵を作成します。



注記

Satellite Server 向け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/sat_cert/satellite_cert_key.pem 4096
```

3. Certificate Signing Request (CSR) の作成
   Certificate Signing Request は、証明書を要求しているサーバーの詳細を含むテキストファイル

です。このコマンドを使用する場合は、(前の手順で出力された) 秘密鍵を提供し、Satellite Server に関するいくつかの質問に答えます。その結果、Certificate Signing Request が作成されます。

注記

証明書の Common Name (CN) は、証明書が使用されるサーバーの完全修飾ドメイン名 (FQDN) に一致する必要があります。Satellite サーバー向けの証明書を要求している場合、これは Satellite サーバーの FQDN です。Capsule サーバー向けの証明書を要求している場合、これは Capsule サーバーの FQDN です。

サーバーの FQDN を確認するには、該当するサーバーでコマンド **hostname -f** を実行します。

```
# openssl req -new \
  -key /root/sat_cert/satellite_cert_key.pem \ 1
  -out /root/sat_cert/satellite_cert_csr.pem   2
```

1. 証明書を署名するために使用する Satellite Server の秘密鍵
2. Certificate Signing Request ファイル

Certificate Signing Request セッションの例

```
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:Queensland
Locality Name (eg, city) [Default City]:Brisbane
Organization Name (eg, company) [Default Company Ltd]:Example
Organizational Unit Name (eg, section) []:Sales
Common Name (eg, your name or your server's hostname)
[]:satellite.example.com
Email Address []:example@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:Example
```

4. 証明書要求を認証局に送信します。
要求を送信する場合は、証明書のライフスパンを指定する必要があります。証明書要求を送信する方法は異なるため、推奨される方法について認証局にお問い合わせください。要求に対する応答で、認証局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。

3.4.6.2. Satellite Server の SSL 証明書の検証

以下の例のように、必要なパラメーターを使用して **katello-certs-check** コマンドを実行します。これにより、カスタム証明書に必要な入力ファイルが検証され、これらを Satellite サーバー、すべての Capsule サーバー、および Satellite で管理されているホストにインストールするために必要なコマンドが出力されます。

1. カスタム SSL 証明書入力ファイルを検証します。ファイルに一致するようファイル名を変更します。

```
# katello-certs-check \
    -c /root/sat_cert/satellite_cert.pem \        1
    -k /root/sat_cert/satellite_cert_key.pem \    2
    -r /root/sat_cert/satellite_cert_csr.pem \    3
    -b /root/sat_cert/ca_cert_bundle.pem          4
```

1. 認証局により署名された Satellite Server 向けの証明書ファイル
2. 証明書を署名するために使用する Satellite Server の秘密鍵
3. Satellite Server 向けの証明書署名要求ファイル
4. 認証局バンドル

katello-certs-check の出力例

```
Validating the certificate subject=
/C=AU/ST=Queensland/L=Brisbane/O=Example/OU=Sales/CN=satellite.example.com
/emailAddress=example@example.com
Check private key matches the certificate: [OK]
Check ca bundle verifies the cert file: [OK]

Validation succeeded.

To install the Satellite main server with the custom certificates, run:

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

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

    satellite-installer --scenario satellite\
                        --certs-server-cert
"/root/sat_cert/satellite_cert.pem"\
                        --certs-server-cert-req
"/root/sat_cert/satellite_cert_csr.pem"\
                        --certs-server-key
"/root/sat_cert/satellite_cert_key.pem"\
```

```
                        --certs-server-ca-cert
"/root/sat_cert/ca_cert_bundle.pem"\
                        --certs-update-server --certs-update-server-ca

To use them inside a $CAPSULE, run this command INSTEAD:

    capsule-certs-generate --capsule-fqdn ""\
                           --certs-tar  "/root/certs.tar"\
                           --server-cert
"/root/sat_cert/satellite_cert.pem"\
                           --server-cert-req
"/root/sat_cert/satellite_cert_csr.pem"\
                           --server-key
"/root/sat_cert/satellite_cert_key.pem"\
                           --server-ca-cert
"/root/sat_cert/ca_cert_bundle.pem"\
                           --certs-update-server
```

3.4.6.3. カスタム証明書パラメーターを使用した Satellite インストーラーの実行

この時点で SSL 証明書が作成され、Red Hat Satellite 6 で使用できることが確認されました。次の手順は、カスタム SSL 証明書を Satellite Server とそのすべてのホストにインストールすることです。

この手順は、Satellite Server がすでにインストールされているかどうかに応じて、少し異なります。Satellite Server が**すでに**インストールされている場合は、既存の証明書を証明書アーカイブの証明書で**更新**する必要があります。

このセクションのコマンドは、「Satellite Server の SSL 証明書の検証」 で説明されたように **katello-certs-check** コマンドで出力され、ターミナルにコピーアンドペーストできます。

1. 状況に応じて、**satellite-installer** コマンドを実行します。

    a. Satellite がすでにインストールされている場合は、Satellite サーバーで以下のコマンドを実行します。

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

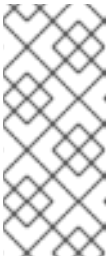
    このコマンドの重要なパラメーターには **--certs-update-server** と **--certs-update-server-ca** が含まれます。これにより、サーバーの SSL 証明書と認証局を更新するよう指定されます。すべてのインストーラーのパラメーターの簡単な説明については、コマンド **satellite-installer --scenario satellite --help** を実行します。

注記

satellite-installer コマンドのすべてのファイルについては、相対パス名ではなく完全パス名を使用します。インストーラーにより、すべてのファイルのパスと名前が記録されます。インストーラーを異なるディレクトリーから再び実行する場合は、元のファイルを見つけることができないため、失敗します。

b. Satellite が**まだインストールされていない**場合は、Satellite サーバーで以下のコマンドを実行します。

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



注記

satellite-installer コマンドのすべてのファイルについては、相対パス名ではなく完全パス名を使用します。インストーラーにより、すべてのファイルのパスと名前が記録されます。インストーラーを異なるディレクトリーから再び実行する場合は、元のファイルを見つけることができないため、失敗します。

2. 証明書をホストにインストールする前に証明書が Satellite サーバーに正常にインストールされていることを確認します。Satellite サーバーへのネットワークアクセスがあるコンピューターで、Web ブラウザーを起動し、URL **https://satellite.example.com** に移動して、証明書の詳細を参照します。

3.4.6.4. Satellite Server に接続されたすべてのホストに新しい証明書をインストール

カスタム SSL 証明書は Satellite サーバーにインストールされたため、Satellite サーバーに登録された各ホストにもインストールする必要があります。すべての該当するホストで以下のコマンドを実行します。

```
# yum -y localinstall http://satellite.example.com/pub/katello-ca-
consumer-latest.noarch.rpm
```

3.4.7. mongod へのアクセスの制限

データ損失の危険を減らすために、MongoDB データベースデーモン **mongod** へのアクセスは **apache** ユーザーと **root** ユーザーにだけ許可する必要があります。

Satellite Server と Capsule Server で **mongod** へのアクセスを制限するには、以下のコマンドを使用します。

Red Hat Enterprise Linux 6 でのファイアウォールの設定

1. Satellite Server と Capsule Server で **iptables** サービスを設定します。

```
# iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner --
```

```
uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner --
uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -j DROP
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner --
uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner --
uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -j DROP
service iptables save
```

Red Hat Enterprise Linux 7 でのファイアウォールの設定

1. Satellite Server と Capsule Server でファイアウォールを設定します。

```
# firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 28017 -j DROP
```

2. **--permanent** オプションを追加してコマンドを繰り返し、設定を永続化します。

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
```

```
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 28017 -j DROP
```

第4章 CAPSULE SERVER のインストール

Capsule Server をインストールする前に、ご使用の環境がインストールの要件を満たしていることを確認する必要があります。Capsule Server のインストール要件は Satellite Server と同じです。詳細については、「インストールのための環境準備」を参照してください。

4.1. SATELLITE SERVER への CAPSULE SERVER の登録

作業を開始する前に

- Satellite Server には、サブスクライブする組織の適切なリポジトリーを使用してマニフェストがインストールされている必要があります。マニフェストには Capsule のベースシステムと Capsule に接続されたすべてのクライアント向けのリポジトリーが含まれる必要があります。リポジトリーは同期する必要があります。マニフェストとリポジトリーの詳細については、『Content Management Guide』を参照してください。
- Satellite Server のベースシステムは、Capsule Server のベースシステムのホスト名を解決できる必要があり、Capsule Server のベースシステムは Satellite Server のベースシステムのホスト名を解決できる必要があります。
- Satellite Server のユーザー名とパスワードが必要です。詳細については、『**Red Hat Satellite 6.2 Server Administration Guide**』を参照してください。

Satellite Server への Capsule Server の登録

1. Capsule Server に Satellite Server の CA 証明書をインストールします。

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

2. 組織で Capsule Server を登録します。

```
# subscription-manager register --org organization_name
```

4.2. CAPSULE SERVER サブスクリプションの識別と割り当て

Capsule Server の登録後は、Capsule Server のサブスクリプションプール ID を識別する必要があります。プール ID を使用すると、必要なサブスクリプションを Capsule Server に割り当てることができます。Capsule Server のサブスクリプションがあると、Capsule Server のコンテンツ、Red Hat Enterprise Linux、Red Hat Software Collections (RHSCL)、および Red Hat Satellite にアクセスできます。これは唯一必要なサブスクリプションです。

1. Capsule Server のサブスクリプションを識別します。

```
# subscription-manager list --all --available
```

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。

```
+
+-------------------------------------------+
    Available Subscriptions
+-------------------------------------------+
```

```
Subscription Name: Red Hat Satellite Capsule Server
Provides:          Red Hat Satellite Proxy
                   Red Hat Satellite Capsule
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Satellite Capsule
                   Red Hat Enterprise Linux Server
                   Red Hat Enterprise Linux High Availability (for
RHEL Server)
                   Red Hat Software Collections (for RHEL Server)
                   Red Hat Enterprise Linux Load Balancer (for RHEL
Server)
SKU:               MCT0369
Pool ID:           9e4cc4e9b9fb407583035861bb6be501
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2022
System Type:       Physical
```

2. Satellite ホストに割り当てることができるように、プール ID をメモします。使用するプール ID は、ここで提示した例とは異なります。

3. プール ID を使用してサブスクリプションを Capsule Server に割り当てます。

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

この出力では、以下のような内容が表示されます。

```
Successfully attached a subscription for: Red Hat Capsule Server
```

4. サブスクリプションが正しく割り当てられたことを確認するには、以下のコマンドを実行します。

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

4.3. リポジトリーの設定

1. すべての既存のリポジトリーを無効にします。

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

2. Red Hat Satellite、Red Hat Enterprise Linux、および Red Hat Software Collections リポジトリーを有効にします。
Red Hat Software Collections リポジトリーは、リモート実行機能を含む一部の Red Hat Satellite 機能で必要な新しいバージョンの Ruby を提供します。

Red Hat Enterprise Linux リポジトリーが、使用してる特定のバージョンに一致することを確認します。

使用しているプラットフォーム	実行するコマンド
Red Hat Enterprise Linux 6	subscription-manager repos --enable rhel-6-server-rpms --enable rhel-6-server-satellite-capsule-6.2-rpms --enable rhel-server-rhscl-6-rpms
Red Hat Enterprise Linux 7	subscription-manager repos --enable rhel-7-server-rpms --enable rhel-7-server-satellite-capsule-6.2-rpms --enable rhel-server-rhscl-7-rpms

3. Red Hat 以外のすべての **yum** リポジトリーから残されたすべてのメタデータを削除します。

```
# yum clean all
```

4. リポジトリーが有効になっていることを確認します。

```
# yum repolist enabled
```

以下のような出力が表示されます。

```
Loaded plugins: langpacks, product-id, subscription-manager
repo id                                                 repo name
status
!rhel-7-server-rpms/7Server/x86_64                      Red Hat
Enterprise Linux 7 Server (RPMs)                         7,617
!rhel-7-server-satellite-capsule-6.2-rpms/x86_64     Red Hat
Satellite Capsule 6.2(for RHEL 7 Server) (RPMs)    176
repolist: 7,793
```

4.4. 時間の同期

時刻の誤差を最小化するには、ホストオペレーティングシステムで時刻シンクロナイザーを起動し、有効にする必要があります。システムの時刻が正しくない場合は、証明書の検証に失敗することがあります。

NTP と **chrony** の 2 つの時刻シンクロナイザーが利用可能です。各シンクロナイザーにはそれぞれの利点があります。**chrony** は、頻繁に一時停止するシステムと、ネットワークから断続的に切断され、ネットワーク接続が再確立されるシステム (モバイルシステムや仮想システムなど) に推奨されます。**NTP** は、実行状態を維持し、中断なしでネットワークに接続することが期待されるシステムに推奨されます。

NTP と **chrony** の違いに関する詳細については、『Red Hat Enterprise Linux 7 System Administrators Guide』または『Red Hat Enterprise Linux 6 Deployment Guide』を参照してください。

NTP を使用した時刻の同期

1. ntp をインストールします。

```
# yum install ntp
```

2. NTP サーバーが利用可能であることを確認します。

```
# ntpdate -q ntp_server_address
```

3. システム時刻を設定します。

```
# ntpdate ntp_server_address
```

4. ntpd サービスを起動して、有効にします。

```
# chkconfig ntpd on
```

chronyd を使用した時刻の同期

1. chronyd をインストールします。

```
# yum install chrony
```

2. chrony サービスを起動し、有効にします。

```
# systemctl start chronyd
# systemctl enable chronyd
```

4.5. CAPSULE SERVER のインストール

1. インストールパッケージをインストールします。

```
# yum install satellite-capsule
```

4.6. CAPSULE SERVER の初期設定の実行

このセクションでは、デフォルトの証明書、DNS、および DHCP の使用を含む Capsule サーバーのデフォルトのインストールのデモを行います。他の高度な設定オプションの詳細については、「Capsule Server での追加設定の実行」 を参照してください。

4.6.1. デフォルトのサーバー証明書を使用した Capsule Server の設定

Capsule Server で使用されているデフォルトの認証局 (CA) を使用できます (この認証局は、サブサービスを認証するためのサーバーおよびクライアントの SSL 証明書両方で使用されます)。

作業を開始する前に

- 必要なサブスクリプションが Capsule Server に割り当てられている必要があります。
- **katello-ca-consumer-latest** パッケージがインストールされている必要があります。
- Capsule Server が Satellite Server に登録されている必要があります。

デフォルトのサーバー証明書を使用した Capsule Server の設定

1. Satellite Server で証明書アーカイブを作成します。
   -

```
# capsule-certs-generate
--capsule-fqdn "mycapsule.example.com"\
--certs-tar "~/mycapsule.example.com-certs.tar"
```

2. **satellite-installer** パッケージが Capsule Server で利用可能であることを確認します。
3. 生成されたアーカイブ .tar ファイルを Satellite Server から Capsule Server にコピーします。
4. ご使用の環境のニーズに基づいて証明書を有効にします。詳細については、**satellite-installer --help** を参照してください。

```
# satellite-installer --scenario capsule\
    --capsule-parent-fqdn "satellite.example.com"\
    --foreman-proxy-register-in-foreman "true"\
    --foreman-proxy-foreman-base-url
"https://satellite.example.com"\
    --foreman-proxy-trusted-hosts "satellite.example.com"\
    --foreman-proxy-trusted-hosts "mycapsule.example.com"\
    --foreman-proxy-oauth-consumer-key
"UVrAZfMaCfBiiWejoUVLYCZHT2xhzuFV"\
    --foreman-proxy-oauth-consumer-secret
"ZhH8p7M577ttNU3WmUGWASag3JeXKgUX"\
    --capsule-pulp-oauth-secret "TPk42MYZ42nAE3rZvyLBh7Lxob3nEUi8"\
    --capsule-certs-tar "~/mycapsule.example.com-certs.tar"
```

4.7. CAPSULE SERVER での追加設定の実行

4.7.1. katello エージェントのインストール

クライアントのリモートアップデートを許可するために、katello エージェントをインストールすることが推奨されます。自己登録 Satellite Server または Capsule Server のベースシステムは Satellite Server のクライアントであるため、katello エージェントがインストールされている必要があります。

作業を開始する前に

- Satellite Server で Satellite Tools リポジトリーが有効にされている必要があります。
- Satellite Server で Satellite Tools リポジトリーが同期されている必要があります。

katello-agent のインストール手順:

1. システムにログインします。
2. このバージョンの Satellite 向け Satellite Tools リポジトリーを有効にします。
   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

     ```
     # subscription-manager repos --enable=rhel-7-server-satellite-
     tools-6.2-rpms
     ```

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable=rhel-6-server-satellite-
tools-6.2-rpms
```

3. パッケージをインストールします。

```
# yum install katello-agent
```

4.7.2. Capsule Server でリモート実行を有効化

Capsule Server のホストでコマンドを実行する場合は、リモート実行が有効である必要があります。リモート実行を有効にするには、Host Configuration Guide で示された手順を実行します。

4.7.3. Capsule Server へのライフサイクル環境の追加

Capsule Server でコンテンツ機能が有効な場合は、環境を追加する必要があります。環境を追加すると、Capsule Server で Satellite Server からのコンテンツを同期し、コンテンツをホストシステムに提供できます。

Capsule Server は、Satellite Serverの Hammer CLI を使用して設定されます。すべてのコマンドを Satellite Server で実行する必要があります。

1. root として Hammer CLI にログインします。
2. すべての Capsule Server のリストを表示し、ID をメモします。

```
# hammer capsule list
```

3. ID を使用して、Capsule Server の詳細を検証します。

```
# hammer capsule info --id capsule_id_number
```

4. 利用可能なライフサイクル環境を検証し、環境 ID をメモします。

```
# hammer capsule content available-lifecycle-environments --id
capsule_id_number
```

利用可能なライフサイクル環境は Capsule Server に対して利用可能ですが、現在接続されていません。

5. ライフサイクル環境を Capsule Server に追加します。

```
# hammer capsule content add-lifecycle-environment --id
capsule_id_number --environment-id environment_id_number
```

6. Capsule Server に追加する各ライフサイクル環境に対して手順を繰り返します。
7. Satellite Server 環境からのすべてのコンテンツを Capsule Server と同期するには、以下のコマンドを実行します。

```
# hammer capsule content synchronize --id capsule_id_number
```

8. Satellite Server 環境からの特定のライフサイクル環境を Capsule Server と同期するには、以下のコマンドを実行します。

```
# hammer capsule content synchronize --id external_capsule_id_number
--environment-id environment_id_number
```

4.7.4. 管理対象ホスト上での電源管理の有効化

Capsule Server でベースボード管理コントローラー (BMC) を有効にすると、IPMI (Intelligent Platform Management Interface) または類似したプロトコルを使用して管理対象ホストで電源管理コマンドを使用できます。

Satellite Capsule サーバー上の BMC サービスを使用すると、さまざまな電源管理タスクを実行できます。この機能の基礎となるプロトコルは IPMI (BMC 機能とも呼ばれます) です。IPMI は、ホストの CPU とは独立して実行される専用プロセッサーに接続された管理対象ハードウェア上の特別なネットワークインターフェースを使用します。多くのインスタンスでは、BMC 機能はシャーシ管理の一部としてシャーシベースのシステムに組み込まれます (シャーシの専用モジュール)。

BMC サービスの詳細については、『**Red Hat Satellite 6.2 Host Configuration Guide**』を参照してください。

作業を開始する前に

- すべての管理対象ホストにタイプが **BMC** のネットワークインターフェースが搭載されている必要があります。Satellite 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。

管理対象ホスト上での電源管理の有効化

1. BMC を有効にするためにオプションを使用してインストーラーを実行します。

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

4.7.5. Capsule Server での DNS と DHCP の設定

Capsule Server で DNS、DHCP、および TFTP を設定できます。

Capsule Server が外部 DNS および DHCP サービスを使用するよう設定することもできます。詳細については、「Satellite Server での外部サービスの設定」を参照してください。

設定可能なオプションの完全なリストを表示するには、**satellite-installer --help** コマンドを実行します。

作業を開始する前に

- DNS サーバーの適切なネットワーク名 (**dns-interface**) が用意されている必要があります。
- DHCP サーバーの適切なインターフェース名 (**dhcp-interface**) が用意されている必要があります。

Capsule Server での DNS、DHCP、および TFTP の設定

1. ご使用の環境に該当するオプションを使用して Capsule インストーラーを実行します。
以下の例は、完全なプロビジョニングサービスを示してます。

```
# satellite-installer --scenario capsule\
--foreman-proxy-tftp=true\
--foreman-proxy-foreman-oauth-key    "your_organization_key"\
--foreman-proxy-foreman-oauth-secret "your_organization_secret"\
--capsule-certs-tar            "~/capsule.example.com-certs.tar"\
--foreman-proxy-templates=true\
--foreman-proxy-dhcp=true\
--foreman-proxy-dhcp-gateway=192.168.122.1\
--foreman-proxy-dhcp-nameservers=192.168.122.1\
--foreman-proxy-dhcp-range="192.168.122.100 192.168.122.200"\
--foreman-proxy-dhcp-interface=eth0\
--foreman-proxy-dns=true\
--foreman-proxy-dns-forwarders=8.8.8.8\
--foreman-proxy-dns-interface=eth0\
--foreman-proxy-dns-zone=example.com

# satellite-installer --scenario capsule\
--foreman-proxy-dns true \
--foreman-proxy-dns-interface eth0 \
--foreman-proxy-dns-zone example.com \
--foreman-proxy-dns-forwarders 172.17.13.1 \
--foreman-proxy-dns-reverse 13.17.172.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "172.17.13.100 172.17.13.150" \
--foreman-proxy-dhcp-gateway 172.17.13.1 \
--foreman-proxy-dhcp-nameservers 172.17.13.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-servername $(hostname) \
--capsule-puppet true \
--foreman-proxy-puppetca true
```

4.7.6. カスタムサーバー証明書を使用した Capsule Server の設定

Red Hat Satellite 6 には、Satellite Server、Capsule Server、およびすべてのホスト間で暗号化された通信を可能にするデフォルトの SSL 証明書が含まれます。必要な場合は、デフォルト証明書をカスタム証明書に置き換えることができます。たとえば、会社のセキュリティーポリシーで、SSL 証明書を特定の認証局から取得することが規定されていることがあります。

前提条件

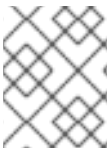
- カスタム証明書が設定された Satellite サーバー。詳細については、「カスタムサーバー証明書を使用した Satellite Server の設定」 を参照してください。
- インストールされ Satellite Server に登録された Capsule サーバー。詳細については、4 章*Capsule Server のインストール* を参照してください。

各 Capsule サーバー上のカスタム証明書を使用するには、以下の手順を実行します。

1. 「Capsule Server 向けの SSL 証明書を取得」
2. 「Capsule Server の SSL 証明書の検証」
3. 「Capsule サーバーの証明書アーカイブファイルの作成」

4. 「Capsule Server のカスタム証明書のインストール」
5. 「すべてのホスト上に Capsule Server の新しい証明書をインストール」

4.7.6.1. Capsule Server 向けの SSL 証明書を取得



注記

- 各サーバーの証明書は一意であるため、Satellite Server の証明書は Capsule Server で**使用しないでください**。

1. **root** ユーザーのみがアクセスできる、すべてのソース証明書ファイルを含むディレクトリーを作成します。

```
# mkdir /root/capsule_cert
# cd /root/capsule_cert
```

これらの例では、ディレクトリーは **/root/capsule_cert** です。複数の Capsule Server がある場合は、一致するディレクトリーを指定します。たとえば、**capsule_apac** と **capsule_emea** という名前の Capsule Server がある場合は、それぞれ **capsule_apac** と **capsule_emea** という名前のディレクトリーを作成できます。これは**必須**ではありませんが、ある Capsule Server のファイルを別の Capsule Server で使用する危険が減少します。

2. Certificate Signing Request (CSR) を署名する秘密鍵を作成します。

注記

Capsule Server 向け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/capsule_cert/satellite_cert_key.pem 4096
```

3. Certificate Signing Request (CSR) の作成
Certificate Signing Request は、証明書を要求しているサーバーの詳細を含むテキストファイルです。このコマンドを使用する場合は、(前の手順で出力された) 秘密鍵を提供し、Capsule Server に関するいくつかの質問に答えます。その結果、Certificate Signing Request がファイルに保管されます。

注記

証明書の Common Name (CN) は、証明書が使用されるサーバーの完全修飾ドメイン名 (FQDN) に一致する必要があります。

サーバーの FQDN を確認するために、サーバーでコマンド **hostname -f** を実行します。

```
# openssl req -new \
  -key /root/capsule_cert/capsule_cert_key.pem \ 1
  -out /root/capsule_cert/capsule_cert_csr.pem   2
```

1 証明書を署名するために使用される Capsule サーバーの秘密鍵

2 Certificate Signing Request ファイル

Certificate Signing Request セッションの例

```
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:AU
State or Province Name (full name) []:Queensland
Locality Name (eg, city) [Default City]:Brisbane
Organization Name (eg, company) [Default Company Ltd]:Example
Organizational Unit Name (eg, section) []:Sales
Common Name (eg, your name or your server's hostname)
[]:capsule.example.com
Email Address []:example@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:password
An optional company name []:Example
```

4. 証明書要求を認証局に送信します。
要求を送信する場合は、証明書のライフスパンを指定する必要があります。証明書要求を送信する方法は異なるため、推奨される方法について認証局にお問い合わせください。要求に対する応答で、認証局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。

4.7.6.2. Capsule Server の SSL 証明書の検証

Satellite サーバーで、**katello-certs-check** コマンドを使用して Capsule サーバーの証明書入力ファイルを検証します。

```
# katello-certs-check \
    -c /root/capsule_cert/capsule_cert.pem \        1
    -k /root/capsule_cert/capsule_cert_key.pem \    2
    -r /root/capsule_cert/capsule_cert_csr.pem \    3
    -b /root/capsule_cert/ca_cert_bundle.pem        4
```

1 認証局により提供された Capsule サーバー証明書ファイル

2 証明書を署名するために使用される Capsule サーバーの秘密鍵

3 Capsule サーバーの証明書署名要求ファイル

4 認証局により提供された認証局バンドル

証明書が正常に検証された場合、出力には以下の内容が含まれます。

■

```
Check private key matches the certificate: [OK]
Check ca bundle verifies the cert file: [OK]
```

「Capsule サーバーの証明書アーカイブファイルの作成」 に進みます。

4.7.6.3. Capsule サーバーの証明書アーカイブファイルの作成

Capsule サーバーのインストーラーでは、サーバーの証明書をアーカイブファイルで提供する必要があります。このファイルを作成するには、Satellite Server で **capsule-certs-generate** コマンドを使用します。

capsule-certs-generate コマンドは、各外部 Capsule Server に対して 1 回だけ実行する必要があります。これらの例では、**capsule.example.com** が FQDN の例であり、**capsule_certs.tar** がアーカイブファイル名の例です。これらをご使用の環境に適切な値に置き換えます。既存の証明書アーカイブファイルを上書きしないように注意してください。たとえば、**capsule1** と **capsule2** という名前の Capsule Server がある場合は、証明書アーカイブファイルの名前として **capsule1_certs.tar** と **capsule2_certs.tar** を指定できます。

1. 「Satellite Server の SSL 証明書の検証」 の **katello-certs-check** コマンドで出力されたようにターミナルに **capsule-certs-generate** コマンドをコピーアンドペーストします。

2. Capsule Server の FQDN に一致するよう **--capsule-fqdn** の値を編集し、証明書アーカイブファイルのファイルパスおよび名前に一致するよう **--certs-tar** の値を編集します。

3. Capsule Server がまだインストールされていない場合は、**--certs-update-server** パラメーターを削除します。これは、既存の Capsule Server の証明書を**更新**するためにのみ使用されます。

4. Satellite サーバーで、該当するコマンドを実行します。

   capsule-certs-generate コマンドの例

   ```
   # capsule-certs-generate --capsule-fqdn "capsule.example.com"\
     --certs-tar  "/root/capsule_cert/capsule_certs.tar"\
     --server-cert "/root/capsule_cert/capsule_cert.pem"\
     --server-cert-req "/root/capsule_cert/capsule_cert_csr.pem"\
     --server-key "/root/capsule_cert/capsule_cert_key.pem"\
     --server-ca-cert "/root/sat_cert/ca_cert_bundle.pem"\
     --certs-update-server
   ```

5. Satellite サーバーで、証明書アーカイブファイルを Capsule サーバーにコピーします。要求された場合は **root** ユーザーのパスワードを提供します。
   この例では、アーカイブファイルは **root** ユーザーのホームディレクトリーにコピーされますが、別の場所にコピーすることもできます。

   ```
   # scp /root/capsule_cert/capsule_certs.tar root@capsule.example.com:
   ```

「Capsule Server のカスタム証明書のインストール」 に進みます。

4.7.6.4. Capsule Server のカスタム証明書のインストール

警告

Capsule サーバーでこの手順を完了します。

Capsule サーバーのカスタム証明書をインストールするには、Satellite インストーラーを実行します。パラメーターを含むコマンドが 「Capsule サーバーの証明書アーカイブファイルの作成」 の **capsule-certs-generate** コマンドにより出力されます。

1. カスタム **capsule-certs-generate** コマンドをコピーアンドペーストします。ただし、コマンドを**実行しないでください**。
2. 証明書アーカイブファイルの場所に一致するように **--capsule-certs-tar** の値を編集します。
3. Capsule サーバーで追加機能を有効にする場合は、それらのパラメーターを **satellite-installer** コマンドに追加します。すべてのインストーラーのパラメーターについては、コマンド **satellite-installer --help** を実行してください。

カスタム satellite-installer コマンドの例

```
# satellite-installer --scenario capsule\
  --capsule-parent-fqdn                    "satellite.example.com"\
  --foreman-proxy-register-in-foreman      "true"\
  --foreman-proxy-foreman-base-url
"https://satellite.example.com"\
  --foreman-proxy-trusted-hosts            "satellite.example.com"\
  --foreman-proxy-trusted-hosts            "capsule.example.com"\
  --foreman-proxy-oauth-consumer-key
"FeQsbASvCjvvaqE6duKH6SoYZWg4jwjg"\
  --foreman-proxy-oauth-consumer-secret
"7UhPXFPDBongvdTbNixbsWR5WFZsKEgF"\
  --capsule-pulp-oauth-secret
"VpQ9587tVmYeuY4Du6VitmZpZE5vy9ac"\
  --capsule-certs-tar                      "/root/capsule_certs.tar"
```

注記

capsule-certs-generate コマンドにより出力されたように **satellite-installer** コマンドは、各 Capsule Server に対して一意です。複数の Capsule Server で 同じコマンドを**使用しないでください**。

証明書が関連するすべてのホストにデプロイされたあとであっても、証明書アーカイブファイル (.tar ファイル) は**削除しないでください**。このファイルは、たとえば、Capsule サーバーをアップグレードするときに必要になります。証明書アーカイブファイルがインストーラーよって検出されない場合は、以下のようなメッセージで失敗します。

```
[ERROR YYYY-MM-DD hh:mm:ss main]  tar -xzf
/var/tmp/srvcapsule01.tar returned 2 instead of one of [0]
```

「すべてのホスト上に Capsule Server の新しい証明書をインストール」 に進みます。

4.7.6.5. すべてのホスト上に Capsule Server の新しい証明書をインストール

外部の Capsule サーバーに接続するホストにはサーバーのカスタム証明書が必要です。すべての Capsule サーバーのホストで以下のコマンドを実行します。

注記

Satellite Server のホスト名 **ではなく**、Capsule サーバーのホスト名を使用します。

```
# yum -y localinstall http://capsule.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

4.7.7. mongod へのアクセスの制限

データ損失の危険を減らすために、MongoDB データベースデーモン **mongod** へのアクセスは **apache** ユーザーと **root** ユーザーにだけ許可する必要があります。

Satellite Server と Capsule Server で **mongod** へのアクセスを制限するには、以下のコマンドを使用します。

Red Hat Enterprise Linux 6 でのファイアウォールの設定

1. Satellite Server と Capsule Server で **iptables** サービスを設定します。

```
# iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner --
uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -m owner --
uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 27017 -j DROP
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner --
uid-owner apache -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -m owner --
uid-owner root -j ACCEPT \
&& iptables -I OUTPUT -o lo -p tcp -m tcp --dport 28017 -j DROP
service iptables save
```

Red Hat Enterprise Linux 7 でのファイアウォールの設定

1. Satellite Server と Capsule Server でファイアウォールを設定します。

```
# firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 27017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p
```

```
tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner apache -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 28017 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd  --direct --add-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd  --direct --add-rule ipv6 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 28017 -j DROP
```

2. **--permanent** オプションを追加してコマンドを繰り返し、設定を永続化します。

```
# firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 27017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 27017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner apache -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 0
-o lo -p tcp -m tcp --dport 28017 -m owner --uid-owner root -j
ACCEPT \
&& firewall-cmd --permanent --direct --add-rule ipv4 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 28017 -j DROP \
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1
-o lo -p tcp -m tcp --dport 28017 -j DROP
```

第5章 外部サービスの設定

一部の環境には DNS、DHCP、および TFTP サービスがすでに存在するため、これらのサービスを提供するために Satellite Server を使用する必要はありません。DNS、DHCP、または TFTP を提供するために外部サーバーを使用する場合は、Satellite Server で使用するよう設定できます。

これらのサービスを手動で管理するために Satellite でサービスを無効にする場合は、詳細について「管理対象外ネットワークに対して DNS、DHCP、および TFTP を無効化」を参照してください。

5.1. 外部 DNS を使用した SATELLITE の設定

DNS サービスを提供するために Satellite が外部サーバーを使用するよう設定できます。

1. Red Hat Enterprise Linux Server をデプロイし、ISC DNS サービスをインストールします。

```
# yum install bind bind-utils
```

2. ドメインの設定を作成します。
以下の例では、ドメイン **virtual.lan** を 1 つのサブネット 192.168.38.0/24 として設定し、**foreman** という名前のセキュリティーキーを設定して、フォワーダーを Google のパブリック DNS アドレス (8.8.8.8 および 8.8.4.4) に設定します。

```
# cat /etc/named.conf
include "/etc/rndc.key";

controls  {
    inet 192.168.38.2 port 953 allow { 192.168.38.1; 192.168.38.2; }
keys { "capsule"; };
};

options  {
    directory "/var/named";
    forwarders { 8.8.8.8; 8.8.4.4; };
};

include "/etc/named.rfc1912.zones";

zone "38.168.192.in-addr.arpa" IN {
    type master;
    file "dynamic/38.168.192-rev";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};

zone "virtual.lan" IN {
    type master;
    file "dynamic/virtual.lan";
    update-policy {
        grant "capsule" zonesub ANY;
    };
};
```

inet 行は、設定ファイルで 1 つの行として入力する必要があります。

3. キーファイルを作成します。

```
# ddns-confgen -k capsule
```

このコマンドが完了するにはしばらく時間がかかることがあります。

4. キーセクションから出力をコピーし、**/etc/rndc.key** という名前の別のファイルに貼り付けます。

```
# cat /etc/rndc.key
key "capsule" {
        algorithm hmac-sha256;
        secret "GeBbgGoLedEAAwNQPtPh3zP56MJbkwM84UJDtaUS9mw=";
};
```

重要

これは、DNS サーバー設定を変更するために使用するキーです。root ユーザーのみが読み書きできるようにする必要があります。

5. ゾーンファイルを作成します。

```
# cat /var/named/dynamic/virtual.lan
$ORIGIN .
$TTL 10800      ; 3 hours
virtual.lan             IN SOA  service.virtual.lan.
root.virtual.lan. (
                                9          ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                3600       ; minimum (1 hour)
                                )
                        NS      service.virtual.lan.
$ORIGIN virtual.lan.
$TTL 86400      ; 1 day
capsule                 A       192.168.38.1
service                 A       192.168.38.2
```

6. 逆引きゾーンファイルを作成します。

```
# cat /var/named/dynamic/38.168.192-rev
$ORIGIN .
$TTL 10800      ; 3 hours
38.168.192.in-addr.arpa IN SOA  service.virtual.lan.
root.38.168.192.in-addr.arpa. (
                                4          ; serial
                                86400      ; refresh (1 day)
                                3600       ; retry (1 hour)
                                604800     ; expire (1 week)
                                3600       ; minimum (1 hour)
                                )
                        NS      service.virtual.lan.
$ORIGIN 38.168.192.in-addr.arpa.
```

```
$TTL 86400      ; 1 day
1                PTR     capsule.virtual.lan.
2                PTR     service.virtual.lan.
```

他の ASCII 以外の文字は使用しないでください。

5.2. DNS サービスの開始と起動

1. 構文を検証します。

```
# named-checkconf -z /etc/named.conf
```

2. サーバーを起動します。

使用しているプラットフォーム	実行するコマンド
Red Hat Enterprise Linux 7	# systemctl restart named
Red Hat Enterprise Linux 6	# service named restart

3. 新しいホストを追加します。
以下のコマンドでは、ホストの例 192.168.38.2 を使用しています。この値は、ご使用の環境に合わせて変更してください。

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

4. DNS サービスが新しいホストを解決できることを確認します。

```
# nslookup aaa.virtual.lan 192.168.38.2
```

5. 必要な場合は、新しいエントリーを削除します。

```
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

6. DNS サービスへの外部アクセスのためにファイアウォールを設定します (ポート 53 上の UDP および TCP)。

- Red Hat Enterprise Linux 7 で実行されている Satellite Server の場合:

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
&& firewall-cmd --permanent --add-port="53/udp" --add-
port="53/tcp"
```

- Red Hat Enterprise Linux 6 で実行されている Satellite Server の場合:

```
# iptables -I INPUT -m state --state NEW -p udp --dport 53 -j
ACCEPT \
&& iptables -I INPUT -m state --state NEW -p tcp --dport 53 -j
```

```
ACCEPT \
&& service iptables save
```

iptables サービスが起動され、有効であることを確認します。

```
# service iptables start
# chkconfig iptables on
```

5.3. CAPSULE SERVER での外部 DNS の設定

1. Red Hat Enterprise Linux Server で、ISC DNS サービスをインストールします。

```
# yum install bind bind-utils
```

nsupdate ユーティリティーがインストールされていることを確認します。Capsule は **nsupdate** ユーティリティーを使用してリモートサーバー上の DNS レコードを更新します。

2. サービスサーバーの **/etc/rndc.key** ファイルを Capsule Server にコピーします。

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

3. キーファイルに適切な所有者、パーミッション、および SELinux ラベルが設定されていることを確認します。

```
# ls /etc/rndc.key -Zla
-rw-r-----. root named system_u:object_r:dnssec_t:s0
/etc/rndc.key
```

4. ホストをリモートで追加して **nsupdate** ユーティリティーをテストします。

```
# echo -e "server 192.168.38.2\n \
update add aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
# nslookup aaa.virtual.lan 192.168.38.2
# echo -e "server 192.168.38.2\n \
update delete aaa.virtual.lan 3600 IN A 192.168.38.10\n \
send\n" | nsupdate -k /etc/rndc.key
```

5. **satellite-installer** スクリプトを実行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="192.168.38.2"
--foreman-proxy-keyfile=/etc/rndc.key
--foreman-proxy-dns-ttl=86400
```

6. foreman-proxy サービスを再起動します。

使用しているプラットフォーム	実行するコマンド
Red Hat Enterprise Linux 7	systemctl restart foreman-proxy
Red Hat Enterprise Linux 6	service foreman-proxy restart

7. Satellite Server Web インターフェースにログインします。
8. **Infrastructure (インフラストラクチャー)** > **Capsules** に移動します。適切な Capsule Server を見つけ、**Actions (アクション)** ドロップダウンリストから **Refresh (更新)** を選択します。この結果、DNS 機能が現れます。
9. 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。

5.4. SATELLITE SERVER での外部 DHCP の設定

重要

Satellite 6.3 以降は、NFS 経由の外部 DHCP 設定はサポートされません。inotify の最適化のため、DHCP Capsule はリモートファイルの変更を検出できなくなります。

1. Red Hat Enterprise Linux Server をデプロイし、ISC DHCP サービスをインストールします。

```
# yum install dhcp
```

2. 空のディレクトリーで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
```

3. テストまたは概念実証のためのデプロイメントの場合は、安全でない非ブロックデバイスコマンドを実行します。

```
# dnssec-keygen -r /dev/urandom -a HMAC-MD5 -b 512 -n HOST omapi_key
```

これにより、キーペアが現在のディレクトリーに 2 つのファイルで作成されます。

4. キーからシークレットハッシュをコピーします。

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

5. すべてのサブネットに対して **dhcpd** 設定ファイルを編集し、キーを追加します。

```
# 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 "jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

6. 2 つのキーファイルを、それらを作成したディレクトリーから削除します。

7. Satellite Server で各サブネットを定義します。
競合を回避するために、リース範囲と予約範囲は別々に設定することが推奨されます。たとえば、リース範囲は 192.168.38.10 から 192.168.38.100 となり、予約範囲 (Satellite Web UI で定義済み) は 192.168.38.101 から 192.168.38.250 となります。定義されたサブネットに対して DHCP Capsule はまだ設定しないでください。

   ISC DHCP は、定義されたサブネットに一致するインターフェースのみをリッスンします。この例では、サーバーには、192.168.38.0 サブネットに直接ルーティングするインターフェースが搭載されています。

8. ファイアウォールで DHCP サーバーへの外部アクセスを設定します。

   - Red Hat Enterprise Linux 7 で実行されている Satellite Server の場合:

     ```
     # firewall-cmd --add-service dhcp \
     && firewall-cmd --permanent --add-service dhcp
     ```

   - Red Hat Enterprise Linux 6 で実行されている Satellite Server の場合:

     ```
     # iptables -I INPUT -m state --state NEW -p tcp --dport 67 -j
     ACCEPT \
     && service iptables save
     ```

     iptables サービスが起動され、有効であることを確認します。

     ```
     # service iptables start
     # chkconfig iptables on
     ```

9. Capsule Server 上の foreman-proxy ユーザーの UID 番号と GID 番号を決定します。DHCP サーバーのと同じユーザーおよびグループを同じ ID で作成します。

   ```
   # groupadd -g 990 foreman-proxy
   # useradd -u 992 -g 990 -s /sbin/nologin foreman-proxy
   ```

10. 設定ファイルを読み取り可能にするために、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
    ```

11. DHCP サービスを起動します。

使用しているプラットフォーム	実行するコマンド
Red Hat Enterprise Linux 7	systemctl start dhcpd
Red Hat Enterprise Linux 6	service dhcpd start

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

```
# yum install nfs-utils
# systemctl enable rpcbind nfs-server
# systemctl start rpcbind nfs-server nfs-lock nfs-idmapd
```

13. NFS を使用して、エクスポートする DHCP 設定およびリースファイルを作成します。

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

14. 新しく作成されたマウントポイントを /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
```

15. /etc/fstab のファイルシステムをマウントします。

```
# mount -a
```

16. /etc/exports に以下の行が存在す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)
```

17. NFS サーバーをリロードします。

```
# exportfs -rva
```

18. ファイアウォールで Capsule Server 向けの DHCP omapi ポート 7911 を設定します。

- Red Hat Enterprise Linux 7 で以下のコマンドを実行します。

```
# firewall-cmd --add-port="7911/tcp" \
&& firewall-cmd --permanent --add-port="7911/tcp"
```

- Red Hat Enterprise Linux 6 で以下のコマンドを実行します。

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 7911 -j
ACCEPT \
&& service iptables save
```

iptables サービスが起動され、有効であることを確認します。

```
# service iptables start
# chkconfig iptables on
```

19. 必要な場合は、ファイアウォールで NFS への外部アクセスを設定します。
クライアントは NFSv3 を使用して設定されます。

- Red Hat Enterprise Linux 7 で、**firewalld** デーモンの NFS サービスを使用してファイアウォールを設定します。

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

- Red Hat Enterprise Linux 6 の場合は、**/etc/sysconfig/nfs** ファイルで NFSv3 向けのポートを設定します。

```
LOCKD_TCPPORT=32803
LOCKD_UDPPORT=32769
MOUNTD_PORT=892
RQUOTAD_PORT=875
STATD_PORT=662
STATD_OUTGOING_PORT=2020
```

サービスを再起動します。

```
# service nfs restart
```

ルールを **/etc/sysconfig/iptables** ファイルに追加します。

```
# iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p udp
--dport 111 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 111 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 2049 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 2049 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 32803 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 32769 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 892 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
```

```
tcp --dport 892 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 875 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 875 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
udp --dport 662 -j ACCEPT \
&& iptables -I INPUT -s 192.168.1.0/24 -m state --state NEW -p
tcp --dport 662 -j ACCEPT \
&& service iptables save
```

ファイアウォールを再起動します。

```
# service iptables restart
```

Red Hat Enterprise Linux 6 においてファイアウォールの背後で NFSv3 を使用する詳細については、『**Red Hat Enterprise Linux 6 Storage Administration Guide**』と、『**Red Hat Enterprise Linux 6 Security Guide**』の「Running NFS Behind a Firewall」という名前の節を参照してください。

5.5. CAPSULE SERVER での外部 DHCP の設定

1. NFS クライアントをインストールします。

```
# yum 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 RPC 通信パスを検証します。

```
# showmount -e 192.168.38.2
# rpcinfo -p 192.168.38.2
```

5. **/etc/fstab** ファイルに以下の行を追加します。

```
192.168.38.2:/exports/etc/dhcp /mnt/nfs/etc/dhcp nfs
ro,vers=3,auto,nosharecache,context="system_u:object_r:dhcp_etc_t:s0
" 0 0
```

```
192.168.38.2:/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** 上のファイルシステムをマウントします。

```
# mount -a
```

7. 関連するファイルを読み取ります。

```
# su foreman-proxy -s /bin/bash
bash-4.2$ cat /mnt/nfs/etc/dhcp/dhcpd.conf
bash-4.2$ cat /mnt/nfs/var/lib/dhcpd/dhcpd.leases
bash-4.2$ exit
```

8. **satellite-installer** スクリプトを実行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 --foreman-proxy-dhcp=true
--foreman-proxy-dhcp-provider=isc
--foreman-proxy-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf
--foreman-proxy-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases
--foreman-proxy-dhcp-key-name=omapi_key
--foreman-proxy-dhcp-key-
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==
--foreman-proxy-dhcp-server dhcp.example.com
```

9. foreman-proxy サービスを再起動します。

使用しているプラットフォーム	実行するコマンド
Red Hat Enterprise Linux 7	systemctl restart foreman-proxy
Red Hat Enterprise Linux 6	service foreman-proxy restart

10. Satellite Server Web インターフェースにログインします。

11. **Infrastructure (インフラストラクチャー)** > **Capsules** に移動します。適切な Capsule Server を見つけ、**Actions (アクション)** ドロップダウンリストから **Refresh (更新)** を選択します。この結果、DHCP 機能が現れます。

12. 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。

5.6. SATELLITE SERVER での外部 TFTP の設定

作業を開始する前に

- NFS が設定され、NFS への外部アクセスのためにファイアウォールが設定されている必要があります。「Satellite Server での外部 DHCP の設定」を参照してください。

Satellite Server での外部 TFTP の設定

1. TFTP サーバーをインストールし、有効にします。

```
# yum install tftp-server syslinux
```

   a. Red Hat Enterprise 7 で、**tftp.socket** ユニットを有効にし、アクティベートします。

```
# systemctl enable tftp.socket
# systemctl start tftp.socket
```

b. Red Hat Enterprise Linux 6 で、**xinetd** サービスを有効にし、起動します。

```
# service xinetd enable
# service xinetd start
```

2. PXELinux 環境を設定します。

```
# mkdir -p /var/lib/tftpboot/{boot,pxelinux.cfg}
# cp /usr/share/syslinux/{pxelinux.0,menu.c32,chain.c32}
/var/lib/tftpboot/
```

3. SELinux ファイルコンテキストを復元します。

```
# restorecon -RvF /var/lib/tftpboot/
```

4. NFS を使用してエクスポートする TFTP ディレクトリーを作成します。

```
# mkdir -p /exports/var/lib/tftpboot
```

5. 新しく作成されたマウントポイントを /etc/fstab ファイルに追加します。

```
/var/lib/tftpboot /exports/var/lib/tftpboot none bind,auto 0 0
```

6. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

7. **/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/var/lib/tftpboot
192.168.38.1(rw,async,no_root_squash,no_subtree_check,nohide)
```

最初の行は DHCP 設定に共通であり、このシステムで以前の手順が完了した場合はすでに存在します。

8. NFS サーバーをリロードします。

```
# exportfs -rva
```

5.6.1. ファイアウォールでの TFTP への外部アクセスの設定

ファイアウォールでの TFTP サービスへの外部アクセスの設定 (Red Hat Enterprise Linux 7 を使用する場合)

1. ファイアウォールを設定します (ポート 69 上の UDP)。

```
# firewall-cmd --add-port="69/udp" \
&& firewall-cmd --permanent --add-port="69/udp"
```

ファイアウォールでの TFTP サービスへの外部アクセスの設定 (Red Hat Enterprise Linux 6 を使用する場合)

1. ファイヤーウォールを設定します。

```
# iptables -I INPUT -m state --state NEW -p tcp --dport 69 -j ACCEPT
\
&& service iptables save
```

2. iptables サービスが起動され、有効であることを確認します。

```
# service iptables start
# chkconfig iptables on
```

5.7. CAPSULE SERVER での外部 TFTP の設定

1. NFS を準備するために 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。

```
192.168.38.2:/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** スクリプトを実行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=true
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot
```

5. TFTP サービスが DHCP サービスとは異なるサーバーで実行されている場合は、**tftp_servername** 設定をそのサーバーの FQDN または IP アドレスで更新します。

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

この結果、すべての設定ファイルが新しい値で更新されます。

6. Satellite Server Web インターフェースにログインします。

7. **Infrastructure (インフラストラクチャー)** > **Capsules** に移動します。適切な Capsule Server を見つけ、**Actions (アクション)** ドロップダウンリストから **Refresh (更新)** を選択します。この結果、TFTP 機能が現れます。

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。

5.8. SATELLITE での外部 IPA DNS の設定

Red Hat Satellite は IPA サーバーを使用して DNS サービスを提供するよう設定できます。この方法では、BIND 設定ファイルを編集し、**rndc.key** を Satellite のベースシステムにコピーするために IPA サーバーへの root アクセスが必要です。

この例では、Satellite Server の設定は以下のようになります。

IP アドレス	**192.168.25.1**
ホスト名	**satellite.example.com**

IPA マスターサーバーの設定は以下のようになります。

ホスト名	**idm1.example.com**
IP アドレス	**192.168.25.2**
ドメイン名	**example.com**

5.8.1. IPA サーバー上

ファイアウォールの設定

1. ファイアウォールで UDP ポートを設定します。

```
# firewall-cmd --add-port="53/udp" \
--add-port="88/udp" --add-port="464/udp" \
--add-port="123/udp" \
&& firewall-cmd --permanent --add-port="53/udp" \
--add-port="88/udp" --add-port="464/udp" \
--add-port="123/udp"
```

2. ファイアウォールで TCP ポートを設定します。

```
# firewall-cmd --add-port="53/tcp" \
 --add-port="80/tcp" --add-port="434/tcp" \
 --add-port="389/tcp" --add-port="636/tcp" \
--add-port="88/tcp" --add-port="464/tcp" \
&& firewall-cmd --permanent --add-port="53/tcp" \
 --add-port="80/tcp" --add-port="434/tcp" \
 --add-port="389/tcp" --add-port="636/tcp" \
--add-port="88/tcp" --add-port="464/tcp"
```

IPA サーバーの DNS ゾーンに対する外部アップデートの有効化

1. 以下の内容を **/etc/named.conf** ファイルの先頭に追加します。

```
// This was added to allow Satellite Server at 192.168.25.1 to make
DNS updates.
################################################################################
########
```

```
include "/etc/rndc.key";
controls  {
inet 192.168.25.2 port 953 allow { 192.168.25.1; } keys { "rndc-
key"; };
};
############################################################################
########
```

2. IPA Web UI で、**Network Services (ネットワークサービス)** > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。ゾーンの名前を選択します。**Settings (設定)** タブで、以下の手順を実行します。

   a. **BIND update policy (BIND アップデートポリシー)** ボックスで以下の内容を追加します。

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

   b. **Dynamic update** が **True** に設定されていることを確認します。

   c. **Update (更新)** をクリックして変更を保存します。

3. あとで使用するために **/etc/rndc.key** ファイルを IPA サーバーから安全な場所にコピーします。または、以下のようにこのファイルを Satellite のベースシステムに直接コピーします。

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

5.8.2. Satellite Server 上

1. Satellite Server が外部 DNS サーバーを使用するよう設定します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="192.168.25.2" \
--foreman-proxy-keyfile=/etc/rndc.key \
--foreman-proxy-dns-ttl=86400
```

2. テストのために **nsupdate** とともに **bind-utils** をインストールします。

```
# yum install bind-utils
```

3. **/etc/rndc.key** ファイルのキーが IPA サーバーで使用されているものと同じであることを確認します。

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

4. ホスト向けのテスト DNS エントリーを作成します (たとえば、**192.168.25.1** の IPA サーバー上に **192.168.25.20** の A レコードがあるホスト **test.example.com**)。

```
# 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
```

5. 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
```

6. IPA Web UI でエントリーを参照するために、**Network Services (ネットワークサービス)** > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。ゾーンの名前を選択し、名前でホストを検索します。

7. 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
```

8. 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た場合は、上記の **nslookup** コマンドが失敗し、SERVFAIL エラーメッセージが出力されます。

第6章 SATELLITE SERVER と CAPSULE SERVER のアップグレード

アップグレードは、Satellite および Capsule Server インストールをあるリリースから次のリリース (たとえば、Satellite 6.1 から Satellite 6.2) に移行するプロセスです。通常、アップグレードは、重要な新機能を利用するために行われます。アップグレードにはインストールされたコードの破棄が関係することがあるため、非常に長い時間が必要になることがあります。アップグレードを実行するときは、運用環境への影響を回避するためにワークフローを計画する必要があります。アップグレードを行う前は、競合を回避するために『Red Hat Satellite Release Notes』を参照してください。

Satellite Server と Capsule Server は別々にアップグレードされます。Satellite Server を最初にアップグレードし、次にすべての Capsule Server をアップグレードします。Satellite 6.1 Capsule Server は Satellite 6.2 と互換性がありませんが、リポジトリーを同期する前にアップグレードする必要があります。

また、Satellite Server と Capsule Server のアップグレード後に Satellite クライアントを新しいバージョンの `katello-agent` に手動でアップグレードする必要があります。詳細については、「Satellite クライアントのアップグレード」を参照してください。

重要

Red Hat Satellite Server と Capsule Server のバージョンは一致する必要があります。たとえば、Satellite 6.1 Satellite Server では 6.2 Capsule Server を実行できず、Satellite 6.2 サーバーでは 6.1 Capsule Server を実行できません。Satellite Server と Capsule Server のバージョンが一致しない場合、Capsule Server はサイレント状態で失敗します。

Satellite 6.2 向けのサポート対象アップグレードパス:

- Satellite 6.0.X GA から 6.1.X GA へのアップグレード。
- Satellite 6.1.9 GA 以降から 6.2.X GA 以降へのアップグレード。

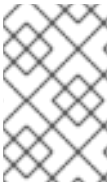
アップグレードは各バージョンから次のバージョンに行う必要があります。ベータから GA バージョンへのアップグレードはサポートされていません。

Satellite 6.2 のストレージ要件

Satellite のストレージ要件は以前のバージョンから変更されました。アップグレードする前に、ストレージの要件と推奨事項で詳しく説明されたストレージ要件を参照し、要件が満たされていることを確認します。

Satellite 6.2 の I/O 速度要件

Satellite または Capsule のリポジトリーのサイズに応じて、Satellite または Capsule 6.1 から 6.2 へのアップグレードには、長時間かかることがあります。`/var/lib/pulp/` ディレクトリーのコンテンツに対するアップグレード処理速度は 1 時間あたり 50 GB〜100 GB と推定されます。コンテンツが 500 GB である場合、アップグレードには 5〜10 時間かかることがあります。



注記

Pulp ディレクトリーが NFS デバイスに格納された場合は、アップグレードにさらに長い時間がかかります。たとえば、600 GB の Pulp ストレージでテストした場合は、NFS 経由での移行に 30 時間かかりました。

`/var/lib/pulp/` ディレクトリーのサイズを確認するには、以下のコマンドを入力します。

```
# df -h /var/lib/pulp
```

アップグレードにかかる時間に影響するため、I/O 速度をチェックすることが重要です。**hdparm** ツールでテストし、報告された **timing buffered disk reads** 要素を調べることにより、潜在的なパフォーマンスの問題を特定できます。**/var/lib/pulp/** ディレクトリーの場所がマウントされた場所をチェックする必要があります。この情報を使用すると、I/O 速度を調べることができます。

1. 最初に、I/O 速度を測定するために **hdparm** をインストールします。

   ```
   # yum install hdparm
   ```

2. **/var** ディレクトリーに関する情報を表示します。

   ```
   # df /var
   ```

   出力を調べ、**var** ディレクトリーで使用された論理デバイスを特定します。この例では、これは **/dev/mapper/rhel_vm37–118-root** です。

   ```
   Filesystem                      1K-blocks     Used Available Use%
   Mounted on
   /dev/mapper/rhel_vm37--118-root 200303044 41739160 158563884  21% /
   ```

3. 論理ボリュームに関する情報を表示します。

   ```
   # lvs -a -o +devices
   ```

   出力を調べ、論理ボリュームに関連付けられたデバイスを特定します。この例では、これは **/dev/vda2** にある **root** デバイスです。

   ```
   LV   VG            Attr       LSize   Pool Origin Data%  Meta%
   Move Log Cpy%Sync Convert Devices
   root rhel_vm37-118 -wi-ao---- 191.12g
   /dev/vda2(0)
   swap rhel_vm37-118 -wi-ao----   7.88g
   /dev/vda2(48926)
   ```

4. 関連するデバイスに対する I/O 速度を測定します。前の手順で特定したデバイスの場所を使用します。この例では、これは **/dev/vda2** です。結果は **timing buffered disk reads** 要素下に報告されます。

   ```
   # hdparm -tT /dev/vda2
   ```

   /dev/vda2 は、**/var/lib/pulp/** ディレクトリーがマウントされているシステムの場所によって異なります。したがって、上記の手順に従って場所を適切に特定することが重要になります。

/var パーティションで使用されたデバイスに対する読み取りアクセスには、最低でも 80 MB/s **Buffered Read Time**が推奨されます。スループットがこれよりも低い場合は、Satellite のインストール、アップグレード、および日々の運用で重大なパフォーマンスの問題が発生することがあります。重大な状況では、**/var** パーティションに対する遅い I/O (10〜20 MB/s) により、Satellite 6.1 から 6.2 へのアップグレードに 24 時間以上かかることがありました。

アップグレードの進捗の追跡

アップグレード時間が長いため、コマンドシェルに連続して接続せずにアップグレード進捗を確認できるよう、通信セッションの中断および再接続を許可する **screen** などのユーティリティーの使用を検討してください。Red Hat ナレッジベースの記事「How do I use the screen command?」には、**screen** のインストールに関する説明が記載されています。また、詳細については、**screen** の man ページを参照してください。インストールコマンドが実行されているシェルへの接続が失われた場合は、**/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ことを判断できます。

Foreman フックの Report クラスの名前が ConfigReport に変更された

Satellite 6.2 では、Report クラスが ConfigReport に変更されました。つまり、Rails イベントによりトリガーされたフックは **/hooks/report/** ディレクトリーに格納されたスクリプトを探さなくなりました。アップグレードが正常に完了するまですべてのフックを削除する必要があります。アップグレードが完了し、Satellite が期待どおり動作していることを確認したら、Foreman フックを復元します。ディレクトリー **/usr/share/foreman/config/hooks/config_report/** を作成し、新しいディレクトリーに **after_create** や **before_create** などのフックを移動します。Rails イベントと Foreman フックについては、『Red Hat Satellite Server Administration Guide』を参照してください。

Satellite 6.2 での Docker サポート

Satellite 6.2 では、Docker のサポートがバージョン 1 からバージョン 2 にアップグレードされました。この変更により、Docker のデータモデルが変更され、Docker イメージのサポートが削除され、マニフェストのサポートが導入されました。この大きな変更の結果として、Docker バージョン 1 のサポートは完全に削除され、既存の Docker バージョン 1 のリポジトリーがアップグレードの一部として削除されます。

以前に作成されたコンテナーは、Satellite Server に引き続き表示され、起動できます。ただし、イメージは存在しないため、新しいコンテナーを作成することはできません。

アップグレード後に Docker バージョン 2 のリポジトリーを作成することを支援するために、アップグレード前に Docker バージョン 1 のリポジトリーの詳細を取得し、保存します。

これは、Satellite Web UI または Hammer CLI を使用して実現できます。

Satellite Web UI を使用した既存の Docker リポジトリーの表示方法:

1. **Content (コンテンツ)** > **Products (製品)** に移動します。
2. リポジトリーを選択して設定を表示します。

CLI を使用した既存の Docker リポジトリーの表示方法:

1. 特定の組織に対するすべての Docker リポジトリーをリストします。

```
# hammer repository list --organization-id 1 --content-type docker
```

2. 特定のリポジトリーの Docker リポジトリー情報を取得します。

```
# hammer repository info --id 3
```

アップグレードが完了し、上記のリポジトリーが Docker バージョン 2 のレジストリーから利用可能な場合は、適切な詳細 (name、docker-upstream-name、URL など) でこれらのリポジトリーを作成できます。Docker バージョン 2 のコンテンツを管理するプロセスは、Docker バージョン 1 と他のコンテンツ

タイプを使用した Satellite 6.1 のプロセスに類似しています。リポジトリーを作成および同期し、コンテンツビューを作成、公開、およびプロモートします。

6.1. SATELLITE SERVER 6.2 へのアップグレード

6.2. 接続された SATELLITE SERVER のアップグレード

作業を開始する前に

- Red Hat Satellite Server 6.1 のマイナーバージョン 6.1.9 以降にアップグレードされている必要があります。最低要件は 6.1.9 です。Red Hat Satellite 6.2 にアップグレードする場合は、6.1.9 よりも大きいマイナーバージョンにアップグレードする必要がありません。それよりも前のマイナーバージョンからの直接アップグレードはサポートされていません。詳細については、『Red Hat Satellite 6.1 Installation Guide の Upgrading Between Minor Versions of Satellite』を参照してください。
- Satellite Server をアップグレードする前に、ファイアウォールの設定を確認し、更新します。追加情報については、「ポートとファイアウォールの要件」を参照してください。
- マニフェストはカスタマーポータルまたは Satellite Web UI で**削除しないでください**。削除すると、すべてのコンテンツホストが登録解除されます。
- アップグレードする前に、すべての Foreman フックをバックアップし、削除します。フックは、アップグレードの完了後に Satellite が動作しているのを確認してから、戻してください。

警告

`root/ssl-build` ディレクトリーと、カスタム証明書に関連付けられたソースファイルを作成したディレクトリー両方の内容を保持する必要があります。カスタム証明書を実装する場合であっても、アップグレードの実行時は `/root/ssl-build` ディレクトリーの内容を保持する必要があります。アップグレード中にこれらのファイルを保持しないと、アップグレードに失敗します。アップグレードが続行するために、これらのファイルは、削除された場合に、バックアップから復元する必要があります。

Satellite Server のアップグレード

1. バックアップを作成します。
   - 仮想マシンで、スナップショットを取得します。
   - 物理マシンで、バックアップを作成します。
2. アップグレード前スクリプトは競合を検出し、アップグレード後に登録解除および削除できる Satellite Server の重複エントリーがあるホストをリストできます。また、組織に割り当てられていないホストを検出します。**Hosts** > **All hosts** を選択して組織の関連付けがないホストがリストされ、同じ名前のコンテンツホストに組織がすでに関連付けられている場合、コンテンツホストは自動的に登録解除されます。これは、アップグレード前にこのようなホストを組織に関連付けることによって回避できます。

アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。

a. アップグレード前スクリプトを使用するには、**ruby193-rubygem-katello-2.2.0.90-1-sat** 以降が必要です。

```
# yum update ruby193-rubygem-katello
```

b. アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。

```
# foreman-rake katello:upgrade_check
```

アップグレードチェックで、タスクが実行中であることが原因の障害が報告された場合は、タスクが完了するまで待機することが推奨されます。一部のタスクはキャンセルすることができますが、Red Hat ナレッジベースソリューション How to manage paused tasks on Red Hat Satellite 6 のアドバイスに従って、安全にキャンセルできるタスクと安全にキャンセルできなタスクについて理解する必要があります。

3. DNS と DHCP の設定ファイルである **/etc/zones.conf** と **/etc/dhcp/dhcpd.conf** をバックアップします。インストーラーでは 1 つのドメインまたはサブネットしかサポートされないため、これらのバックアップから変更を復元する必要がある場合があります。

4. DNS または DHCP の設定ファイルを手動で編集し、変更を上書きしたくない場合は、以下のコマンドを実行します。

```
# katello-installer --capsule-dns-managed=false --capsule-dhcp-
managed=false
```

5. Red Hat Satellite 6.1 向けリポジトリーを無効にします。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --disable rhel-6-server-satellite-
6.1-rpms
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --disable rhel-7-server-satellite-
6.1-rpms
```

6. 必要な場合は、Satellite 6.1 リポジトリーが無効であることを確認するために、以下のようなコマンドを入力します。

```
# subscription-manager repos --list-enabled
```

7. Red Hat Satellite 6.2 向けリポジトリーを有効にします。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable=rhel-6-server-rpms \
--enable=rhel-server-rhscl-6-rpms \
--enable=rhel-6-server-satellite-6.2-rpms
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-satellite-6.2-rpms
```

8. Red Hat 以外の yum リポジトリーから残されたすべてのメタデータを消去します。

```
# yum clean all
```

9. リポジトリーが有効になっていることを確認します。

```
# yum repolist enabled
```

以下のような出力が表示されます。

```
Loaded plugins: product-id, subscription-manager
repo id                                          repo name
status
!rhel-7-server-rpms/x86_64                       Red Hat Enterprise
Linux 7 Server (RPMs)                                      9,889
!rhel-7-server-satellite-6.2-rpms/x86_64         Red Hat Satellite 6.2
(for RHEL 7 Server) (RPMs)                               545
!rhel-server-rhscl-7-rpms/x86_64                 Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server     4,279
repolist: 14,713
```

10. Satellite Web UI で、**Hosts (ホスト)** > **Discovered hosts (検出されたホスト)** に移動します。検出されたホストが利用可能な場合は、それらを無効にし、**Discovered hosts (検出されたホスト)** ページ下のすべてのエントリーを削除します。組織設定メニューを使用して他のすべての組織を順番に選択し、必要に応じてこのアクションを繰り返します。アップグレードが完了したら、これらのホストを再起動します。

11. すべての外部 Capsule Server が組織に割り当てられていることを確認します。割り当てられていない場合、これらのサーバーは、ホスト統合の変更により登録解除された可能性があります。

12. Satellite Web UI でリポジトリーを設定します。

    a. Satellite Web UI で、**Content** > **Red Hat Repositories** に移動し、**RPM** タブを選択します。

    b. Red Hat Enterprise Linux Server **製品** を見つけ、展開します。

    c. Red Hat Satellite Tools 6.2 (RHEL X Server 用) (RPM) を見つけ展開します。

    d. RHEL X Server RPMs x86_64 用 Red Hat Satellite Tools 6.2 を選択します。

13. 新しく有効になったリポジトリーを同期します。

a. Satellite Web UI で、**Content (コンテンツ)** > **Sync Status (同意ステータス)** に移動します。

b. 製品の横にある矢印をクリックして利用可能なリポジトリーを表示します。

c. 6.2 用リポジトリーを選択します。

d. **Synchronize Now** をクリックします。
Satellite Tools リポジトリーを更新しようとするときにエラーが発生した場合は、カスタマーポータルまたは Satellite Web UI でマニフェストを**削除しないでください**。削除すると、すべてのコンテンツホストが登録解除されます。詳細については、Red Hat ナレッジベースソリューション「Cannot enable Red Hat Satellite Tools Repo on Satellite 6.2」を参照してください。

14. 6.1 バージョンリポジトリーを使用する既存のコンテンツビューを 6.2 向けの新しいバージョンで更新します。新しい 6.2 バージョンリポジトリーがあるコンテンツビューの更新されたバージョンを公開し、プロモートします。

15. リポジトリーキャッシュを削除します。

```
# yum clean all
```

16. アップグレードチェックを再び実行して、すでに行った手順により、アップグレード中にタスクが停止する状況になっていないことを確認します。

```
# foreman-rake katello:upgrade_check
```

17. Katello サービスを停止します。

```
# katello-service stop
```

18. すべてのパッケージを更新します。

```
# yum update
```

19. カスタム Apache サーバー設定がある場合は、次の手順でインストールデフォルト値に戻ります。**アップグレードの実行時に**変更される内容を確認する場合は、**--noop** (no operation) オプションとともにアップグレードコマンドを入力し、次の手順でアップグレードコマンドを入力するときに適用される変更内容を確認できます。このテストを行わない場合は、次の手順に進みます。または、以下のように手順を続行します。

a. 次の行を **/etc/httpd/conf/httpd.conf** 設定ファイルに追加します。

```
Include /etc/httpd/conf.modules.d/*.conf
```

b. **httpd** サービスを再起動します。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service httpd restart
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl restart httpd
```

c. **postgresql** データベースサービスおよび **mongod** データベースサービスを起動します。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service postgresql start
# service mongod start
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl start postgresql
# systemctl start mongod
```

d. 以下のように **--noop** オプションとともにコマンドを入力します。

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

/var/log/foreman-installer/satellite.log を参照して、**--noop** オプションが省略された場合に**適用される変更**を確認します。設定ファイルの変更を示す **+++** と **---** の記号を探します。上記の "no operation" コマンドにより実際にはファイルは作成されず、モジュール内の一部の Puppet リソースではファイルがそこに存在することが期待されるため、いくつかのエラーメッセージが表示されるはずです。

e. Katello サービスを停止します。

```
# katello-service stop
```

20. **--upgrade** オプションを使用してインストーラースクリプトを実行することによりアップグレードを実行します。

```
# satellite-installer --scenario satellite --upgrade
```

警告

config サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再び実行します。

21. これまでに行ったバックアップを使用して DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。

22. 前の手順で変更を行った場合は、Katello サービスを再起動します。

```
# katello-service restart
```

23. Satellite Web UI で Discovery テンプレートを更新します。

    a. **Hosts (ホスト)** > **Provisioning templates (テンプレートのプロビジョニング)** に移動します。

    b. **PXELinux global default (PXELinux グローバルデフォルト)** を選択します。

    c. **Template editor (テンプレートエディター)** ダイアログボックスで、以下のテキストに一致するよう **LABEL discovery** で始まるスタンザを更新することにより **PXELinux global default (PXELinux グローバルデフォルト)** テンプレート検出メニューエントリーを編集します。

```
LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2
```

    - **`proxy.type`** オプションは **`proxy`** または **`foreman`** のいずれかになります。**`proxy`** の場合は、すべての通信が Capsule 経由で行われます。**`foreman`** の場合は、通信が直接 Satellite Server に行われます。

    - **`proxy.url`** には、Satellite Capsule または Server の URL を指定します。HTTP と HTTPS の両方のプロトコルがサポートされます。

24. OpenSCAP プラグインがインストールされており、デフォルトの OpenSCAP コンテンツが利用可能でない場合は、以下のコマンドを実行します。

```
# foreman-rake foreman_openscap:bulk_upload:default
```

25. Satellite Web UI で **Configure (設定)** > **Discovery Rules (検出ルール)** に移動し、選択された組織および場所を検出ルールに関連付けます。

6.3. 切断された SATELLITE SERVER のアップグレード

作業を開始する前に

- Red Hat Satellite Server 6.1 の最新マイナーリリースにアップグレードされている必要があります。それよりも前のマイナーバージョンからの直接アップグレードはサポートされていません。詳細については、『Red Hat Satellite 6.1 Installation Guide』の「Upgrading Between Minor Versions of Satellite」を参照してください。

- Satellite Server をアップグレードする前に、ファイアウォールの設定を確認し、更新します。追加情報については、「ポートとファイアウォールの要件」を参照してください。

- マニフェストはカスタマーポータルまたは Satellite Web UI で**削除しないでください**。削除すると、すべてのコンテンツホストが登録解除されます。

- アップグレードする前に、すべての Foreman フックをバックアップし、削除します。フックは、アップグレードの完了後に Satellite が動作しているのを確認してから、戻してください。

警告

root/ssl-build ディレクトリーと、カスタム証明書に関連付けられたソースファイルを作成したディレクトリー両方の内容を保持する必要があります。カスタム証明書を実装する場合であっても、アップグレードの実行時は **/root/ssl-build** ディレクトリーの内容を保持する必要があります。アップグレード中にこれらのファイルを保持しないと、アップグレードに失敗します。アップグレードが続行するために、これらのファイルは、削除された場合に、バックアップから復元する必要があります。

切断された Satellite Server のアップグレード

1. バックアップを作成します。
   - 仮想マシンで、スナップショットを取得します。
   - 物理マシンで、バックアップを作成します。
2. アップグレード前スクリプトは競合を検出し、アップグレード後に登録解除および削除できる Satellite Server の重複エントリーがあるホストをリストできます。また、組織に割り当てられていないホストを検出します。**Hosts** > **All hosts** を選択して組織の関連付けがないホストがリストされ、同じ名前のコンテンツホストに組織がすでに関連付けられている場合、コンテンツホストは自動的に登録解除されます。これは、アップグレード前にこのようなホストを組織に関連付けることによって回避できます。
   アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。
   a. アップグレード前スクリプトを使用するには、**ruby193-rubygem-katello-2.2.0.90-1-sat** 以降が必要です。

      ```
      # yum update ruby193-rubygem-katello
      ```

   b. アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。

      ```
      # foreman-rake katello:upgrade_check
      ```

      アップグレードチェックで、タスクが実行中であることが原因の障害が報告された場合は、タスクが完了するまで待機することが推奨されます。一部のタスクはキャンセルすることができますが、Red Hat ナレッジベースソリューション How to manage paused tasks on Red Hat Satellite 6 のアドバイスに従って、安全にキャンセルできるタスクと安全にキャンセルできなタスクについて理解する必要があります。

3. DNS と DHCP の設定ファイルである **/etc/zones.conf** と **/etc/dhcp/dhcpd.conf** をバックアップします。インストーラーでは 1 つのドメインまたはサブネットしかサポートされないため、これらのバックアップから変更を復元する必要がある場合があります。

4. DNS または DHCP の設定ファイルを手動で編集し、変更を上書きしたくない場合は、以下のコマンドを実行します。

```
# katello-installer --capsule-dns-managed=false --capsule-dhcp-
managed=false
```

5. Satellite Web UI で、**Hosts (ホスト)** > **Discovered hosts (検出されたホスト)** に移動します。検出されたホストが利用可能な場合は、それらを無効にし、**Discovered hosts (検出されたホスト)** ページ下のすべてのエントリーを削除します。組織設定メニューを使用して他のすべての組織を順番に選択し、必要に応じてこのアクションを繰り返します。アップグレードが完了したら、これらのホストを再起動します。

6. すべての外部 Capsule Server が組織に割り当てられていることを確認します。割り当てられていない場合、これらのサーバーは、ホスト統合の変更により登録解除された可能性があります。

7. Katello サービスを停止します。

```
# katello-service stop
```

8. ISO ファイルを取得してマウントし、パッケージをインストールします。詳細については、「切断されたネットワークからのダウンロードおよびインストール」を参照してください。

9. カスタム Apache サーバー設定がある場合は、次の手順でインストールデフォルト値に戻ります。**アップグレードの実行時に**変更される内容を確認する場合は、**--noop** (no operation) オプションとともにアップグレードコマンドを入力し、次の手順でアップグレードコマンドを入力するときに適用される変更内容を確認できます。このテストを行わない場合は、次の手順に進みます。または、以下のように手順を続行します。

   a. 次の行を **/etc/httpd/conf/httpd.conf** 設定ファイルに追加します。

```
Include /etc/httpd/conf.modules.d/*.conf
```

   b. **httpd** サービスを再起動します。

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service httpd restart
```

   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl restart httpd
```

   c. **postgresql** データベースサービスおよび **mongod** データベースサービスを起動します。

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service postgresql start
# service mongod start
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl start postgresql
# systemctl start mongod
```

d. 以下のように **--noop** オプションとともにコマンドを入力します。

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

/var/log/foreman-installer/satellite.log を参照して、**--noop** オプションが省略された場合に**適用される変更**を確認します。設定ファイルの変更を示す **+++** と **---** の記号を探します。上記の "no operation" コマンドにより実際にはファイルは作成されず、モジュール内の一部の Puppet リソースではファイルがそこに存在することが期待されるため、いくつかのエラーメッセージが表示されるはずです。

e. Katello サービスを停止します。

```
# katello-service stop
```

10. **--upgrade** オプションを使用してインストーラースクリプトを実行することによりアップグレードを実行します。

```
# satellite-installer --scenario satellite --upgrade
```

警告

config サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再び実行します。

11. これまでに行ったバックアップを使用して DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。

12. 前の手順で変更を行った場合は、Katello サービスを再起動します。

```
# katello-service restart
```

13. Satellite Web UI で Discovery テンプレートを更新します。

a. **Hosts (ホスト)** > **Provisioning templates (テンプレートのプロビジョニング)** に移動します。

b. **PXELinux global default (PXELinux グローバルデフォルト)** を選択します。

c. **Template editor (テンプレートエディター)** ダイアログボックスで、以下のテキストに一致するよう **LABEL discovery** で始まるスタンザを更新することにより **PXELinux global default (PXELinux グローバルデフォルト)** テンプレート検出メニューエントリーを編集します。

```
LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2
```

- **`proxy.type`** オプションは **`proxy`** または **`foreman`** のいずれかになります。**`proxy`** の場合は、すべての通信が Capsule 経由で行われます。**`foreman`** の場合は、通信が直接 Satellite Server に行われます。
- **`proxy.url`** には、Satellite Capsule または Server の URL を指定します。HTTP と HTTPS の両方のプロトコルがサポートされます。

14. OpenSCAP プラグインがインストールされており、デフォルトの OpenSCAP コンテンツが利用可能でない場合は、以下のコマンドを実行します。

```
# foreman-rake foreman_openscap:bulk_upload:default
```

15. Satellite Web UI で **Configure (設定)** > **Discovery Rules (検出ルール)** に移動し、選択された組織および場所を検出ルールに関連付けます。

6.4. CAPSULE SERVER のアップグレード

作業を開始する前に

- Capsule Server のアップグレード前に、Satellite Server がアップグレードされている必要があります。
- Capsule は、最低でも Red Hat Satellite Server 6.1 の 6.1.9 マイナーバージョン上にある必要があります。最低要件は 6.1.9 です。Red Hat Satellite 6.2 にアップグレードする場合は、6.1.9 よりも大きいマイナーバージョンにアップグレードする必要がありません。それよりも前のマイナーバージョンからの直接アップグレードはサポートされていません。詳細については、『Red Hat Satellite 6.1 Installation Guide』の「Upgrading Between Minor Versions of Satellite」を参照してください。
- 新しくアップグレードされた Satellite Server に Capsule のベースシステムが'登録されていることを確認します。
- 新しくアップグレードされた Satellite Server で Capsule の組織と場所の設定が適切であることを確認します。
- Capsule Server をアップグレードする前に、ファイアウォールの設定を確認し、更新します。追加情報については、「ポートとファイアウォールの要件」を参照してください。

警告

root/ssl-build ディレクトリーと、カスタム証明書に関連付けられたソースファイルを作成したディレクトリー両方の内容を保持する必要があります。カスタム証明書を実装する場合であっても、アップグレードの実行時は **/root/ssl-build** ディレクトリーの内容を保持する必要があります。アップグレード中にこれらのファイルを保持しないと、アップグレードに失敗します。アップグレードが続行するために、これらのファイルは、削除された場合に、バックアップから復元する必要があります。

Capsule Server のアップグレード

1. バックアップを作成します。
   - 仮想マシンで、スナップショットを取得します。
   - 物理マシンで、バックアップを作成します。
2. DNS と DHCP の設定ファイルである **/etc/zones.conf** と **/etc/dhcp/dhcpd.conf** をバックアップします。インストーラーでは 1 つのドメインまたはサブネットしかサポートされないため、これらのバックアップから変更を復元する必要がある場合があります。
3. DNS または DHCP の設定ファイルを手動で編集し、変更を上書きしたくない場合は、以下のコマンドを入力します。

```
# capsule-installer --foreman-proxy-dns-managed=false --foreman-
proxy-dhcp-managed=false
```

4. Red Hat Satellite 6.1 向けリポジトリーを無効にします。
   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --disable rhel-6-server-satellite-
capsule-6.1-rpms
```

   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --disable rhel-7-server-satellite-
capsule-6.1-rpms
```

5. 新しいリポジトリーを有効にします。
   Red Hat Software Collections リポジトリーは、リモート実行機能を含む一部の Red Hat Satellite 機能で必要な新しいバージョンの Ruby を提供します。
   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable rhel-7-server-satellite-
capsule-6.2-rpms --enable rhel-server-rhscl-7-rpms
```

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable rhel-6-server-satellite-
capsule-6.2-rpms --enable rhel-server-rhscl-6-rpms
```

6. Satellite Web UI で、**Hosts (ホスト)** > **Discovered hosts (検出されたホスト)** に移動します。検出されたホストが利用可能な場合は、それらを無効にし、**Discovered hosts (検出されたホスト)** ページ下のすべてのエントリーを削除します。組織設定メニューを使用して他のすべての組織を順番に選択し、必要に応じてこのアクションを繰り返します。アップグレードが完了したら、これらのホストを再起動します。

7. リポジトリーキャッシュを削除します。

```
# yum clean all
```

8. すべてのパッケージを更新します。

```
# yum update
```

9. Satellite Server で、新しい証明書を使用してアーカイブを生成します。

```
# capsule-certs-generate --capsule-fqdn "mycapsule.example.com" --
certs-tar "mycapsule.example.com-certs.tar"
```

mycapsule.example.com を Capsule Server の完全修飾ドメイン名に置き換える必要があります。

10. アーカイブファイルを Capsule Server にコピーします。

```
# scp mycapsule.example.com-certs.tar mycapsule.example.com
```

11. Capsule Server を検出済みホストのプロキシーとして使用する場合は、検出プラグインをインストールします。

```
# yum install rubygem-smart_proxy_discovery.noarch
```

12. Capsule Server で **foreman_url** 設定が正しいことを確認します。

```
# grep foreman_url /etc/foreman-proxy/settings.yml
```

Satellite Server の完全修飾ドメイン名が表示されます。

13. カスタム Apache サーバー設定がある場合は、次の手順でインストールデフォルト値に戻ります。**アップグレードの実行時に**変更される内容を確認する場合は、**--noop** (no operation) オプションとともにアップグレードコマンドを入力し、次の手順でアップグレードコマンドを入力するときに適用される変更内容を確認できます。このテストを行わない場合は、次の手順に進みます。または、以下のように手順を続行します。

    a. 次の行を **/etc/httpd/conf/httpd.conf** 設定ファイルに追加します。

```
Include /etc/httpd/conf.modules.d/*.conf
```

    b. **httpd** サービスを再起動します。

        - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service httpd restart
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl restart httpd
```

c. **mongod** データベースサービスを起動します。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service mongod start
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl start mongod
```

d. 以下のように **--noop** オプションとともにコマンドを入力します。

```
# satellite-installer --scenario capsule --upgrade --verbose --
noop
```

/var/log/foreman-installer/capsule.log を参照して、**--noop** オプションが省略された場合に**適用される変更**を確認します。設定ファイルの変更を示す **+++** と **---** の記号を探します。上記の "no operation" コマンドにより実際にはファイルは作成されず、モジュール内の一部の Puppet リソースではファイルがそこに存在することが期待されるため、いくつかのエラーメッセージが表示されるはずです。

e. Katello サービスを停止します。

```
# katello-service stop
```

14. **--upgrade** オプションを使用してインストーラースクリプトを実行してアップグレードを行い、Satellite Server で以前に作成された証明書アーカイブへのパスを指定します。

```
# satellite-installer --scenario capsule --upgrade \
  --certs-tar mycapsule.example.com-certs.tar
```

警告

config サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/capsule.yaml) was not found,
can not continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再び実行します。

15. これまでに行ったバックアップを使用して DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。

16. Satellite Server で foreman-discovery パッケージをアップグレードし、アップグレード前にシャットダウンされたホストを有効にします。

6.5. CAPSULE SERVER での DISCOVERY のアップグレード

1. 関連するすべてのパッケージが Satellite Server で最新であることを確認します。

```
# yum upgrade tfm-rubygem-foreman_discovery
```

2. 必要な場合は、Katello サービスを再起動します。

```
# katello-service restart
```

3. 検出されたホストとのプロビジョニングネットワークに接続された、または検出されたホストに TFTP サービスを提供するSatellite Capsule 上の Discovery イメージをアップグレードします。

```
# yum upgrade foreman-discovery-image
```

4. 同じインスタンスで、プロキシーサービスを提供するパッケージをインストールし、foreman-proxy サービスを再起動します。

```
# yum install rubygem-smart_proxy_discovery
# service foreman-proxy restart
```

5. Satellite Web UI で、**Infrastructure (インフラストラクチャー)** > **Capsules** に移動し、関連するプロキシーにより Discovery 機能がリストされていることを確認します。必要な場合は、**Refresh features (機能の更新)** をクリックします。

6. **Infrastructure (インフラストラクチャー)** > **Subnets (サブネット)** に移動し、検出を使用する各サブネットに必要なスマートプロキシーを選択して、検出プロキシーに接続されていることを確認します。

Satellite Web UI で Discovery テンプレートを更新します。

a. **Hosts (ホスト)** > **Provisioning Templates (テンプレートのプロビジョニング)** に移動します。

b. **PXELinux global default (PXELinux グローバルデフォルト)** を選択します。

c. **Template editor (テンプレートエディター)** ダイアログボックスで、以下のテキストに一致するよう **LABEL discovery** で始まるスタンザを更新することにより **PXELinux global default (PXELinux グローバルデフォルト)** テンプレート検出メニューエントリーを編集します。

```
LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
```

```
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2
```

- **proxy.type** オプションは **proxy** または **foreman** のいずれかになります。**proxy** の場合は、すべての通信が Capsule 経由で行われます。**foreman** の場合は、通信が直接 Satellite Server に行われます。
- **proxy.url** には、Satellite Capsule または Server の URL を指定します。HTTP と HTTPS の両方のプロトコルがサポートされます。
- proxy.url オプションを省略し、SRV レコードから Capsule DNS 名を決定することができます。これは、複数の検出サブネットが存在する場合に役に立ちます。詳細については、『Red Hat Satellite 6.2 Host Configuration Guide』を参照してください。

6.6. SATELLITE クライアントのアップグレード

クライアントと Satellite Server との互換性を保持するために、すべてのクライアントを **katello-agent** の新しいバージョンにアップグレードする必要があります。この場合は、Satellite Tools リポジトリーを 6.1 から 6.2 に変更する必要があります。この変更は、手動で行うか、**satellite-tools-upgrade** パッケージをインストールすることによって行うことができます。このパッケージには、Satellite Tools リポジトリーのバージョンを変更するインストール後スクリプトのみが含まれます。

作業を開始する前に

- Satellite Server がアップグレードされている必要があります。
- Satellite で新しい Satellite Tools リポジトリーが有効である必要があります。
- Satellite で新しいリポジトリーが同期されている必要があります。
- 以前にクライアントで **katello-agent** がインストールされていない場合は、手動で作業を行います。

警告

root/ssl-build ディレクトリーと、カスタム証明書に関連付けられたソースファイルを作成したディレクトリー両方の内容を保持する必要があります。カスタム証明書を実装する場合であっても、アップグレードの実行時は **/root/ssl-build** ディレクトリーの内容を保持する必要があります。アップグレード中にこれらのファイルを保持しないと、アップグレードに失敗します。アップグレードが続行するために、これらのファイルは、削除された場合に、バックアップから復元する必要があります。

satellite-tools-upgrade パッケージを使用した Satellite クライアントのアップグレード

1. Satellite Web UI で **Hosts (ホスト)** > **Content Hosts (コンテンツホスト)** または **Host Collections (ホストコレクション)** に移動し、アップグレードするコンテンツホストを選択します。

2. **Packages (パッケージ)** タブで、検索フィールドにパッケージ名 **`satellite-tools-upgrade`** を入力します。

   a. Content Host (コンテンツホスト) ビューを使用して単一のホストをアップグレードする場合は、**Perform (実行)** を選択してパッケージをインストールします。

   b. Bulk Actions (一括処理) ビューを使用してホストのコレクションをアップグレードする場合は、**Install (インストール)** を選択してパッケージをインストールします。

3. **Packages (パッケージ)** タブで、検索フィールドにパッケージ名 **`katello-agent`** を入力します。

   a. Content Host (コンテンツホスト) ビューを使用して単一のホストをアップグレードする場合は、**Package Update (パッケージの更新)** を選択し、**Perform (実行)** を使用してパッケージを更新します。

   b. Bulk Actions (一括処理) ビューを使用してホストのコレクションをアップグレードする場合は、**Update (更新)** を選択してパッケージを更新します。

注記

Red Hat Bugzilla 1291960 が解決されるまでは、Web UI または **hammer** CLI を使用して **`katello-agent`** のアップグレードを試行した後に、システムにインストールされた重複するパッケージバージョンが表示されます。詳細については、このバグを参照してください。

Satellite クライアントの手動アップグレード

1. クライアントシステムにログインします。
2. 以前のバージョンの Satellite 向けリポジトリーを無効にします。
   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

     ```
     # subscription-manager repos --disable rhel-7-server-satellite-tools-6.1-rpms
     ```

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

     ```
     # subscription-manager repos --disable rhel-6-server-satellite-tools-6.1-rpms
     ```

3. このバージョンの Satellite 向け Satellite Tools リポジトリーを有効にします。
   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

     ```
     # subscription-manager repos --enable=rhel-7-server-satellite-tools-6.2-rpms
     ```

   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

     ```
     # subscription-manager repos --enable=rhel-6-server-satellite-tools-6.2-rpms
     ```

4. Katello エージェントパッケージをアップグレードします。

```
# yum upgrade katello-agent
```

6.7. 自己登録 SATELLITE SERVER のアップグレード

Red Hat カスタマーポータルで自己登録 Satellite Server を同期し、コンテンツビューを公開およびプロモートし、自己登録 Satellite Server をアップグレードします。

作業を開始する前に

- 最低でも Red Hat Satellite Server 6.1 のマイナーバージョン 6.1.9 にアップグレードされている必要があります。最低要件は 6.1.9 です。Red Hat Satellite 6.2 にアップグレードする場合は、6.1.9 よりも大きいマイナーバージョンにアップグレードする必要がありません。それよりも前のマイナーバージョンからの直接アップグレードはサポートされていません。詳細については、『Red Hat Satellite 6.1 Installation Guide』の「Upgrading Between Minor Versions of Satellite」を参照してください。
- Satellite Server をアップグレードする前に、ファイアウォールの設定を確認し、更新します。追加情報については、「ポートとファイアウォールの要件」を参照してください。
- マニフェストはカスタマーポータルまたは Satellite Web UI で**削除しないでください**。削除すると、すべてのコンテンツホストが登録解除されます。
- アップグレードする前に、すべての Foreman フックをバックアップし、削除します。フックは、アップグレードの完了後に Satellite が動作しているのを確認してから、戻してください。

警告

root/ssl-build ディレクトリーと、カスタム証明書に関連付けられたソースファイルを作成したディレクトリー両方の内容を保持する必要があります。カスタム証明書を実装する場合であっても、アップグレードの実行時は **/root/ssl-build** ディレクトリーの内容を保持する必要があります。アップグレード中にこれらのファイルを保持しないと、アップグレードに失敗します。アップグレードが続行するために、これらのファイルは、削除された場合に、バックアップから復元する必要があります。

自己登録 Satellite Server のアップグレード

1. バックアップを作成します。
   - 仮想マシンで、スナップショットを取得します。
   - 物理マシンで、バックアップを作成します。
2. アップグレード前スクリプトは競合を検出し、アップグレード後に登録解除および削除できる Satellite Server の重複エントリーがあるホストをリストできます。また、組織に割り当てられていないホストを検出します。**Hosts** > **All hosts** を選択して組織の関連付けがないホストがリストされ、同じ名前のコンテンツホストに組織がすでに関連付けられている場合、コンテンツホストは自動的に登録解除されます。これは、アップグレード前にこのようなホストを組織に関連付けることによって回避できます。

アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。

a. アップグレード前スクリプトを使用するには、**ruby193-rubygem-katello-2.2.0.90-1-sat** 以降が必要です。

```
# yum update ruby193-rubygem-katello
```

b. アップグレードの前に、アップグレード前チェックスクリプトを実行して、削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードの前に組織にそれらのホストを関連付けることが推奨されます。

```
# foreman-rake katello:upgrade_check
```

アップグレードチェックで、タスクが実行中であることが原因の障害が報告された場合は、タスクが完了するまで待機することが推奨されます。一部のタスクはキャンセルすることができますが、Red Hat ナレッジベースソリューション How to manage paused tasks on Red Hat Satellite 6 のアドバイスに従って、安全にキャンセルできるタスクと安全にキャンセルできなタスクについて理解する必要があります。

3. DNS と DHCP の設定ファイルである **/etc/zones.conf** と **/etc/dhcp/dhcpd.conf** をバックアップします。インストーラーでは 1 つのドメインまたはサブネットしかサポートされないため、これらのバックアップから変更を復元する必要がある場合があります。

4. DNS または DHCP の設定ファイルを手動で編集し、変更を上書きしたくない場合は、以下のコマンドを実行します。

```
# katello-installer --capsule-dns-managed=false --capsule-dhcp-
managed=false
```

5. 有効なリポジトリーをリストします。

```
# subscription-manager repos --list-enabled
```

6. 以下のリポジトリーのみが有効であることを確認します。

```
rhel-X-server-satellite-tools-6.1-rpms
rhel-server-rhscl-X-rpms
rhel-X-server-satellite-6.1-rpms
rhel-X-server-rpms
```

ここで、X はベースシステムのメジャーバージョンです。他のリポジトリーが検出された場合は、リポジトリーの設定の手順に従ってそれらのリポジトリーを削除します。

7. ベースシステムで Satellite の以前のバージョン向けリポジトリーを無効にします。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# subscription-manager repos --disable rhel-6-server-satellite-
6.1-rpms
# subscription-manager repos --disable rhel-6-server-satellite-
tools-6.1-rpms
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# subscription-manager repos --disable rhel-7-server-satellite-
6.1-rpms
# subscription-manager repos --disable rhel-7-server-satellite-
tools-6.1-rpms
```

8. 必要な場合は、Satellite 6.1 リポジトリーが無効であることを確認するために、以下のようなコマンドを入力します。

```
# subscription-manager repos --list-enabled
```

9. Satellite Web UI でリポジトリーを設定します。
   a. Satellite Web UI で、**Content** > **Red Hat Repositories** に移動し、**RPM** タブを選択します。
   b. Red Hat Satellite **製品** を見つけ、展開します。
   c. **Repository Set** Red Hat Satellite 6.1 (RHEL X Server 用) (RPM) を見つけ展開します。
   d. RHEL X Server RPM x86_64 用 Red Hat Satellite 6.1 を選択解除します。
   e. **Repository Set** Red Hat Satellite 6.2 (RHEL X Server 用) (RPM) を見つけ展開します。
   f. RHEL X Server RPM x86_64 用 Red Hat Satellite 6.2 を選択します。
   g. Red Hat Enterprise Linux Server **製品** を見つけ、展開します。
   h. Red Hat Satellite Tools 6.2 (RHEL X Server 用) (RPM) を見つけ展開します。
   i. RHEL X Server RPMs x86_64 用 Red Hat Satellite Tools 6.2 を選択します。
10. 新しく有効になったリポジトリーを同期します。
    a. Satellite Web UI で、**Content (コンテンツ)** > **Sync Status (同意ステータス)** に移動します。
    b. 製品の横にある矢印をクリックして利用可能なリポジトリーを表示します。
    c. 6.2 用リポジトリーを選択します。
    d. **Synchronize Now** をクリックします。
    Satellite Tools リポジトリーを更新しようとするときにエラーが発生した場合は、カスタマーポータルまたは Satellite Web UI でマニフェストを**削除しないでください**。削除すると、すべてのコンテンツホストが登録解除されます。詳細については、Red Hat ナレッジベースソリューション「Cannot enable Red Hat Satellite Tools Repo on Satellite 6.2」を参照してください。
11. 6.1 バージョンリポジトリーを使用する既存のコンテンツビューを 6.2 向けの新しいバージョンで更新します。新しい 6.2 バージョンリポジトリーがあるコンテンツビューの更新されたバージョンを公開し、プロモートします。
12. リポジトリーの同期が完了したら、ベースシステムで新しいリポジトリーを有効にします。
    - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# subscription-manager repos --enable rhel-6-server-satellite-
6.2-rpms
# subscription-manager repos --enable rhel-6-server-satellite-
tools-6.2-rpms
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# subscription-manager repos --enable rhel-7-server-satellite-
6.2-rpms
# subscription-manager repos --enable rhel-7-server-satellite-
tools-6.2-rpms
```

13. Satellite Web UI で、**Hosts (ホスト)** > **Discovered hosts (検出されたホスト)** に移動します。検出されたホストが利用可能な場合は、それらを無効にし、**Discovered hosts (検出されたホスト)** ページ下のすべてのエントリーを削除します。組織設定メニューを使用して他のすべての組織を順番に選択し、必要に応じてこのアクションを繰り返します。アップグレードが完了したら、これらのホストを再起動します。

14. すべての外部 Capsule Server が組織に割り当てられていることを確認します。割り当てられていない場合、これらのサーバーは、ホスト統合の変更により登録解除された可能性があります。

15. リポジトリーキャッシュを削除します。

```
# yum clean all
```

16. 以下のパッケージをダウンロードします。

```
# yum install --downloadonly rubygem-
smart_proxy_remote_execution_ssh rubygem-smart_proxy_openscap
rubygem-smart_proxy_dynflow tfm-rubygem-smart_proxy_dynflow_core
tfm-rubygem-foreman_remote_execution katello-client-bootstrap
```

17. 更新されたすべてのパッケージをダウンロードします。

```
# yum update --downloadonly
```

18. Katello サービスを停止します。

```
# katello-service stop
```

19. 以前にダウンロードされたパッケージをインストールします。

```
# yum install rubygem-smart_proxy_remote_execution_ssh rubygem-
smart_proxy_openscap rubygem-smart_proxy_dynflow tfm-rubygem-
smart_proxy_dynflow_core tfm-rubygem-foreman_remote_execution
katello-client-bootstrap
```

20. 更新されたすべてのパッケージをインストールします。

```
# yum update
```

21. カスタム Apache サーバー設定がある場合は、次の手順でインストールデフォルト値に戻りま

す。**アップグレードの実行時に**変更される内容を確認する場合は、**--noop** (no operation) オプションとともにアップグレードコマンドを入力し、次の手順でアップグレードコマンドを入力するときに適用される変更内容を確認できます。このテストを行わない場合は、次の手順に進みます。または、以下のように手順を続行します。

a. 次の行を **/etc/httpd/conf/httpd.conf** 設定ファイルに追加します。

```
Include /etc/httpd/conf.modules.d/*.conf
```

b. **httpd** サービスを再起動します。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service httpd restart
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl restart httpd
```

c. **postgresql** データベースサービスおよび **mongod** データベースサービスを起動します。

- Red Hat Enterprise Linux 6 の場合は、以下のコマンドを入力します。

```
# service postgresql start
# service mongod start
```

- Red Hat Enterprise Linux 7 の場合は、以下のコマンドを入力します。

```
# systemctl start postgresql
# systemctl start mongod
```

d. 以下のように **--noop** オプションとともにコマンドを入力します。

```
# satellite-installer --scenario satellite --upgrade --verbose --
noop
```

/var/log/foreman-installer/satellite.log を参照して、**--noop** オプションが省略された場合に**適用される変更**を確認します。設定ファイルの変更を示す **+++** と **---** の記号を探します。上記の "no operation" コマンドにより実際にはファイルは作成されず、モジュール内の一部の Puppet リソースではファイルがそこに存在することが期待されるため、いくつかのエラーメッセージが表示されるはずです。

e. Katello サービスを停止します。

```
# katello-service stop
```

22. **--upgrade** オプションを使用してインストーラースクリプトを実行することによりアップグレードを実行します。

```
# satellite-installer --scenario satellite --upgrade
```

警告

config サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/satellite.yaml) was not
found, can not continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再び実行します。

23. これまでに行ったバックアップを使用して DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。

24. 前の手順で変更を行った場合は、Katello サービスを再起動します。

```
# katello-service restart
```

25. Satellite Web UI で Discovery テンプレートを更新します。

    a. **Hosts (ホスト)** > **Provisioning templates (テンプレートのプロビジョニング)** に移動します。

    b. **PXELinux global default (PXELinux グローバルデフォルト)** を選択します。

    c. **Template editor (テンプレートエディター)** ダイアログボックスで、以下のテキストに一致するよう **LABEL discovery** で始まるスタンザを更新することにより **PXELinux global default (PXELinux グローバルデフォルト)** テンプレート検出メニューエントリーを編集します。

```
LABEL discovery
MENU LABEL Satellite 6 Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0
nomodeset proxy.url=https://SATELLITE_CAPSULE_URL:9090
proxy.type=proxy
IPAPPEND 2
```

    - **proxy.type** オプションは **proxy** または **foreman** のいずれかになります。**proxy** の場合は、すべての通信が Capsule 経由で行われます。**foreman** の場合は、通信が直接 Satellite Server に行われます。

    - **proxy.url** には、Satellite Capsule または Server の URL を指定します。HTTP と HTTPS の両方のプロトコルがサポートされます。

26. OpenSCAP プラグインがインストールされており、デフォルトの OpenSCAP コンテンツが利用可能でない場合は、以下のコマンドを実行します。

```
# foreman-rake foreman_openscap:bulk_upload:default
```

27. Satellite Web UI で **Configure (設定)** > **Discovery Rules (検出ルール)** に移動し、選択された組織および場所を検出ルールに関連付けます。

28. Red Hat カスタマーポータルで Satellite Server を同期します。

    a. **Content (コンテンツ) > Sync Status (同期ステータス)** に移動します。
    同期可能な製品リポジトリーのリストが表示されます。

    b. 製品コンテンツ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。

    c. 同期するコンテンツを選択します。

    d. **Synchronize Now** をクリックします。
    コンテンツの同期には時間がかかることがあります。同期に要する時間は、ディスクドライブの速度やネットワーク接続の速度、同期対象として選択されたコンテンツの量によって異なります。

29. (オプション) 必要なコンテンツビューを公開します。
コンテンツビューを公開して、ホストが参照および使用できるようにする必要があります。公開する前に、コンテンツビュー定義に必要な製品、リポジトリー、およびフィルターが含まれることを確認する必要があります。

    a. メインメニューから、**Content (コンテンツ) > Content Views (コンテンツビュー)** を選択します。

    b. Name (名前) 列から、Satellite Server コンテンツビューを選択します。

    c. **Publish New Version (新規バージョンの公開)** をクリックします。

    d. コメントを入力し、**Save (保存)** をクリックします。

30. (オプション) コンテンツビューをプロモートします。

    a. メインメニューから、**Content (コンテンツ) > Content Views (コンテンツビュー)** を選択します。

    b. Name (名前) 列で、Satellite Server コンテンツビューを選択します。

    c. Versions (バージョン) タブで、最新バージョンを選択し、**Promote (プロモート)** をクリックします。

    d. プロモーションパスを特定し、適切なライフサイクル環境を選択して、**Promote Version (バージョンのプロモート)** をクリックします。
    処理が完了したら、更新されたコンテンツビューステータスが **Versions (バージョン)** タブに表示されます。

6.8. アップグレード後のクリーンアップ

このセクションのすべての手順はオプションです。ご使用のインストールに関連する手順のみを実行できます。

6.8.1. 冗長ファイアウォールルールの削除

Red Hat Satellite 6.2 は Elasticsearch を使用しないため、Elasticsearch に関連するファイアウォールルールは削除できます。これらは、宛先ポートが 9200 の行です。

Red Hat Enterprise Linux 6 での冗長ファイアウォールルールの削除

1. ファイアウォールルールをリストします。

```
# iptables -nL --line-numbers
```

2. 以下の行を特定し、削除します。チェーン名は OUTPUT であり、行番号は異なることがあることに注意してください。

```
Chain OUTPUT (policy ACCEPT)
num  target   prot opt source         destination
1    ACCEPT   tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
owner UID match 496
2    ACCEPT   tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
owner UID match 0
3    DROP     tcp  --  0.0.0.0/0      0.0.0.0/0      tcp dpt:9200
```

3. iptables ルールを削除します。

```
iptables -D <chain-name> <line-number>
```

たとえば、上記の出力から行 1 を削除するには、以下のようなコマンドを入力します。

```
# iptables -D OUTPUT 1
```

4. 行の削除後に、変更を保存します。

```
# service iptables save
```

5. iptables サービスが起動され、有効であることを確認します。

```
# service iptables start
# chkconfig iptables on
```

Red Hat Enterprise Linux 7 での冗長ファイアウォールルールの削除

1. IPv4 直接ルールをリストします。

```
# firewall-cmd --direct --get-rules ipv4 filter OUTPUT
```

2. IPv6 直接ルールをリストします。

```
# firewall-cmd --direct --get-rules ipv6 filter OUTPUT
```

3. IPv4 と IPv6 両方に対して以下の行を特定し、削除します。チェーン名は OUTPUT であり、最初の番号は優先度であることに注意してください。

```
0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner foreman -j
ACCEPT
```

```
0 -o lo -p tcp -m tcp --dport 9200 -m owner --uid-owner root -j
ACCEPT
1 -o lo -p tcp -m tcp --dport 9200 -j DROP
```

4. firewalld 直接ルールを削除します。

```
firewall-cmd --direct --remove-rule <inet_family> filter
<chain_name> rule
```

ここで、<inet_family> は IPv4 または IPv6 です。

たとえば、上記の IPv4 の行を削除する場合は、以下のようになります。

```
# firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner foreman -j ACCEPT \
&& firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 0 -o lo -p
tcp -m tcp --dport 9200 -m owner --uid-owner root -j ACCEPT \
&& firewall-cmd --direct --remove-rule ipv4 filter OUTPUT 1 -o lo -p
tcp -m tcp --dport 9200 -j DROP
```

5. IPv6 に対してコマンドを繰り返します。

6. firewall サービスが有効であり、起動されていることを確認します。

```
# systemctl enable firewalld
# systemctl start firewalld
```

6.8.2. Elasticsearch の削除

Red Hat Satellite 6.2 は Elasticsearch を使用しないため、Elasticsearch で使用されるパッケージとディレクトリーは削除できます。

Elasticsearch と関連パッケージの削除

必要なくなった以下のパッケージを削除します。

```
# yum erase elasticsearch sigar-java sigar snappy-java lucene4-contrib
lucene4
```

Elasticsearch ユーザーの削除

Elasticsearch により作成されたユーザーを削除します。

```
# userdel -r elasticsearch
```

Elasticsearch ディレクトリーの削除

データベースディレクトリーとその内容を削除します。

```
# rm -rf /var/lib/elasticsearch
```

6.8.3. 以前のバージョンの Satellite Tools リポジトリーの削除

Satellite 6.2 へのアップグレードが完了したら、Red Hat Satellite Tools 6.1 リポジトリーをコンテンツビューから削除し、無効にできます。

1. バージョン 6.1 の Satellite Tools リポジトリーの無効化

    a. Satellite Web UI で、**Content** > **Red Hat Repositories** に移動し、**RPM** タブを選択します。

    b. Red Hat Enterprise Linux Server **製品** を見つけ、展開します。

    c. **Repository Set** Red Hat Satellite Tools 6.1 (RHEL X Server 用) (RPM) を見つけ展開します。

    d. RHEL X Server RPM x86_64 用 Red Hat Satellite Tools 6.1 を選択解除します。

チェックボックスが選択不可である場合、リポジトリーはまだコンテンツビューに含まれます。リポジトリーの孤立したパッケージは、スケジュールされたタスク (cron job) により自動的に削除されます。

第7章 SATELLITE SERVER、CAPSULE SERVER、およびコンテンツホストの更新

Satellite のマイナーバージョン間の更新

更新は、Satellite Server、Capsule Server、およびコンテンツホストを新しいマイナーバージョンに移行するプロセスです。通常、更新ではセキュリティーの脆弱性にパッチが適用され、コードのリリース後に検出されたマイナーな問題が修正されます。一般的に、更新にはほとんど時間がかからず、ご使用の運用環境は影響を受けません。更新前に、潜在的な競合について、『Red Hat Satellite Release Notes』を参照してください。

以下の手順に従って、たとえば、6.2.0 から 6.2.1 へのマイナーバージョン間で更新を行います。

7.1. SATELLITE SERVER の更新

前提条件

- Satellite、Capsule、および Satellite Tools 向けの Satellite Server リポジトリーが同期されていることを確認します。
- 関連するすべてのコンテンツビューに対して更新済みリポジトリーをプロモートすることにより、各 Capsule およびコンテンツホストを更新できることを確認します。

Satellite Server を次のマイナーバージョンに更新

Satellite Server の更新手順:

1. 適切なリポジトリーのみが有効であることを確認します。
   a. 有効なリポジトリーをリストします。

   ```
   # subscription-manager repos --list-enabled
   ```

   b. 以下のリポジトリーのみが有効であることを確認します。

   ```
   rhel-X-server-rpms
   rhel-X-server-satellite-6.2-rpms
   rhel-server-rhscl-X-rpms
   ```

   ここで、**X** は、使用している Red Hat Enterprise Linux のメジャーバージョンです。必要な場合は、リポジトリーの無効化および有効化の詳細について、リポジトリーの設定を参照してください。自己登録 Satellite を使用するときは、Katello エージェントを提供する **rhel-X-server-satellite-tools-6.2-rpms** リポジトリーも存在することがあります。必要な場合は、詳細について「katello エージェントのインストール」を参照してください。
2. 自己登録 Satellite を使用している場合は、Satellite Server を**停止する前に**すべてのパッケージをダウンロードします。

   ```
   # yum update --downloadonly
   ```

   この手順は、**自己登録されていない** Satellites の場合はオプションです。
3. Katello を停止します。

```
# katello-service stop
```

4. すべてのパッケージを更新します。

```
# yum update
```

5. カーネルの更新が行われたら、システムを再起動します。

```
# reboot
```

6. 更新を実行します。

```
# satellite-installer --scenario satellite --upgrade
```

7. 自己登録 Satellite を使用している場合は、**goferd** を再起動します。
   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# service goferd restart
```

   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# systemctl restart goferd
```

7.2. CAPSULE SERVER の更新

Capsule Server を次のマイナーバージョンに更新

Capsule Server の更新手順:

1. 適切なリポジトリーのみが有効であることを確認します。
   a. 有効なリポジトリーをリストします。

```
# subscription-manager repos --list-enabled
```

   b. 以下のリポジトリーのみが有効であることを確認します。

```
rhel-X-server-rpms
rhel-X-server-satellite-capsule-6.2-rpms
rhel-server-rhscl-X-rpms
rhel-X-server-satellite-tools-6.2-rpms
```

   ここで、**X** は、使用している Red Hat Enterprise Linux のメジャーバージョンです。必要な場合は、リポジトリーの無効化および有効化の詳細について、「リポジトリーの設定」を参照してください。**rhel-X-server-satellite-tools-6.2-rpms** リポジトリーは、Katello エージェントを提供します。必要な場合は、詳細について「katello エージェントのインストール」を参照してください。Red Hat Software Collections リポジトリーはオプションですが、リモート実行機能を使用するには必要です。

2. Katello を停止します。

```
# katello-service stop
```

3. すべてのパッケージを更新します。

```
# yum update
```

4. カーネルの更新が行われたら、システムを再起動します。

```
# reboot
```

5. 更新を実行します。

```
# satellite-installer --scenario capsule --upgrade
```

6. **goferd** を再起動します。
   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# service goferd restart
```

   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# systemctl restart goferd
```

7.3. コンテンツホストの更新

コンテンツホストを次のマイナーバージョンに更新

コンテンツホストを更新するには、以下のコマンドを入力します。

1. すべてのパッケージを更新します。

```
# yum update
```

2. カーネルの更新が行われたら、システムを再起動します。

```
# reboot
```

3. **goferd** を再起動します。
   - Red Hat Enterprise Linux 6 の場合は、以下のコマンドを実行します。

```
# service goferd restart
```

   - Red Hat Enterprise Linux 7 の場合は、以下のコマンドを実行します。

```
# systemctl restart goferd
```

第8章 SATELLITE SERVER および CAPSULE SERVER のアンインストール

Satellite Server または Capsule Server は、必要なくなったらアンインストールできます。

8.1. SATELLITE SERVER のアンインストール

Satellite Server と Capsule Server をアンインストールすると、ターゲットシステムで使用されたすべてのアプリケーションが削除されます。アプリケーションまたはアプリケーションデータを Satellite Server 以外の目的で使用する場合は、削除する前にそれらの情報をバックアップする必要があります。

作業を開始する前に

アンインストールスクリプトを実行すると、2 つの警告が 発生し、システムのすべてのパッケージと設定ファイルを削除する前に確認が求められます。

警告

このスクリプトを実行すると、多くのパッケージと設定ファイルが削除されます。以下のような重要なパッケージが削除されます。

- httpd (apache)
- mongodb
- tomcat
- puppet
- Ruby
- rubygems
- すべての Katello および Foreman パッケージ

Satellite Server のアンインストール

1. Satellite Server をアンインストールします。

```
# katello-remove
```

```
Once these packages and configuration files are removed there is no
going back.
If you use this system for anything other than Katello and Foreman
you probably
do not want to execute this script.
Read the source for a list of what is removed.  Are you sure(Y/N)? y
ARE YOU SURE?: This script permanently deletes data and
configuration.
Read the source for a list of what is removed.  Type [remove] to
```

```
continue? remove
Shutting down Katello services...
```

8.2. CAPSULE SERVER のアンインストール

Capsule Server をアンインストールすると、ターゲットシステムで使用されたすべてのアプリケーションが削除されます。アプリケーションまたはアプリケーションデータを Satellite Server 以外の目的で使用する場合は、削除する前にそれらの情報をバックアップする必要があります。

作業を開始する前に

アンインストールスクリプトを実行すると、2 つの警告が 発生し、システムのすべてのパッケージと設定ファイルを削除する前に確認が求められます。

警告

このスクリプトを実行すると、パッケージと設定ファイルが削除されます。以下のような重要なパッケージが削除されます。

- httpd (apache)
- mongodb
- tomcat
- puppet
- Ruby
- rubygems
- すべての Katello および Foreman パッケージ

Capsule Server のアンインストール

1. Capsule Server をアンインストールします。

```
$ capsule-remove
```

次のようなメッセージが表示されます。

```
Once these packages and configuration files are removed there is no
going back.
If you use this system for anything other than Katello and Foreman
you probably
do not want to execute this script.
Read the source for a list of what is removed.  Are you sure(Y/N)? y
ARE YOU SURE?: This script permanently deletes data and
configuration.
```

```
Read the source for a list of what is removed.  Type [remove] to
continue? remove
Shutting down Katello services...
```

第9章 詳細情報の提供元

最初のインストールおよびセットアップの最後に、追加設定を実行し、Satellite 環境をセットアップできます。詳細については、以下の Satellite ドキュメンテーションリソースを参照してください。

- **Hammer CLI Guide**
- **Server Administration Guide**
- **Host Configuration Guide**
- **Content Management Guide**
- **Puppet Guide**
- **Virtual Instances Guide**

付録A 大規模デプロイメントに関する考慮事項

Apache 向けファイル記述子の最大数の増加

800 を超えるコンテンツホストが登録されている場合、Apache では複数のシステムレベルの制限に到達し、新しいコンテンツホストの登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

1. Red Hat Enterprise Linux 7 の場合は、**/etc/systemd/system/httpd.service.d/limits.conf** ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
```

3. Katello サービスを再起動します。

```
# katello-service restart
```

qpid 向けファイル記述子の最大数の増加

1100 を超えるコンテンツホストでエラータ更新のために goferd が実行されている場合、qpid ではシステムレベルの制限に到達し、登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

Red Hat Enterprise Linux 7 を使用した qpid 向けファイル記述子の最大数の増加

1. **/etc/systemd/system/qpidd.service.d/limits.conf** ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
# systemctl restart qpidd.service
```

Red Hat Enterprise Linux 6 を使用した qpid 向けファイル記述子の最大数の増加

1. **/etc/security/limits.conf** ファイルを編集し、以下のテキストを挿入します。

```
qpidd - nofile 65536
```

2. qpidd サービスを再起動します。

```
# service qpidd restart
```

共有バッファーと作業メモリーの増加

shared_buffer と **work_mem** を 256M と 4M にそれぞれ増加できます。

1. Red Hat Enterprise Linux 7 の場合は、**/var/lib/pgsql/data/postgresql.conf** ファイルを作成し、以下のテキストを挿入します。

```
work_mem = 4MB
shared_buffers = 256MB
```

2. postgresql サービスを再起動します。

```
# service postgresql restart
```

同時コンテンツホスト登録の増加

システムレベルの制限への到達を回避するために、最大 250 の同時コンテンツホストを処理するようグローバルパッセンジャーキュー制限を増加できます。

1. 最大パッセンジャープールサイズを、Satellite Server で利用可能な物理 CPU コアの 1.5 倍に調整します。
たとえば、Satellite Server に 16 コアある場合、最大パッセンジャープールサイズは 24 です。
この数は例として示されており、ご使用の環境に応じた数を使用する必要があります。

2. **/etc/httpd/conf.d/passenger.conf** ファイルを編集して以下のテキストに一致するよう **IfModule** スタンザを更新します。

```
<IfModule mod_passenger.c>
  PassengerRoot /usr/share/gems/gems/passenger-
4.0.18/lib/phusion_passenger/locations.ini
  PassengerRuby /usr/bin/ruby
  PassengerMaxPoolSize 24
  PassengerMaxRequestQueueSize 200
  PassengerStatThrottleRate 120
</IfModule>
```

3. Foreman Passenger アプリケーション設定ファイル **/etc/httpd/conf.d/05-foreman-ssl.conf** を編集して、以下のテキストに一致するよう **PassengerAppRoot** で始まるスタンザを更新します。

```
 PassengerAppRoot /usr/share/foreman
 PassengerRuby /usr/bin/tfm-ruby
 PassengerMinInstances 6
 PassengerStartTimeout 90
 PassengerMaxPreloaderIdleTime 0
 PassengerMaxRequests 10000
 PassengerPreStart https://example.com
```

4. Puppet Passenger アプリケーション設定ファイル **/etc/httpd/conf.d/25-puppet.conf** を編集して以下のテキストを仮想ホスト定義の最後に追加します。

```
 PassengerMinInstances 6
 PassengerStartTimeout 90
 PassengerMaxPreloaderIdleTime 0
 PassengerMaxRequests 10000
 PassengerPreStart https://example.com:8140
```

5. **/var/lib/pgsql/data/postgresql.conf** ファイルで最大接続数を変更します。

```
max_connections = 500
```

6. postgresql サービスを再起動します。

```
# service postgresql restart
```

qdrouterd 向けオープンファイルの最大数の増加

1000 を超えるコンテンツホストが登録されている場合、**qdrouterd** はオープンファイルのデフォルトの最大数に到達することがあります。この問題を回避するには、Satellite サーバーとすべての外部 Capsule サーバーのオープンファイルの最大数を増加します。

1. 以下の式を使用して、オープンファイルの必要な最大数を計算します。

```
(3 x コンテンツホストの数) + 100
```

たとえば、1020 のコンテンツホストの場合、新しい最大数は 3160 ((3 x 1020) + 100) に設定します。

2. Red Hat Enterprise Linux 7 の場合は、ファイル **/etc/systemd/system/qdrouterd.service.d/limits.conf** を作成し、以下のテキストを追加します。

```
[Service]
LimitNOFILE=maximum_number_of_files
```

a. 変更をユニットに適用します。

```
# systemctl daemon-reload
```

b. Satellite のサービスを再起動します。

```
# katello-service restart
```

3. Red Hat Enterprise Linux 6 の場合は、ファイル **/etc/security/limits.conf** を編集し、以下の行を追加します。

```
qdrouterd   -   nofile  maximum_number_of_files
```

新しい行は **# End of file** 行 (これ以降の情報は無視されます) の**前に**追加します。

a. **qdrouterd** サービスを再起動します。

```
# service qdrouterd restart
```

付録B CAPSULE SERVER のスケーラビリティーに関する考慮事項

Satellite Server がサポートできる Capsule Server の最大数には固定された制限がありません。テスト済みの制限は、Red Hat Enterprise Linux 6.6 および 7 ホストの Satellite Server で 17 の Capsule Server と 2 の vCPU です。ただし、スケーラビリティーは非常に柔軟です (特に Puppet クライアントを管理する場合)。

Puppet クライアントを管理するときの Capsule Server のスケーラビリティーは、CPU の数、実行間隔の分散、および Puppet 管理リソースの数によって異なります。Capsule Server には、ある時点で実行されている同時 Puppet エージェントの数が 100 という制限があります。100 を超える同時 Puppet エージェントを実行すると、503 HTTP エラーが発生します。

たとえば、Puppet エージェントの実行が、1 つの実行間隔のある時点で実行されている 100 未満の同時 Puppet エージェントで均等に分散されると仮定した場合に、4 CPU で構成される Capsule Server での最大値は 1250～1600 Puppet クライアントになり、各 Puppet クライアントに中程度のワークロードである 10 Puppet クラスが割り当てられます。必要な Puppet クライアントの数に応じて、Satellite のインストールでは、Capsule Server の数をスケールアウトできます。

Puppet クライアントの管理時に Capsule Server をスケーリングする場合は、以下のことを前提とします。

- Satellite 6 統合 Capsule に直接報告する外部 Puppet クライアントが存在しません。
- 他のすべての Puppet クライアントは外部 Capsule に直接報告します。
- すべての Puppet エージェンの実行間隔が均等に分散されています。

注記

均等に分散されないと、パッセンジャー要求キューがいっぱいになる可能性が高くなります。100 の同時要求の制限が適用されます。

以下の表は、推奨される 4 CPU と Red Hat Enterprise Linux 7 を使用した場合のスケーラビリティーの制限を示しています。

表B.1 Red Hat Enterprise Linux 7 で 4 CPU を使用した場合の Puppet のスケーラビリティー (推奨)

1 つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	3000～2500
10	2400～2000
20	1700～1400

以下の表は、最小 2 CPU と Red Hat Enterprise Linux 7 を使用した場合のスケーラビリティーの制限を示しています。

表B.2 Red Hat Enterprise Linux 7 で 2 CPU を使用した場合の Puppet のスケーラビリティー

1 つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	1700～1450
10	1500～1250
20	850～700

以下の表は、推奨される 4 CPU と Red Hat Enterprise Linux 6 を使用した場合のスケーラビリティーの制限を示しています。

表B.3 Red Hat Enterprise Linux 6 で 4 CPU を使用した場合の Puppet のスケーラビリティー (推奨)

1 つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	2250～1875
10	1600～1250
20	700～560

以下の表は、最小 2 CPU を使用した場合のスケーラビリティーの制限を示しています。

表B.4 Red Hat Enterprise Linux 6 で 2 CPU を使用した場合の Puppet のスケーラビリティー

1 つのホストあたりの Puppet 管理リソース数	実行間隔の分散
1	未テスト
10	1020～860
20	375～330