Red Hat Satellite 6.5

Capsule Server のインストール

Red Hat Satellite Capsule Server のインストール

Last Updated: 2019-06-20

Red Hat Satellite 6.5 Capsule Server のインストール

Red Hat Satellite 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

本書では、Red Hat Satellite 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を説明します。

目次

第1章 インストールのための環境準備

1.1. システム要件

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。

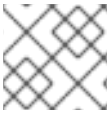
- 64 ビットアーキテクチャー
- Red Hat Enterprise Linux 7 Server の最新バージョン
- 最低 4 コア 2.0 GHz CPU
- Satellite Server が機能するには、最低 20 GB のメモリーが必要です。また、最低 4 GB のスワップ領域が推奨されます。最低値よりも少ないメモリーで実行している Satellite は正常に動作しないことがあります。
- 一意なホスト名 (小文字、数字、ドット (.)、ハイフン (-) を使用できます)
- 現在の Red Hat Satellite サブスクリプション
- 管理ユーザー (root) アクセス
- システム umask 0022
- 完全修飾ドメイン名を使用した完全な正引きおよび逆引きの DNS 解決

Satellite 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要件を満たしていることを確認する必要があります。

Satellite Server は、新規にプロビジョニングされたシステムにインストールする必要があり、このシステムで Satellite Server の実行以外の機能は提供されません。

注記

Red Hat Satellite Server と Capsule Server のバージョンは同じでなければなりません。たとえば、Satellite 6.2 Server は 6.5 Capsule Server を実行できず、Satellite 6.5 Server は 6.2 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、警告なしで Capsule Server が失敗します。



注記

自己登録の Satellites はサポートされません。

コンテンツホストが多数ある場合には、「大規模なデプロイメントに関する考慮事項」を参照して、お使いの環境が正しく設定されていることを確認してください。

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。

認定ハイパーバイザー

Red Hat Satellite は、Red Hat Enterprise Linux の実行をサポートするハイパーバイザーで稼働する物理システムおよび仮想マシン両方を完全にサポートしています。認定ハイパーバイザーに関する詳細は、「Red Hat Enterprise Linux の実行が認定されているハイパーバイザー 」 を参照してください。

FIPS モード

FIPS モードで稼働する Red Hat Enterprise Linux システムに、Satellite Server および Capsule Server をインストールできます。詳細は、『**Red Hat Enterprise Linux セキュリティーガイド**』の「FIPS モードの有効化」を参照してください。

1.2. ストレージの要件とガイドライン

このセクションでは、最小ストレージ要件を紹介し、Capsule Server のインストールのストレージに関するガイドラインについて説明します。

1.2.1. ストレージ要件

以下の表には、特定のディレクトリーに推奨されるストレージ要件が詳述されています。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。表を参照する場合には、ご自身にあったユースケースに注目してください。たとえば、Capsule Server で Pulp を有効化せずに使用できます。そのような場合には、**/var/lib/pulp/** など、Pulp 関連のディレクトリーと同レベルのストレージ要件は必要ありません。

以下の表では、ランタイムサイズは Red Hat Enterprise Linux 5、6、および 7 のリポジトリーと同期して測定されています。

表1.1 Capsule Server インストールのストレージ要件

ディレクトリー	インストールサイズ	ランタイムサイズ
/var/cache/pulp/	1M バイト	20 GB (最小)
/var/lib/pulp/	1 MB	500 GB
/var/lib/mongodb/	3.5 GB	50 GB
/opt	500 MB	適用外

1.2.2. ストレージのガイドライン

Capsule Server をインストールして効率性を向上する場合には、以下のガイドラインを考慮してください。

- Capsule Server データの多くは **/var** ディレクトリーに格納されるため、LVM ストレージに **/var** をマウントして、システムがスケーリングできるようにしてください。
- **/var/lib/pulp/** ディレクトリーと **/var/lib/mongodb/** ディレクトリーには、ハードディスクドライブ (HDD) ではなく、帯域幅が高く、レーテンシーの低いストレージおよび SSD (ソリッドステートドライブ) を使用するようにしてください。Red Hat Satellite には I/O を大量に使用する操作が多数あるため、高レイテンシーで低帯域幅のストレージを使用すると、パフォーマンス低下の問題が発生します。インストールに、毎秒 60 - 80 メガバイトのスピードがあることを確認してください。**fio** ツールを使用すると、このデータが取得できます。**fio** ツールの詳細な使用方法は、Red Hat ナレッジベースのソリューション「Impact of Disk Speed on Satellite 6 Operations」を参照してください。

- **/var/lib/qpidd/** ディレクトリーでは、**goferd** サービスが管理するコンテンツホスト 1 つに対して使用される容量は 2 MB を少し超えます。たとえば、コンテンツホストの数が 10,000 個の場合、**/var/lib/qpidd/** に 20 GB のディスク容量が必要になります。

- **/var/cache/pulp/** と **/var/lib/pulp/** ディレクトリーに同じボリュームを使用することで、同期後に **/var/cache/pulp/** から **/var/lib/pulp/** にコンテンツを移動する時間を短縮できます。

ファイルシステムのガイドライン

- 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 では XFS ファイルシステムを使用してください。Capsule Server は多く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。

- MongoDB は従来の I/O を使用してデータファイルにアクセスしないので、MogoDB では NFS を使用しないでください。また、NFS でデータファイルとジャーナルファイルの両方がホストされている場合にはパフォーマンスの問題が発生します。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。

NFS マウントを使用する場合の SELinux の考慮事項

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

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

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

```
# chcon -R system_u:object_r:httpd_sys_rw_content_t:s0 /var/lib/pulp
```

重複パッケージ

異なるリポジトリーで重複するパッケージは、ディスク上に一度しか格納されないため、重複するパッケージを含む追加リポジトリーで必要なストレージが少なくて済みます。ストレージの多くは、**/var/lib/mongodb/** ディレクトリーおよび **/var/lib/pulp/** ディレクトリーに使用されます。これらのエンドポイントは手動で設定できません。ストレージの問題を回避するために、ストレージが **/var** ファイルシステムで利用可能であることを確認してください。

一時的なストレージ

/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保管されるファイルは最大 5 RPM になります。各ファイルは、同期後に **/var/lib/pulp/** ディレクトリーに移動します。デフォルトでは、同時に最大 8 個の RPM コンテンツ同期タスクを実行でき、それぞれに対して最大 1 GB のメタデータが使用されます。

ISO イメージ

ISO 形式のコンテンツについては、同期タスクごとに ISO ファイルはすべて、タスクが完了するまで **/var/cache/pulp/** に保存されます。タスクが完了すると **/var/lib/pulp/** ディレクトリーに移動します。

インストールや更新に ISO イメージを使用する予定の場合には、外部ストレージを提供するか、ISO ファイルを一時的に保存するために **/var/tmp** に領域を空けるようにする必要があります。

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/** ディレクトリーに合計 16 GB 必要になります。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るので、同期する ISO ファイルの数を考慮してください。

ソフトウェアコレクション

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 ユーザーによる書き込みパーミッションおよび実行パーミッションが必要です。

シンボリックリンク

/var/lib/pulp/ および **/var/lib/mongodb/** にはシンボリックリンクは使用できません。

ログのストレージ

ログファイルは、**/var/log/messages/**、**/var/log/httpd/**、および **/var/lib/foreman-proxy/openscap/content/** の場所で確認できます。ログファイルのサイズを管理するには、**logrotate** 設定ファイルを使用します。詳細は、Red Hat Enterprise Linux 7 **『システム管理者のガイド』**の「ログローテーション」 を参照してください。

1.3. 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 6.5 のインストール時に入手可能な Red Hat Enterprise Linux 7 Server の最新バージョンでのみサポートされています。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。機能強化や Red Hat 以外のセキュリティーソフトウェアもこの制限に含まれます。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、Satellite Server が完全に機能することを最初に確認し、その後でシステムのバックアップを作成して、Red Hat 以外のソフトウェアを追加します。

新たにプロビジョニングされたシステムで、Satellite Server および Capsule Server をインストールし、Capsule Server は Red Hat コンテンツ配信ネットワーク (CDN) に登録しないでください。Red Hat は、Satellite 以外を実行するシステムの使用はサポートしません。

1.4. 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、Capsule をインストールするベースのオペレーティングシステムで、特定のネットワークポートを開放し、ネットワークベースのファイアウォールを無効にしておく必要があります。インストールの開始前に Satellite Server と Capsule Server の間のポートが開放されていない場合には、Capsule Server のインストールに失敗します。

以下の表では、宛先ポートおよびネットワークトラフィックの方向を紹介します。この情報を使用して、ネットワークベースのファイアウォールを設定してください。クラウドソリューションによっては、ネットワークベースのファイアウォールと同様にマシンが分離されるので、特にマシン間の通信ができるように設定する必要があります。アプリケーションベースのファイアウォールを使用する場合に

は、アプリケーションベースのファイアウォールで、テーブルに記載のアプリケーションすべてを許可して、ファイアウォールに既知の状態にするようにしてください。可能であれば、アプリケーションのチェックを無効にして、プロトコルをベースにポートの通信を開放できるようにしてください。

統合 Capsule

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下の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。

Capsule のクライアント

Satellite と統合された Capsule ではなく、Capsule のクライアントであるホストには、Satellite Server へのアクセスが必要ありません。Satellite トポロジーの詳細は**『Red Hat Satellite 6 のプランニング』**の「Capsule ネットワーク」を参照してください。

使用している設定に応じて、必要なポートは変わることがあります。

表1.2 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
53	TCP および UDP	DNS	Capsule の DNS サービスに問い合わせるクライアント DNS (オプション)

ポート	プロトコル	サービス	用途
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 (オプション)
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 (オプション)

表1.3 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
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	Katello、Foreman、Foreman API、および Pulp への接続
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

表1.4 クライアントに通信する Capsule 向けポート

ポート	プロトコル	サービス	用途
7	TCP および UDP	ICMP	DHCP Capsule からクライアントネットワークへ、IP アドレスが空きであることを確認するために ICMP ECHO を送信 (オプション)
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 (オプション)

ポート	プロトコル	サービス	用途
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 からクライアントへの "reboot" コマンド (オプション)

Satellite Server に直接接続された管理対象ホストは、統合された Capsule のクライアントとなるため、このコンテキストではクライアントになります。これには、Capsule Server が稼働しているベースシステムが含まれます。

表1.5 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
22	TCP	SSH	Remote Execution (Rex) および Ansible 向けの Satellite および Capsule からの通信
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。

注記

DHCP Capsule は ICMP ECHO を送信して、IP アドレスが空であることを確認します。**応答なし** などが返されるはずです。ICMP はネットワークベースのファイアウォールで切断される場合がありますが、**どのような** 応答でも IP アドレスの割り当てが妨げられます。

1.5. SATELLITE SERVER およびクライアントから CAPSULE SERVER への接続の有効化

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。

1. Capsule をインストールするベースシステムに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="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

2. 変更を永続化します。

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

1.6. ファイアウォール設定の確認

firewall-cmd コマンドを使用して、ファイアウォール設定の変更を確認できます。

ファイアウォール設定の確認

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

詳細情報は、『**Red Hat Enterprise Linux 7 セキュリティーガイド**』の「firewalld の概要」を参照してください。

第2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 と同じですが、Red Hat CDN への接続にはプロキシーを使用しない設定になっている必要があります。詳細は「システム要件」を参照してください。

2.1. SATELLITE SERVER への登録

この手順を使用して、Capsule をインストールするベースシステムを Satellite Server に登録します。

サブスクリプションのマニフェストに関する考慮事項

- Satellite Server には、今後 Capsule を所属させる組織の適切なリポジトリーで、マニフェストをインストールする必要があります。
- マニフェストには、Capsule をインストールするベースシステうのリポジトリーと、Capsule に接続するクライアントが含まれている必要があります。
- リポジトリーは、同期されている必要があります。

マニフェストとリポジトリーに関する詳しい情報は、**『Red Hat Satellite コンテンツ管理ガイド』**の「サブスクリプションの管理」を参照してください。

プロキシーおよびネットワークの考慮事項

- Satellite Server のベースシステムは、Capsule のインストール先のベースシステムのホスト名を、また Capsule も 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。
- Red Hat Satellite へのアクセスを妨げるプロキシーの使用に関連した変更を元に戻す必要があります。
- ホストとネットワークベースのファイアウォールが設定済みである必要があります。詳細は「ポートとファイアウォールの要件」を参照してください。
- Satellite Server のユーザー名とパスワードが必要です。詳細は**『Red Hat Satellite の管理』**の「外部認証の設定」 を参照してください。

Satellite Server への登録

1. Capsule をインストールするベースシステムに 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. 今後 Capsule を所属させる環境で、Capsule のインストール先のベースシステムを登録します。アクティベーションキーを使用して、環境の指定を簡素化します。

```
# subscription-manager register --org=organization_name --
activationkey=example_activation_key
```

2.2. 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。その他のサブスクリプションは必要ありません。

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。

```
Subscription Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Beta
                     Red Hat Satellite
                     Red Hat Satellite with Embedded Oracle
                     Red Hat Satellite 5 Managed DB
                     Red Hat Satellite Proxy
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat Satellite Capsule
                     Red Hat Software Collections Beta (for RHEL Server)
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Ansible Engine
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat Enterprise Linux Server
SKU:                 MCT3718
Contract:
Account:             540155
Serial:              6960416325892182336
Pool ID:             8a85f999655dc45a01658350d80d7164
Provides Management: Yes
Active:              True
Quantity Used:       1
Service Level:       Premium
Service Type:        L1-L3
Status Details:      Subscription is current
Subscription Type:   Standard
Starts:              08/28/2018
Ends:                09/19/2019
System Type:         Physical
```

2. 後で Satellite ホストに割り当てるために、プール ID をメモします。実際に使用するプール ID は、この例で使用されているものとは異なります。

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
```

2.3. リポジトリーの設定

1. 既存のリポジトリーをすべて無効にします。

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

2. Red Hat Satellite Capsule、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 Capsule 機能で必要な、新しいバージョンの Ruby を提供します。

```
# subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-7-server-satellite-capsule-6.5-rpms \
--enable rhel-server-rhscl-7-rpms \
--enable rhel-7-server-satellite-maintenance-6-rpms \
--enable rhel-7-server-ansible-2.6-rpms
```

3. Red Hat 以外のすべての **yum** リポジトリーのメタデータをすべて削除します。

```
# 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
```

2.4. 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と、証明書の検証に失敗することがあります。

NTP ベースの時刻シンクロナイザーは **chronyd** と **ntpd** の 2 種類利用できます。**chronyd** 実装は、特に、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るシステムに推奨されます。**ntpd** 実装は、**chronyd** でまだサポートされていないプロトコルまたはドライバーに対するサポートが必要な場合にのみ使用してください。

ntpd と **chronyd** の違いについては、**『システム管理者のガイド』**の「ntpd と chronyd の違い」を参照してください。

chronyd を使用した時間の同期

1. chronyd をインストールします。

```
# yum install chrony
```

2. chronyd サービスを起動して、有効にします。

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

2.5. CAPSULE SERVER のインストール

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-capsule
```

2.6. 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を紹介します。他の高度な設定オプションの詳細は「Capsule Server への追加の設定」を参照してください。

2.6.1. 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)。

Satellite Server がカスタムの SSL 証明書を使用するように設定した場合には、「カスタムサーバー証明書を使用した Capsule Server の設定」に移動します。

作業開始前の準備

- Capsule がインストールされており、**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。
- ホストとネットワークベースのファイアウォールが設定済みである必要があります。詳細は「ポートとファイアウォールの要件」を参照してください。
- **katello-ca-consumer-latest** パッケージがインストール済みである必要があります。詳細は、「Satellite Server への登録」 を参照してください。
- Capsule Server が Satellite Server に登録されている必要があります。
- 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。

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 \
--foreman-proxy-fqdn mycapsule.example.com \
--certs-tar mycapsule.example.com-certs.tar
```

capsule-certs-generate コマンドの出力である **satellite-installer** コマンドをメモし、Capsule Server 証明書をインストールします。

2. 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。
3. Capsule Server で **capsule-certs-generate** コマンドが出力する **satellite-installer** コマンドを実行して、Capsule Server 証明書をインストールします。

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-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 \
--foreman-proxy-content-certs-tar mycapsule.example.com-certs.tar \
--puppet-server-foreman-url "https://satellite.example.com"
```

注記

Satellite へのネットワーク接続やポートをまだ開いていない場合は、**--foreman-proxy-register-in-foreman** オプションを **false** に設定すると、Capsule が Satellite へ接続を試行しなくなり、エラー報告がなくなります。ネットワークとファイアウォールを適切に設定したら、このオプションを **true** にして再度インストーラーを実行します。

第3章 CAPSULE SERVER での追加設定の実行

3.1. 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。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。

作業開始前の準備

- 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。

```
# subscription-manager repos \
--enable=rhel-7-server-satellite-tools-6.5-rpms
```

3. パッケージをインストールします。

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

3.2. CAPSULE SERVER でリモート実行の有効化

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。

注記

デフォルトでは、外部の Capsule はリモート実行機能が無効になっています。Capsule Server でリモート実行を使用するには、以下のコマンドを実行して、これを有効にする必要があります。

```
# satellite-installer --scenario capsule \
--enable-foreman-proxy-plugin-remote-execution-ssh
```

3.3. 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、環境を追加して、Capsule が Satellite Server 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るようにする必要があります。

リポジトリーが CDN から更新されるたびに自動で Capsule が同期されるようになるので、**ライブラリー** ライフサイクル環境を Capsule Server に割り当てないでください。自動的に同期されると、Capsule 上の複数のシステムリソースや Satellite と Capsule 間のネットワーク帯域幅、および Capsule 上の利用可能なディスク領域が消費される可能性があります。

Satellite Server の Hammer CLI または Satellite Web UI を使用できます。

手順

ライフサイクル環境を Capsule Server に追加するには、以下の手順を実行します。

1. Satellite Web UI で、**インフラストラクチャー** > **Capsule** に移動し、ライフサイクルを追加する Capsule を選択します。
2. **編集** をクリックしてから、**ライフサイクル環境** タブをクリックします。
3. 左のメニューから、Capsule に追加するライフサイクル環境を選択し、 **送信** をクリックします。
4. Capsule のコンテンツを同期するには、**概要** タブをクリックしてから **同期** ボタンをクリックします。
5. **最適化された同期** または **完全な同期** を選択します。

CLI をご利用の場合

1. Capsule Serverの全一覧を表示するには、以下のコマンドを入力します。

```
# hammer capsule list
```

返された ID をメモします。

2. その ID を使用して、Capsule Server の詳細を確認します。

```
# hammer capsule info --id capsule_id
```

3. 利用可能なライフサイクル環境を確認し、環境 ID を書き留めます。

```
# hammer capsule content available-lifecycle-environments \
--id capsule_id
```

4. Capsule Server で利用可能なライフサイクル環境を表示するには、以下のコマンドを入力して、組織名と ID をメモします。

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

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 --organization "My_Organization" \
--environment-id environment_id
```

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。

Satellite Server 環境のすべてのコンテンツを Capsule Server に同期するには、以下のコマンドを実行します。

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

Satellite Server 環境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 \
--environment-id environment_id
```

3.4. 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 です (BMC 機能とも呼ばれます)。IPMI は、ホストの CPU から独立して実行す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で、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 サービスの詳細は**『ホストの管理』**の「追加のネットワークインターフェースの設定」を参照してください。

作業開始前の準備

- すべての管理対象ホストに **BMC** タイプ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. オプションを使用してインストーラーを実行し、BMC を有効にします。

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

3.5. CAPSULE SERVER で DNS、DHCP、および TFTP 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、「外部サービスの設定」を参照してください。

設定可能な全オプションを表示するには、**satellite-installer --scenario capsule --help** コマンドを実行します。

作業開始前の準備

- 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-dns true \
--foreman-proxy-dns-managed 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-managed 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-managed true \
--foreman-proxy-tftp-servername $(hostname)
```

DHCP、DNS および TFTP サービスの情報は、**『プロビジョニングガイド』**の「ネットワークサービスの設定」セクションを参照してください。

3.6. 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、デフォルトの SSL 証明書が含まれており、Satellite Server、Capsule Server、およびすべてのホスト間で暗号化通信を可能にします。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、特定の認証局から SSL 証明書を取得する必要があると規定されている場合などです。

前提条件

- カスタムの証明書で設定した Satellite Server。詳しい情報は、**『オンラインネットワークからの Satellite Server のインストール』**の「カスタムサーバー証明書での 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。詳細は「2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 証明書の取得」
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. 「すべてのホストへの Capsule Server の新しい証明書のインストール」

3.6.1. Capsule Server 向けの SSL 証明書の取得

重要

SSL 証明書には、PEM エンコードを使用してください。

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**。

手順

Satellite Serve で、Capsule Server 向けのカスタム SSL 証明書を取得します。

1. **root** ユーザーのみがアクセスできる、全ソース証明書ファイルの保存用のディレクトリーを作成します (例: **/root/capsule_cert**)。

```
# mkdir /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、ディレクトリー名をサーバー名と同じに指定します。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すると良いでしょう。これは**必須**ではありませんが、特定の Capsule Server のファイルを別の Capsule Server で使用するリスクが軽減されます。

2. 証明書署名要求 (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/capsule_cert_key.pem 4096
```

3. 証明書署名要求 (CSR) 用に **/root/sat_cert/openssl.cnf** 設定ファイルを作成し、以下の内容を追加します。**[req_distinguished_name]** セクションで、組織の情報を入力します。

注記

証明書の Common Name (CN) およびは Subject Alternative Name (SAN) の DNS.1 は、証明書を使用するサーバーの完全修飾ドメイン名 (FQDN) と同じでなければなりません。要求しているのが Satellite Server の証明書の場合には、これは Satellite Server の FQDN で、Capsule サーバーの証明書の場合には、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。

```
[ req ]
req_extensions = v3_req
distinguished_name = req_distinguished_name
x509_extensions = usr_cert
prompt = no

[ req_distinguished_name ]
C  = Country Name (2 letter code)
ST = State or Province Name (full name)
L  = Locality Name (eg, city)
O  = Organization Name (eg, company)
```

```
OU = The division of your organization handling the certificate
CN = capsule.example.com

[ v3_req ]
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
subjectAltName = @alt_names

[ usr_cert ]
basicConstraints=CA:FALSE
nsCertType = client, server, email
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
nsComment = "OpenSSL Generated Certificate"
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer

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

4. 証明書署名要求 (CSR) を作成します。

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

5. 証明局に証明書署名要求を送信します。同じ証明局が Satellite Server と Capsule Server の証明書に署名する必要があります。
要求を送信する場合は、証明書の有効期限を指定してください。証明書署名要求を送信する方法は異なるため、推奨の方法について認証局にお問い合わせください。要求への応答で、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。

3.6.2. 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。このプロセスでは、Capsule Server の鍵、CSR、および SSL 証明書が Capsule Server から Satellite Server にコピーされている必要があります。

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

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. 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、出力には以下の情報が含まれます。

-

```
Validation succeeded
```

katello-certs-check コマンドの出力である **capsule-certs-generate** コマンドをメモして、以下の手順で使用します。

「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。

3.6.3. 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、サーバーの証明書がアーカイブファイルで必要になります。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。

Satellite Server で実行すると、パラメーターなど **capsule-certs-generate** コマンドが **katello-certs-check** で出力されます。詳しい情報は、**『オンラインネットワークからの Satellite Server のインストール』**の「カスタムサーバー証明書での Satellite Server の設定」 を参照してください。

1. エディターで **capsule-certs-generate** コマンドのコピーを準備します。
2. Capsule Server の FQDN に一致するよう **--foreman-proxy-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 をインストールしていない場合は、**--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. 編集した **capsule-certs-generate** コマンドをテキストエディターから端末にコピーします。
5. 編集した **capsule-certs-generate** コマンドを実行します。

   capsule-certs-generate コマンドの例

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

6. 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を **root** ユーザーのホームディレクトリーにコピーしていますが、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-certs-generate コマンドの出力である **satellite-installer** コマンドをメモして、以下の手順で使用します。

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

3.6.4. Capsule Server のカスタム証明書のインストール

警告

この手順は、Capsule サーバーで完了してください。

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-installer** スクリプトをカスタムパラメーターで実行します。コマンドとパラメーターは、「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** コマンドで得た出力を使用します。

1. エディターで **satellite-installer** コマンドのコピーを準備をします。
2. **--foreman-proxy-content-certs-tar** の値を、証明書アーカイブファイルの場所に変更します。
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。インストーラーの全パラメーターを確認するには、**satellite-installer --scenario capsule --help** コマンドを実行してください。
4. 編集した **satellite-installer** コマンドをテキストエディターから端末にコピーします。
5. 編集した **satellite-installer** コマンドを実行します。

カスタム satellite-installer コマンドの例

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-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" \
--foreman-proxy-content-certs-tar "/root/capsule_certs.tar" \
--puppet-server-foreman-url "https://satellite.example.com"
```

注記

satellite-installer コマンドの値は、**capsule-certs-generate** コマンドの出力からも分かるように、各 Capsule Server に対して一意です。したがって、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後であっても、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、Capsule サーバーをアップグレードする場合などに必要になります。インストーラーで証明書アーカイブファイルが検出されない場合は、以下のようなメッセージが表示されて失敗します。

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

「すべてのホストへの Capsule Server の新しい証明書のインストール」 に進みます。

3.6.5. すべてのホストへの Capsule Server の新しい証明書のインストール

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
```

3.7. 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。

1. ファイヤー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. 変更を永続化します。

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

第4章 外部サービスの設定

本セクションでは、 Red Hat Satellite Capsule Server が外部 の DNS、DHCP、TFTP サーバーを使用する設定について説明します。

4.1. 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. 所有者、パーミッション、SELinux コンテキストが正しいことを確認します。

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

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

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

8. **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、 **アクション** ドロップダウンリストから **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。

4.2. CAPSULE SERVER での外部 DHCP の設定

外部 DHCP で Capsule Server を設定するには、先に DHCP サーバーを設定して、NFS 経由でその DHCP 設定とリースファイルを共有する必要があります。

DHCP サーバーの設定、DHCP 設定およびリースファイルの共有

1. Red Hat Enterprise Linux Server をデプロイし、ISC DHCP サービスおよび BIND (Berkeley Internet Name Domain) をインストールします。

```
# yum install dhcp bind
```

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
```

上記のコマンドが完了するまで、しばらく時間がかかることがあります。安全性が高くない概念実証のデプロイメントでは、非ブロック型乱数ジェネレーターを使用できます。

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

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

3. キーからシークレットハッシュをコピーします。

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

4. 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。例を示します。

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

subnet 192.168.38.0 netmask 255.255.255.0 {
 range 192.168.38.10 192.168.38.100;
 option routers 192.168.38.1;
 option subnet-mask 255.255.255.0;
 option domain-search "virtual.lan";
 option domain-name "virtual.lan";
 option domain-name-servers 8.8.8.8;
}

omapi-port 7911;
key omapi_key {
 algorithm HMAC-MD5;
 secret "jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

5. 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。

6. 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、リース範囲を 192.168.38.10 から 192.168.38.100 とし、予約範囲 (Satellite Web UI で定義済み) を 192.168.38.101 から 192.168.38.250 とします。ここでは、定義されたサブネットに DHCP Capsule を設定しないでください。

7. 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。

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

8. Capsule Server の foreman ユーザーの UID 番号と GID 番号を確認します。

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

9. ユーザー ID とグループ ID が同じユーザーとグループを、DHCP サーバーに作成します。

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

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。

```
# systemctl start dhcpd
```

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. **/etc/fstab** ファイルに以下の行を追加して、新規作成ディレクトリー用にマウントポイントを作成します。

```
/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 Satellite Server 向けの DHCP omapi ポート 7911 を設定します。

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

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。

- **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 --runtime-to-permanent
```

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 your_DHCP_server_FQDN
# rpcinfo -p your_DHCP_server_FQDN
```

5. **/etc/fstab** ファイルに以下の行を追加します。

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

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

6. **/etc/fstab** 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=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=your_DHCP_server_FQDN
```

9. foreman-proxy サービスを再起動します。

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

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

11. **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、 **アクション** ドロップダウンリストから **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。

4.3. 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. DHCP サービスとは異なるサーバーで TFTP サービスを実行している場合は、**tftp_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。

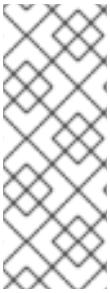
6. Satellite Server Web インターフェースにログインします。
7. **インフラストラクチャー** > **Capsules** に移動します。適切な Capsule Server を見つけ、**アクション** ドロップダウンリストから **更新** を選択します。この結果、TFTP 機能が現れます。
8. TFTP サービスに適切なサブネットとドメインを関連付けます。

4.4. SATELLITE または CAPSULE での外部 IDM DNS の設定

Red Hat Satellite は、 Red Hat Identity Management (IdM) サーバーを使って DNS サービスを提供するように設定できます。これには 2 つ方法があり、その両方でトランザクションキーを使用します。Red Hat Identity Management の詳細は『Linux ドメイン ID、認証、およびポリシーガイド』を参照してください。

1 つ目の方法では、RFC3645 で定義された **generic security service algorithm for secret key transaction** (GSS-TSIG) 技術を使用してプロセスを自動化する IdM クライアントをインストールします。この方法では、Satellite Server か Capsule のベースシステムに IdM クライアントをインストールし、Satellite 管理者が使用するアカウントを IdM サーバーの管理者が作成する必要があります。詳細は「GSS-TSIG 認証を使用した動的 DNS 更新の設定」を参照してください。

2 つ目の方法である **secret key transaction authentication for DNS**(TSIG) では、認証に **rndc.key** を使用します。root 権限で IdM サーバーにアクセスして BIND 設定ファイルを編集する必要があります。Satellite Server に **BIND** ユーティリティーをインストールし、システム間で **rndc.key** をコピーします。この技術は、RFC2845 で定義されています。詳細は「TSIG 認証を使用した動的 DNS 更新の設定」を参照してください。



注記

DNS の管理には、Satellite を使用する必要はありません。Satellite のレルム登録機能を使用していて、プロビジョニングされたホストが自動的に IdM に登録されている場合は、**ipa-client-install** スクリプトでクライアント用に DNS レコードが作成されます。このため、以下の手順とレルム登録は、同時に使用することはできません。レルム登録の詳細は**『Red Hat Satellite の管理』**の 「プロビジョンされたホストの外部認証」 を参照してください。

IdM クライアントのインストール先

Satellite Server がホスト用に DNS レコードを追加する際には、まずどの Capsule がそのドメインの DNS を提供しているかを判断します。その後に Capsule と通信し、レコードを追加します。ホスト自

体はこのプロセスに関与していません。つまり、IdM クライアントをインストールして設定する Satellite または Capsule は、IdM サーバーを使って管理するドメインに DNS サービスを提供するように現在設定されているものにすべきということになります。

4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

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

ホスト名	**satellite.example.com**
ネットワーク	**192.168.55.0/24**

IdM 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**
ドメイン名	**example.com**

作業開始前の準備

1. IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されていることを確認します。詳細は『**Linux ドメイン ID、認証、およびポリシーガイド**』の「ポート要件」を参照してください。
2. IdM サーバーに、IdM サーバーにゾーンを作成するパーミッションのあるアカウントを作成します。
3. Satellite または外部 Capsule がドメインの DNS を管理していることを確認します。
4. Satellite または外部 Capsule が正常に機能していることを確認します。
5. 新たにインストールしたシステムの場合は、まず本ガイドにあるインストール手順を完了させます。特に、DNS と DHCP の設定は完了させてください。
6. 変更を元に戻す場合に備えて、応答ファイルのバックアップを作成します。詳細は「インストールオプションの指定」を参照してください。

IdM サーバー上で Kerberos プリンシパルの作成

1. Kerberos チケットがあることを確認します。

```
# kinit idm_user
```

ここでの **idm_user** は、IdM 管理者が作成したアカウントになります。

2. IdM サーバーに認証する際に使用する Satellite または Capsule 用の新規 Kerberos プリンシパルを作成します。

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

IdM クライアントのインストールと設定

以下の手順は、ドメインの DNS サービスを管理している Satellite または Capsule サーバーで行います。

1. IdM クライアントパッケージをインストールします。

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

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

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

3. Kerberos チケットがあることを確認します。

```
# kinit admin
```

4. 既存の keytab を削除します。

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

5. このシステム用に作成された keytab を取得します。

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

注記

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

6. 以下のように、**foreman-proxy** への keytab ファイルのグループと所有者を設定します。

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

7. 必要に応じて、keytab が有効か確認します。

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

IdM web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。

   a. **Network Services (ネットワークサービス)** > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。

   b. **Add** を選択し、ゾーン名を入力します。この例では、**example.com** になります。

   c. **Add and Edit** をクリックします。

   d. 設定タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

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

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

f. **Allow PTR sync** を有効にします。

g. **Save** を選択して、変更を保存します。

2. 逆引きゾーンを作成、設定します。

a. **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。

b. **Add** を選択します。

c. **Reverse zone IP network** を選択して、CIDR 形式でネットワークアドレスを追加し、逆引き参照を有効にします。

d. **Add and Edit** をクリックします。

e. **Settings** タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

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

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

g. **Save** を選択して、変更を保存します。

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

- Satellite Server のベースシステムでは、以下を実行します。

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate_gss \
--foreman-proxy-dns-server="idm1.example.com" \
--foreman-proxy-dns-tsig-principal="capsule/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab \
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

- Capsule Server のベースシステムでは、以下を実行します。

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

```
--foreman-proxy-dns-reverse="55.168.192.in-addr.arpa" \
--foreman-proxy-dns-zone=example.com \
--foreman-proxy-dns-ttl=86400
```

Satellite または Capsule のプロキシーサービスを再起動します。

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

Satellite web UI での設定更新

インストールスクリプトを実行して Capsule に変更を加えた後に、Satellite が該当する各 Capsule の設定をスキャンするようにします。

1. **インフラストラクチャー** > **Capsules (スマートプロキシー)** に移動します。
2. 更新する Capsule で、**アクション** ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。
   a. **インフラストラクチャー** > **ドメイン** に移動し、ドメイン名を選択します。
   b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている Capsule に選択されていることを確認します。
4. サブネットを設定します。
   a. **インフラストラクチャー** > **サブネット** に移動し、サブネット名を選択します。
   b. **サブネット** タブで、**IPAM** を **None** に設定します。
   c. **ドメイン** タブで、IdM サーバーが管理するドメインが選択されていることを確認します。
   d. **Capsules** タブで、**Reverse DNS Capsule**、サブネットが接続されている Capsule に選択されていることを確認します。
   e. **送信** をクリックして変更を保存します。

4.4.2. TSIG 認証を使用した動的 DNS 更新の設定

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**

IdM 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**

作業開始前の準備

1. IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されていることを確認します。詳細は『**Linux ドメイン ID、認証、およびポリシーガイド**』の「ポート要件」を参照してください。
2. IdM サーバーで **root** 権限を取得します。
3. Satellite または外部 Capsule がドメインの DNS を管理していることを確認します。
4. Satellite または外部 Capsule が正常に機能していることを確認します。
5. 新たにインストールしたシステムの場合は、まず本ガイドにあるインストール手順を完了させます。特に、DNS と DHCP の設定は完了させてください。
6. 変更を元に戻す場合に備えて、応答ファイルのバックアップを作成します。詳細は「インストールオプションの指定」を参照してください。

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

```
// 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. **named** をリロードして、変更を有効にします。

```
# systemctl reload named
```

3. IdM Web UI で、**Network Services (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。
4. 以下のように、IdM サーバーから Satellite のベースシステムへ **/etc/rndc.key** ファイルをコピーします。

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

5. 所有者、パーミッション、SELinux コンテキストが正しいことを確認します。

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

6. Satellite Server で以下のようにインストールスクリプトを実行し、外部 DNS サーバーを使用します。

```
# 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
```

IdM サーバーの DNS ゾーンに対する外部アップデートのテスト

1. 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
```

2. Satellite Server 上の **/etc/rndc.key** ファイルのキーが IdM サーバーで使用されている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==";
};
```

3. Satellite Server で、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 IdM 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
```

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

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

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

5. IdM 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。

6. 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
```

7. DNS エントリーが削除されたことを確認します。

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

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

4.4.3. 内部 DNS サービス使用への復元

Satellite Server と Capsule Server を DNS プロバイダーとして使用するように戻すには、以下の手順に従います。

ドメインの DNS を管理する Satellite または Capsule Server

- 外部 DNS への変更前に応答ファイルをバックアップした場合は、応答ファイルを復元して、インストールスクリプトを実行します。

```
# satellite-installer
```

- 応答ファイルのバックアップがない場合は、現行の応答ファイルでバックアップを作成し、以下にあるように Satellite および Capsules でインストールスクリプトを実行します。
応答ファイルに関する情報は、「インストールオプションの指定」を参照してください。

応答ファイルを使用せずに Satellite または Capsule を DNS サーバーとして設定

```
# satellite-installer \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=true \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="127.0.0.1"  \
--foreman-proxy-dns-tsig-principal="foremanproxy/satellite.example.com@EXAMPLE.COM" \
--foreman-proxy-dns-tsig-keytab=/etc/foreman-proxy/dns.keytab
```

詳しい情報は、「Capsule Server での DNS、DHCP および TFTP の設定」を参照してください。

Satellite web UI での設定更新

インストールスクリプトを実行して Capsule に変更を加えた後に、Satellite が該当する各 Capsule の設定をスキャンするようにします。

1. **インフラストラクチャー** > **Capsules (スマートプロキシー)** に移動します。
2. 更新する Capsule で、**アクション** ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。
   a. **インフラストラクチャー** > **ドメイン** に移動し、ドメイン名を選択します。
   b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている Capsule に選択されていることを確認します。

4. サブネットを設定します。

    a. **インフラストラクチャー** > **サブネット** に移動し、サブネット名を選択します。

    b. **サブネット** タブで、**IPAM** を **DHCP** または **Internal DB** に設定します。

    c. **ドメイン** タブで、Satellite または Capsule が管理するドメインが選択されていることを確認します。

    d. **Capsules** タブで、**Reverse DNS Capsule**、サブネットが接続されている Capsule に選択されていることを確認します。

    e. **送信** をクリックして変更を保存します。

第5章 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。

作業開始前の準備

katello-remove スクリプトを実行すると、2 つの警告が出され、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

このスクリプトは、以下のものを含む、パッケージおよび設定ファイルを削除します。

- httpd (apache)
- mongodb
- tomcat
- puppet
- Ruby
- rubygems
- すべての Katello および Foreman パッケージ

手順

1. Satellite Web UI で、**ホスト** > **すべてのホスト** に移動し、ターゲットホストを選択します。Capsule Server インスタンスの右側の **編集** リストから **削除** を選択します。
2. **インフラストラクチャー** > **Capsule** に移動し、Capsule Server インスタンスの右側の **編集** リストから **削除** を選択します。
3. Capsule Server で **katello-remove** コマンドを入力して Capsule Server をアンインストールします。

```
# katello-remove
```

CLI をご利用の場合

1. Satellite Server で、全 Capsule Server をリストして、削除する Capsule Server インスタンスの FQDN および ID を検索します。

```
# hammer capsule list
```

2. Satellite Server で、**hammer host delete** コマンドを入力して、**--name** オプションを使用して Capsule Server の FQDN を指定し、Satellite ホストから Capsule Server を削除します。

```
# hammer host delete --name Capsule_Server_FQDN
```

3. Satellite Server で、**hammer capsule delete** コマンドを入力して、**--id** オプションを使用して Capsule Server の ID を指定し、Satellite ホストから Capsule Server を削除します。

```
# hammer capsule delete --id Capsule_Server_ID
```

4. Capsule Server で **katello-remove** コマンドを入力して Capsule Server をアンインストールします。

```
# katello-remove
```

付録A CAPSULE SERVER のスケーラビリティーに関する考慮事項

Satellite Server がサポート可能な Capsule Server の最大数には上限がありません。テスト済みの上限は、Red Hat Enterprise Linux 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満たされるリスクが高くなります。100 の同時要求の制限が適用されます。

以下の表は、推奨の 4 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A.1 4 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	3000 ~ 2500
10	2400 ~ 2000
20	1700 ~ 1400

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A.2 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

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
10	1500 ~ 1250
20	850 ~ 700