Red Hat Satellite 6.6

Capsule Server のインストール

Red Hat Satellite Capsule Server のインストール

Last Updated: 2020-04-29

Red Hat Satellite 6.6 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。

- x86_64 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
- Capsule Server が機能するには、最低 12 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 Capsule 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

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。

Capsule Server は、新たにプロビジョニングしたシステムにインストールしておく。このシステムは、Capsule Server を実行するためだけに使用するようにします。Capsule Server が作成するローカルのユーザーとの競合を回避するため、新たにプロビジョニングしたシステムには、外部アイデンティティープロバイダーで設定した、以下のユーザーを使用しないようにしてください。

- postgres
- mongodb
- apache
- tomcat
- foreman
- foreman-proxy
- qpidd
- qdrouterd
- squid
- puppet

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 6.6 Capsule Server を実行できず、Satellite 6.6 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。

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

Capsule server は、Red Hat Enterprise Linux の実行をサポートするハイパーバイザーで稼働する物理システムおよび仮想マシン両方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、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. ストレージ要件

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

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

表1.1 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	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/opt	500 MB	適用外

1.3. 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。

- Pulp データストレージに NFS を使用しないでください。Pulp に NFS を使用すると、コンテンツの同期のパフォーマンスが低下します。

- 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。

1.4. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする方法であれば他の方法でもインストールできます。Red Hat Capsule Server は、Capsule Server 6.6 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 など、以前の Red Hat Enterprise Linux バージョンはサポートされません。

Red Hat 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、Capsule 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、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、Capsule Server 以外を実行するシステムの使用はサポートしません。

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

Satellite Server と Capsule Server の間のポートがインストール開始前に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。

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

統合 Capsule

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下のセクションのコンテキストでは Satellite のクライアントになります。これには、Capsule Server が実行されているベース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。

ポートのマトリックス表は、Red Hat ナレッジベースソリューションの「Red Hat Satellite 6.6 List of Network Ports」を参照してください。

以下の表は、宛先ポートとネットワークトラフィックの方向を示しています。

表1.2 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
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

表1.3 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

ポート	プロトコル	サービス	用途
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
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)
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)
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続 (オプション)

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)
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。

1.6. CAPSULE SERVER から SATELLITE SERVER への接続の有効化

Satellite Server で、Capsule Server から Satellite Server に対する受信接続を有効にして、再起動後にもルールが保持されるようにする必要があります。

前提条件

- Capsule Server は Satellite Server のクライアントであるため、クライアント が Satellite と通信できるように、Satellite Server でファイアウォールルールが有効にされていること。詳細は、『**オンラインネットワークからの Satellite Server のインストール**』の「クライアントから Satellite Server への接続の有効化」を参照してください。

手順

1. Satellite Server で、次のコマンドを入力して Capsule から Satellite への通信に使用するポートを開放します。

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

2. 変更を永続化します。

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

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

Capsule のインストール先のベースオペレーティングシステムで、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て、再起動後にもこれらのルールが維持されるようにします。

手順

1. Capsule のインストール先のベースオペレーティングシステムで、次のコマンドを入力して、Satellite Server およびクライアントから Capsule Server への通信に使用するポートを開放します。

```
# 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
```

1.8. ファイアウォール設定の確認

この手順を使用して、ファイアウォール設定への変更を検証します。

手順

ファイアウォールの設定を検証するには、以下の手順を実行します。

1. 以下のコマンドを実行します。

```
# 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、お使いの環境がインストール要件を満たしていることを確認してください。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 Server 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。

前提条件

Satellite Server に登録する前に、Capsule のインストール先のベースシステムが次の条件を満たしていることを確認してください。

サブスクリプションのマニフェストの前提条件

- Satellite Server にマニフェストをインストールし、Capsule が所属する組織に適したリポジトリーが含まれている必要がある。
- マニフェストには、Capsule をインストールするベースシステムのリポジトリーと、Capsule に接続するクライアントが含まれている必要がある。
- リポジトリーは、同期されている必要がある。

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。

プロキシーとネットワークの前提条件

- Satellite Server のベースシステムは、Capsule 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 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る。
- Capsule Server のインストール先のベースシステムには、Red Hat コンテンツ配信ネットワーク (CDN) への接続にプロキシーを使用しないように設定しておく。
- 要件に合わせてホストとネットワークベースのファイアウォールを設定する必要がある。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。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。

手順

Satellite Server にシステムを登録するには、以下の手順を実行します。

1. Capsule をインストールするベースシステムに、**katello-ca-consumer-latest.noarch.rpm** パッケージをダウンロードします。コンシューマー RPM で、ホストが Red Hat Satellite で指定したコンテンツソースからコンテンツをダウンロードするように設定します。

```
# curl --insecure --output katello-ca-consumer-latest.noarch.rpm
https://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

2. **katello-ca-consumer-latest.noarch.rpm** パッケージをインストールします。

```
# yum localinstall katello-ca-consumer-latest.noarch.rpm
```

3. Capsule の所属先の環境で、Capsule 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. SATELLITE INFRASTRUCTURE サブスクリプションのアタッチ

Capsule Server の登録後に、サブスクリプションプール ID を特定して、利用可能なサブスクリプションをアタッチする必要があります。Red Hat Satellite Infrastructure のサブスクリプションを使用すると、Red Hat Satellite、Red Hat Enterprise Linux および Red Hat Software Collections (RHSCL) コンテンツにアクセスできるようになります。必要なサブスクリプションはこれだけです。

Red Hat Satellite Infrastructure は、Smart Management を提供するサブスクリプションすべてに含まれます。詳細は、Red Hat ナレッジベースソリューション「Satellite Infrastructure Subscriptions MCT3718 MCT3719」を参照してください。

サブスクリプションがシステムにアタッチされていない場合には、利用可能として分類されます。利用可能な Satellite サブスクリプション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で使用されているかどうかを確認できます。

手順

Satellite Infrastructure サブスクリプションをアタッチするには、以下の手順を実行します。

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

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

このコマンドを実行すると、以下のような出力が表示されます。

```
Subscription Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Satellite
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Ansible Engine
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Satellite Capsule
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Satellite
                     Red Hat Satellite 5 Managed DB
                     Red Hat Satellite 6
                     Red Hat Discovery
SKU:                 MCT3719
Contract:            11878983
Pool ID:             8a85f99968b92c3701694ee998cf03b8
```

```
Provides Management: No
Available:           1
Suggested:           1
Service Level:       Premium
Service Type:        L1-L3
Subscription Type:   Standard
Ends:                03/04/2020
System Type:         Physical
```

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

3. Capsule Server の実行先のベースシステムに、Satellite Infrastructure サブスクリプションをアタッチします。

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

このコマンドを実行すると、以下のような出力が表示されます。

```
サブスクリプションを正常に割り当てます: Red Hat Satellite Infrastructure サブスクリプション
```

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

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

2.3. リポジトリーの設定

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

手順

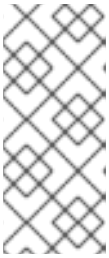
必要なリポジトリーを設定するには、以下の手順を実行します。

1. すべてのリポジトリーを無効にします。

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

2. 次のリポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-satellite-capsule-6.6-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-tools-6.6-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-ansible-2.8-rpms
```



注記

Red Hat Virtualization (RHV) がホストする仮想マシンとして、Capsule Server をインストールする場合は、**Red Hat Common** リポジトリーを有効にして、RHV ゲストエージェントとドライバーもインストールする必要があります。詳細は『**仮想マシン管理ガイド**』の「ゲストエージェントおよびドライバーのインストール」を参照してください。

3. **yum** メタデータを消去します。

```
# yum clean all
```

4. オプション: 必要な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. CHRONYD とシステムクロックの同期

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

chrony スイートに関する詳細は、『**Red Hat Enterprise Linux 7 システム管理者ガイド**』の「chrony スイートを使用した NTP の設定」を参照してください。

手順

システムクロックを同期するには、以下の手順を実行します。

1. **chrony** パッケージ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 パッケージのインストール

Capsule Server パッケージをインストールする前に、ベースシステムにインストールした全パッケージを更新する必要があります。

手順

Capsule Server をインストールするには、以下の手順を実行します。

1. すべてのパッケージを更新します。

```
# yum update
```

2. **satellite-capsule** パッケージをインストールします。

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

2.6. SSL 証明書を使用した CAPSULE SERVER の設定

Red Hat Satellite は SSL 証明書を使用して、Satellite Server、外部 Capsule Server、全ホストの間の暗号化通信を有効にします。組織の要件によっては、デフォルトの証明書またはカスタムの証明書で Capsule Server を設定する必要があります。

- また、デフォルトの SSL 証明書を使用する場合には、外部 Capsule Server ごとに異なるデフォルトの SSL 証明書を設定する必要があります。詳細は、「デフォルトの SSL 証明書を使用した Capsule Server の設定」 を参照してください。
- また、カスタムの SSL 証明書を使用する場合には、外部 Capsule Server ごとに異なるカスタムの SSL 証明書を使用して設定する必要があります。詳細は、「カスタム SSL 証明書を使用した Capsule Server の設定」 を参照してください。

2.6.1. デフォルトの SSL 証明書を使用した Capsule Server の設定

本セクションを使用して、Satellite Server のデフォルトの証明局 (CA) が署名した SSL 証明書を使用して Capsule Server を設定します。

前提条件

デフォルトのサーバー証明書で Capsule Server を設定する前に、Capsule Server が以下の条件を満たすように確認してください。

- Capsule Server が Satellite Server に登録されている。詳細は、「Satellite Server への登録」 を参照してください。
- Capsule Server パッケージがインストールされている。詳細は、「Capsule Server パッケージのインストール」を参照してください。

手順

デフォルトのサーバー証明書で Capsule Server を設定するには、以下の手順を実行します。

1. Satellite Server で Capsule Server の全ソース証明書ファイルを保存するには、**root** ユーザーのみがアクセスできるディレクトリーを作成します (例: **/root/capsule_cert**)。

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

2. Satellite Server で、Capsule Server の **/root/capsule_cert/capsule_certs.tar** 証明書アーカイブを生成します。

```
# capsule-certs-generate \
--foreman-proxy-fqdn capsule.example.com \
--certs-tar /root/capsule_cert/capsule_certs.tar
```

capsule-certs-generate コマンドが返す **satellite-installer** コマンドのコピーをメモし、Capsule Server に証明書をデプロイします。

capsule-certs-generate の出力例

```
Installing              Done                                [100%]
  Success!
```

```
To finish the installation, follow these steps:

If you do not have the Capsule registered to the Satellite instance, then please do the
following:

1. yum -y localinstall http://satellite.example.com.com/pub/katello-ca-consumer-
latest.noarch.rpm
2. subscription-manager register --org "Default_Organization"

Once this is completed run the steps below to start the Capsule installation:

1. Ensure that the satellite-capsule package is installed on the system.
2. Copy the following file /root/capsule_cert/capsule_certs.tar to the system
capsule.example.com at the following location /root/capsule_certs.tar
scp /root/capsule_cert/capsule_certs.tar
root@capsule.example.com:/root/capsule_certs.tar
3. Run the following commands on the Capsule (possibly with the customized
   parameters, see satellite-installer --scenario capsule --help and
   documentation for more info on setting up additional services):

satellite-installer \
--scenario capsule \
--certs-tar-file                    "/root/capsule_certs.tar"\
--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       "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f"\
--foreman-proxy-oauth-consumer-secret    "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"\
--puppet-server-foreman-url              "https://satellite.example.com"
```

3. Satellite Server から、証明書アーカイブファイルを Capsule Server にコピーします。

```
# scp /root/capsule.example.com-certs.tar
root@capsule.example.com:/root/capsule.example.com-certs.tar
```

4. Capsule Server で、証明書をデプロイするには、**capsule-certs-generate** コマンドにより返された **satellite-installer** コマンドを入力します。
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。

重要

証明書をデプロイした後に、証明書アーカイブを削除しないでください。このアーカイブは、Capsule Server のアップグレード時などに必要になります。

2.6.2. カスタム SSL 証明書を使用した Capsule Server の設定

Satellite Server がカスタムの SSL 証明書を使用するように設定する場合は、この設定時に、外部の各 Capsule Server も、異なるカスタム SSL 証明書で設定する必要があります。

カスタム証明書を使用して Capsule Server を設定するには、Capsule Server ごとに以下の手順を実行します。

1. 「Capsule Server のカスタム SSL 証明書の作成」
2. 「カスタムの SSL 証明書の Capsule Server へのデプロイ」
3. 「ホストへの カスタム SSL 証明書のデプロイ」

2.6.2.1. Capsule Server のカスタム SSL 証明書の作成

Satellite Server で、Capsule Server 用にカスタムの証明書を作成します。Capsule Server 用のカスタムの SSL 証明書がすでにある場合には、以下の手順は省略してください。

カスタム証明書を使用して Satellite を設定する場合には、次の点を考慮してください。

- SSL 証明書には、Privacy-Enhanced Mail (PEM) エンコードを使用する必要がある。
- Satellite Server と Capsule Server の両方に、同じ証明書を使用できない。
- 同じ証明局を使用して Satellite と Capsule の証明書を署名する必要がある。

手順

カスタムの SSL 証明書を作成するには、以下の手順を実行します。

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

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

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

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

```
# openssl genrsa -out /root/capsule_cert/capsule_cert_key.pem 4096
```

3. 証明書署名要求 (CSR) 用の **/root/capsule_cert/openssl.cnf** 設定ファイルを作成して、以下のコンテンツを追加します。

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

[ req_distinguished_name ] (1)
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 (2)
```

```
[ v3_req ]
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 (3)
```

1. **[req_distinguished_name]** セクションに、貴社の組織の情報を入力します。
2. 証明書のコモンネーム **CN** を、Capsule Server の完全修飾ドメイン名 (FQDN) と一致するように設定します。FQDN を確認するには、対象の Capsule Server で **hostname -f** コマンドを入力します。これは、**katello-certs-check** コマンドが証明書を正しく検証することを確認するために必要です。
3. サブジェクトの別名 (SAN: Subject Alternative Name) **DNS.1** を、お使いのサーバーの完全修飾ドメイン名 (FQDN) に一致する用に設定します。

4. 証明書署名要求 (CSR) を作成します。

```
# openssl req -new \
-key /root/capsule_cert/capsule_cert_key.pem \ (1)
-config /root/capsule_cert/openssl.cnf \ (2)
-out /root/capsule_cert/capsule_cert_csr.pem (3)
```

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

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。

2.6.2.2. カスタムの SSL 証明書の Capsule Server へのデプロイ

この手順を使用して、証明局が署名したカスタムの SSL 証明書で、Capsule Server を設定します。**capsule-certs-generate** コマンドにより返される、**satellite-installer** コマンドは、Capsule Server ごとに一意となっています。複数の Capsule Server に同じコマンドを使用しないでください。

前提条件

カスタムのサーバー証明書で Capsule Server を設定する前に、Satellite Server と Capsule Server が以下の条件を満たすように確認してください。

- Satellite Server は、カスタムの証明書で設定されている。詳細は、『**オンラインネットワークからの Satellite Server のインストール**』の「カスタムの SSL 証明書を使用した Satellite Server の設定」を参照してください。
- Capsule Server が Satellite Server に登録されている。詳細は、「Satellite Server への登録」を参照してください。
- Capsule Server パッケージがインストールされている。詳細は、「Capsule Server パッケージのインストール」を参照してください。

手順

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

1. Satellite Server で、カスタムの SSL 証明書の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
-b /root/capsule_cert/ca_cert_bundle.pem 3
```

1. 認証局が署名した Capsule Server の証明書ファイルへのパス
2. Capsule Server 証明書の署名に使用した秘密鍵へのパス
3. 認証局バンドルへのパス

このコマンドに成功すると、**capsule-certs-generate** コマンド 2 つが返されます。このうちのいずれか 1 つを、Capsule Server の証明書アーカイブの生成に使用する必要があります。

katello-certs-check の出力例

```
Validation succeeded.

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

capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
    --certs-tar  "~/$CAPSULE-certs.tar" \
    --server-cert "/root/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \

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

  capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" \
    --certs-tar  "~/$CAPSULE-certs.tar" \
    --server-cert "/root/capsule_cert/capsule_cert.pem" \
    --server-key "/root/capsule_cert/capsule_cert_key.pem" \
    --server-ca-cert "/root/capsule_cert/ca_cert_bundle.pem" \
    --certs-update-server
```

2. Satellite Server で、**katello-certs-check** コマンドの出力をもとに、要件に合わせて、**capsule-certs-generate** コマンドを入力し、新規または既存の Capsule の証明書を生成します。このコマンドでは、**$CAPSULE** は、お使いの Capsule Server の FQDN に置き換えます。

3. **capsule-certs-generate** コマンドが返す **satellite-installer** コマンドのコピーをメモし、Capsule Server に証明書をデプロイします。

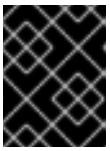
capsule-certs-generate の出力例

```
Installing     Done                                       [100%]
  Success!

  To finish the installation, follow these steps:

  If you do not have the Capsule registered to the Satellite instance, then please do the
following:

  1. yum -y localinstall http://satellite.example.com.com/pub/katello-ca-consumer-
latest.noarch.rpm
  2. subscription-manager register --org "Default_Organization"

  Once this is completed run the steps below to start the Capsule installation:

  1. Ensure that the satellite-capsule package is installed on the system.
  2. Copy the following file /root/capsule_cert/capsule_certs.tar to the system
capsule.example.com at the following location /root/capsule_certs.tar
  scp /root/capsule_cert/capsule_certs.tar
root@capsule.example.com:/root/capsule_certs.tar
  3. Run the following commands on the Capsule (possibly with the customized
     parameters, see satellite-installer --scenario capsule --help and
     documentation for more info on setting up additional services):

satellite-installer \
--scenario capsule \
--certs-tar-file                    "/root/capsule_certs.tar"\
--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      "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f"\
--foreman-proxy-oauth-consumer-secret   "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"\
--puppet-server-foreman-url             "https://satellite.example.com"
```

4. Satellite Server から、証明書アーカイブファイルを Capsule Server にコピーします。

```
# scp /root/capsule.example.com-certs.tar \
root@capsule.example.com:/root/capsule.example.com-certs.tar
```

5. Capsule Server で、証明書をデプロイするには、**capsule-certs-generate** コマンドにより返された **satellite-installer** コマンドを入力します。
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。



重要

証明書をデプロイした後に、証明書アーカイブを削除しないでください。このアーカイブは、Capsule Server のアップグレード時などに必要になります。

2.6.2.3. ホストへの カスタム SSL 証明書のデプロイ

Capsule Server がカスタムの SSL 証明書を使用するよう設定した後に、Capsule Server に登録されている全ホストに **katello-ca-consumer** パッケージもインストールする必要があります。

BZ#1683835 が解決されるまで、**katello-ca-consumer** パッケージはアップグレードできません。以前のパッケージを削除して、新しいパッケージをインストールする必要があります。**katello-ca-consumer** パッケージをアップグレードすると、**subscription.rhsm.redhat.com** の **baseurl** 設定が元に戻るため、アップグレードに失敗します。

手順

各ホストで、以下の手順を実行し、**katello-ca-consumer** パッケージをインストールします。

1. ホストで現行の **katello-ca-consumer** パッケージを削除します。

```
# yum remove 'katello-ca-consumer*'
```

2. ホストに **katello-ca-consumer** パッケージをインストールします。

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

第3章 CAPSULE SERVER での追加設定の実行

以下の章を使用して、Capsule Server の追加設定を行います。

3.1. KATELLO エージェントのインストール

Satellite クライアントをリモートで更新するには、Katello エージェントをインストールする必要があります。

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。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の情報を提供できるようにするには、このサービスを有効化する必要があります。

前提条件

Katello エージェントのインストール前に、以下の条件が満たされていることを確認してください。

- Satellite Server で、Satellite Tools リポジトリーを有効化しておく。詳細は、『**オンラインネットワークからの Satellite Server のインストール**』の「Satellite Tools リポジトリーのインストール」を参照してください。
- Satellite Server で Satellite Tools リポジトリーを同期しておく。詳細は、『**オンラインネットワークからの Satellite Server のインストール**』の「Satellite Tools リポジトリーの同期」を参照してください。
- クライアントで Satellite Tools リポジトリーを有効化しておく。たとえば、Red Hat Enterprise Linux 7 クライアントでリポジトリーが有効化されているかを確認するには、以下のコマンドをクライアントで実行してください:

```
# subscription-manager repos --enable rhel-7-server-satellite-tools-6.6-rpms
```

手順

Katello エージェントをインストールするには、以下の手順を実行します。

1. **libvirt-client** パッケージをインストールします。

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

2. **goferd** サービスを開始します。

```
# systemctl start goferd
```

3.2. CAPSULE SERVER でリモート実行の有効化

Capsule Server に登録されているホストでコマンドを実行するには、Capsule でリモート実行機能を有効にする必要があります。

外部 Capsule でのリモート実行は、デフォルトで無効になっています。

手順

- Capsule Server でリモート実行を有効化するには、以下のコマンドを入力します。

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

3.3. 外部 CAPSULE での OPENSCAP の有効化

Satellite Server および Satellite Server に統合された Capsule では、デフォルトで OpenSCAP は有効になっています。

外部 Capsule で OpenSCAP プラグインとコンテンツを使用する場合には、各 Capsule で OpenSCAP を有効にする必要があります。

手順

- OpenSCAP を有効にするには、次のコマンドを入力します。

```
# satellite-installer --scenario capsule \
--enable-foreman-proxy-plugin-openscap
```

3.4. 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** 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。
   各同期の定義については、『**コンテンツ管理ガイド**』の「リポジトリーの復元」を参照してください。

CLI をご利用の場合

1. Satellite Server で、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
```

ライフサイクルを追加する Capsule の Capsule ID を書き留めます。

2. その ID を使用して、Capsule 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。

6. Satellite から Capsule にコンテンツを同期します。

- 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
```

3.5. 管理対象ホスト上での電源管理の有効化

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

前提条件

- すべての管理対象ホストには、BMC タイプのネットワークインターフェースが必要。Capsule Server はこの NIC を使用して、適切な認証情報をホストに渡します。詳細は、『**ホストの管理**』ガイドの「ベースボード管理コントローラー (BMC) インターフェースの追加」を参照してください。

手順

- BMC を有効にするには、以下のコマンドを入力します。

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

3.6. CAPSULE SERVER で DNS、DHCP、および TFTP の設定

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

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

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

前提条件

DNS、DHCP および TFTP サービスを設定する前に、以下の条件を満たしていることを確認してください。

- 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る。
- 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る。
- ネットワーク管理者に連絡して正しい設定が行われていることを確認する。

手順

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

```
# satellite-installer --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-reverse 2.0.192.in-addr.arpa \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-interface eth0 \
--foreman-proxy-dhcp-range "192.0.2.100 192.0.2.150" \
--foreman-proxy-dhcp-gateway 192.0.2.1 \
--foreman-proxy-dhcp-nameservers 192.0.2.2 \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true \
--foreman-proxy-tftp-servername 192.0.2.3
```

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。

3.7. 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** ユーザーにだけ設定する必要があります。

ご使用の Capsule Server の **mongod** へのアクセスを制限するには、ファイアウォール構成を更新する必要があります。

手順

1. 以下のコマンドを入力して、ファイアウォール構成を更新します。

```
# 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 サービス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

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

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

前提条件

外部 DNS で Capsule Server を設定する前に、以下の条件を満たしていることを確認してください。

- 外部 DNS サーバーが構成されている必要がある。

手順

外部の DNS を使用して Capsule Server のロギングを設定するには、次の手順を実行します。

1. **bind-utils** パッケージをインストールしておく。

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

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

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

3. 所有者、パーミッション、SELinux コンテキストを設定します。

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

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

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

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

```
# satellite-installer --foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
```

```
--foreman-proxy-dns-server="_DNS_IP_Address_" \
--foreman-proxy-keyfile=/etc/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 UI にログインし、**インフラストラクチャー** > **Capsule** に移動します。
8. 外部 DNS で設定する Capsule Server の場所を特定して、**アクション** コラムのリストから **リフレッシュ** を選択します。
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

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

1. 「Capsule Server を使用するための外部 DHCP サーバーの設定」
2. 「外部 DHCP サーバーを使用した Capsule Server の設定」

4.2.1. Capsule Server を使用するための外部 DHCP サーバーの設定

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

手順

外部の DHCP サーバーを Capsule Server で使用できるように設定するには、以下の手順を実行します。

1. Red Hat Enterprise Linux Server で、ISC DHCP サービスおよび BIND (Berkeley Internet Name Domain) パッケージをインストールします。

```
# yum install dhcp bind
```

2. セキュリティートークンを生成します。

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

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

3. キーからシークレットハッシュをコピーします。

```
# 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。以下に例を示します。

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

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

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

option routers の値は、外部の DHCP サービスと使用する Satellite または Capsule IP アドレスに置き換える点に注意してください。

5. キーファイルが作成されたディレクトリーから、2 つのキーファイルを削除します。

6. Satellite Server で各サブネットを定義します。定義済みのサブネットに DHCP Capsule は設定しないでください。
競合を回避するには、リースと予約範囲を別に設定します。たとえば、リース範囲を 192.168.38.10 から 192.168.38.100 に設定した場合には、Satellite Web UI で予約範囲を 192.168.38.101 から 192.168.38.250 に設定します。

7. DHCP サーバーに外部アクセスできるように、ファイアウォールを設定します。

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

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

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

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

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

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

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

11. DHCP サービスを起動します。

```
# systemctl start dhcpd
```

12. NFS を使用して DHCP 設定ファイル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 設定ファイルとリースファイルのディレクトリーを作成します。

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

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

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

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** に以下の行があることを確認します。

```
/exports 192.168.38.1(rw,async,no_root_squash,fsid=0,no_subtree_check)

/exports/etc/dhcp 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)

/exports/var/lib/dhcpd 192.168.38.1(ro,async,no_root_squash,no_subtree_check,nohide)
```

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

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 DHCP omapi ポート 7911 を設定します。

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

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

- **firewalld** 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
```

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

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

前提条件

- 外部の DHCP サーバーを設定し、Capsule Server と DHCP 設定ファイルとリースファイルを共有していることを確認する。詳細は、「Capsule Server を使用するための外部 DHCP サーバーの設定」を参照してください。

手順

外部の DHCP を使用して Capsule Server のロギングを設定するには、次の手順を実行します。

1. **nfs-utils** ユーティリティー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リモートプロシージャコール (RPC: Remote Procedure Call) 通信パスを検証します。

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

5. **/etc/fstab** ファイルに以下の行を追加します。

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

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

6. **/etc/fstab** でファイルシステムをマウントします。

```
# mount -a
```

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

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

```
# satellite-installer --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=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 UI にログインします。
11. **Infrastructure** > **Capsules** に移動して、DHCP サーバーで設定する Capsule Server を特定し、**Actions** コラムの一覧から **Refresh** を選択します。
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

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

手順

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

1. NFS 用に TFTP ディレクトリーを作成します。

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

2. **/etc/fstab** ファイルで以下の行を追加します。

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

3. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

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

```
# satellite-installer --foreman-proxy-tftp=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 サービスを実行するサーバーの FQDN または IP アドレスに、**tftp_servername** 設定を更新します。

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

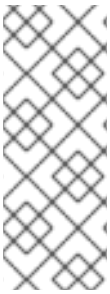
6. Satellite Server Web UI にログインします。
7. **インフラストラクチャー** > **Capsules** に移動し、適切な Capsule Server の場所を特定して、**Actions** コラムの一覧から、**Refresh** を選択します。
8. TFTP サービスに適切なサブネットとドメインを関連付けます。

4.4. 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 クライアントは、IdM サーバーで管理するドメイン向けに DNS サービスを提供するように現在設定されている Satellite または Capsule に、インストールして設定する必要があります。

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

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

1. Satellite Server または Capsule Server に **ipa-client** パッケージをインストールします。
   - Satellite Server で以下のコマンドを入力します。

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

- Capsule Server で以下のコマンドを入力します。

```
# 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. **追加** 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 アップデートポリシー** 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. **設定** タブの **BIND アップデートポリシー** 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 **リフレッシュ** 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 に設定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** が、サブネットが接続されている Capsule に設定されていることを確認します。
   e. **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

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、**ネットワークサービス** > **DNS** > **DNS ゾーン** に移動します。ゾーンの名前を選択します。**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 エラーメッセージを返します。

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
```

詳細は、「Capsule Server での DNS、DHCP および TFTP の設定」を参照してください。

Satellite 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 **リフレッシュ** 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 に設定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** が、サブネットが接続されている Capsule に設定されていることを確認します。

e. **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るすべてのアプリケーションが削除されます。ターゲットシステムで、アプリケーションまたはアプリケーションデータを Capsule Server の実行以外の目的で使用する場合は、Capsule Server をアンインストールする前に、情報をバックアップする必要があります。

Capsule Server をアンインストールするには、**katello-remove** コマンドを使用します。このコマンドは、システム内のすべてのパッケージと設定ファイルを削除する前に、警告を 2 つ表示して確認を求めます。

katello-remove コマンドは、以下のパッケージおよび設定ファイルを削除します。

- httpd (apache)
- mongodb
- tomcat6
- 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** に移動します。
2. アンインストールする Capsule Server の右側にある **編集** リストから、**削除** を選択します。
3. **インフラストラクチャー** > **Capsules** に移動します。
4. アンインストールする Capsule Server の右側にある **編集** リストから、**削除** を選択します。
5. Capsule Server で **katello-remove** コマンドを入力します。

```
# katello-remove
```

CLI をご利用の場合

1. Satellite Server で、Capsule Server の全一覧を表示し、アンインストールする Capsule Server の FQDN および ID を書き留めます。

```
# hammer capsule list
```

2. Satellite Server で、Satellite ホストの一覧から Capsule Server を削除するには、**hammer host delete** コマンドを入力し、**--name** オプションを使用して Capsule Server の FQDN を指定します。

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

3. Satellite Server で、Satellite Capsule の一覧から Capsule Server を削除するには、**hammer capsule delete** コマンドを入力し、**--id** オプションを使用して Capsule Server の ID を指定します。

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

4. Capsule Server で **katello-remove** コマンドを入力します。

```
# 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