



# Red Hat Satellite 6.9

## Capsule Server のインストール

Red Hat Satellite Capsule Server のインストール



# Red Hat Satellite 6.9 Capsule Server のインストール

---

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

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

## 法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installing\_Capsule\_Server.ent file |.

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

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

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

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

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

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

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

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

All other trademarks are the property of their respective owners.

## 概要

本書では、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

<b>第1章 インストールのための環境準備</b>	<b>3</b>
1.1. システム要件	3
1.2. ストレージ要件	4
1.3. ストレージのガイドライン	4
1.4. サポート対象オペレーティングシステム	6
1.5. ポートとファイアウォールの要件	6
1.6. CAPSULE SERVER から SATELLITE SERVER への接続の有効化	9
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	10
1.8. ファイアウォール設定の確認	10
<b>第2章 CAPSULE SERVER のインストール</b>	<b>11</b>
2.1. SATELLITE SERVER への登録	11
2.2. SATELLITE INFRASTRUCTURE サブスクリプションのタッチ	12
2.3. リポジトリの設定	13
2.4. CAPSULE SERVER パッケージのインストール	14
2.5. CHRONYD とシステムクロックの同期	14
2.6. SSL 証明書を使用した CAPSULE SERVER の設定	15
2.7. SATELLITE WEB UI での適切な組織および場所の CAPSULE SERVER への割り当て	15
2.7.1. デフォルトの SSL 証明書を使用した Capsule Server の設定	16
2.7.2. カスタム SSL 証明書を使用した Capsule Server の設定	17
2.7.2.1. Capsule Server のカスタム SSL 証明書の作成	17
2.7.2.2. カスタムの SSL 証明書の Capsule Server へのデプロイ	19
2.7.2.3. ホストへの カスタム SSL 証明書のデプロイ	21
<b>第3章 CAPSULE SERVER での追加設定の実行</b>	<b>22</b>
3.1. KATELLO エージェントのインストール	22
3.2. 外部 CAPSULE での OPENSCAP の有効化	22
3.3. CAPSULE SERVER へのライフサイクル環境の追加	23
3.4. 管理対象ホスト上での電源管理の有効化	24
3.5. CAPSULE SERVER での DNS、DHCP、および TFTP の設定	24
3.6. MONGOD へのアクセスの制限	25
<b>第4章 外部サービスを使用した CAPSULE SERVER の設定</b>	<b>27</b>
4.1. 外部 DNS を使用した CAPSULE SERVER の設定	27
4.2. CAPSULE SERVER での外部 DHCP の設定	28
4.2.1. Capsule Server を使用するための外部 DHCP サーバーの設定	28
4.2.2. 外部 DHCP サーバーを使用した Capsule Server の設定	31
4.3. CAPSULE SERVER での外部 TFTP の設定	32
4.4. 外部 IDM DNS を使用した CAPSULE SERVER の設定	33
4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定	33
4.4.2. TSIG 認証を使用した動的 DNS 更新の設定	37
4.4.3. 内部 DNS サービス使用への復元	39
<b>付録A CAPSULE SERVER のスケーラビリティに関する考慮事項</b>	<b>41</b>



## 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のオペレーティングシステムには、以下の要件が適用されます。

- x86\_64 アーキテクチャー
- Red Hat Enterprise Linux 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 を実行する機能として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
- qpidd
- qdrouterd
- squid
- foreman-proxy
- puppet
- puppetserver

Capsule Server のスケーリングの詳細は、[Capsule Server のスケーラビリティに関する考慮事項](#) 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の両方で完全にサポートされています。認定ハイパーバイザーに関する詳細は、[Which hypervisors are certified to run Red Hat Enterprise Linux?](#) を参照してください。

## SELinux モード

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

## FIPS モード

FIPS モードで稼働する Red Hat Enterprise Linux システムに、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、想定ユースケースシナリオに基づいており、各環境ごとに異なることがあります。

ランタイムサイズは Red Hat Enterprise Linux 6、7、および 8 のリポジトリ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	32 GB (最小)
/var/lib/pulp/	1 MB	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/spool/squid/	0 GB	10 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させる場合は、以下のガイドラインを考慮してください。

- **/tmp** ディレクトリーを別のファイルシステムとしてマウントする場合は、**/etc/fstab** ファイルの **exec** マウントオプションを使用する必要があります。**/tmp** が、**noexec** オプションを指定してすでにマウントされている場合は、オプションを **exec** に変更して、ファイルシステムを再マウントする必要があります。これは、**puppetserver** サービスが機能するために必要です。
- 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/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。



- `/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/lib/pulp/` ディレクトリーと `/var/lib/mongodb/` ディレクトリーには、高帯域幅で低レイテンシーのストレージの使用をお勧めします。Red Hat Satellite には I/O を大量に使用する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。**fiio** ツールを使用すると、このデータが取得できます。**fiio** 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 Operations](#) を参照してください。

## ファイルシステムのガイドライン

- 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、MongoDB 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**、**noatime**、および **noauto** 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。

## ログファイルのストレージ

ログファイルは、`/var/log/messages/`、`/var/log/httpd/`、および `/var/lib/foreman-proxy/openscap/content/` に書き込まれます。`logrotate` を使って、これらのファイルのサイズを管理できます。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。

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

## NFS マウントを使用する場合の 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

同じパッケージが異なるリポジトリーで重複して存在する場合には、ディスク上に一度しか保存されません。そのため、重複するパッケージを別のリポジトリーに追加するときに必要な追加ストレージが少

なくて済みます。ストレージの多くは、**/var/lib/mongodb/** および **/var/lib/pulp/** ディレクトリーにあります。これらのエンドポイントは手動で設定できません。ストレージの問題を回避するために、ストレージが **/var** ファイルシステムで利用可能であることを確認してください。

## 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存するために使用します。同期タスクがすべて完了したら、コンテンツは **/var/lib/pulp/** ディレクトリーに移動されます。

RPM 形式のコンテンツの場合は、各 RPM ファイルは同期後に **/var/lib/pulp** ディレクトリーに移動されます。一度に、**/var/cache/pulp/** ディレクトリーに保存される RPM ファイルは 5 つです。デフォルトでは、RPM コンテンツの同期タスクは最大 8 つまで同時に実行でき、それぞれ最大 1 GB のメタデータを使用します。

## 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/foreman/** 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。

## 同期された RHEL ISO

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

## 1.4. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする方法であれば他の方法でもインストールできます。Red Hat Capsule Server は、Capsule Server 6.9 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が含む Red Hat Enterprise Linux インストールが必要です。他のパッケージセットの変更や、サーバーの運用に直接必要でないサードパーティーの設定やソフトウェアは含めないようにしてください。この制限は、ハード化や Red Hat 以外の他社のセキュリティソフトウェアが該当します。機能強化や Red Hat 以外のセキュリティ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。

新しくプロビジョニングしたシステムに Capsule Server をインストールします。

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

## 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オペレーティングシステムが含まれます。

## Capsule のクライアント

Satellite と統合された Capsule ではない Capsule のクライアントであるホストには、Satellite Server へのアクセスは必要ありません。Satellite トポロジーの詳細は [Red Hat Satellite 6 の計画の Capsule のネットワーク](#) を参照してください。

使用している設定に応じて、必要なポートは変わることがあります。

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 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
80	TCP	HTTP	Anaconda、yum、および Katello 証明書更新の取得向け
443	TCP	HTTPS	Katello、Satellite、Satellite API、および Pulp への接続
5000	TCP	HTTPS	Docker レジストリーのための Katello 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

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 証明書更新の取得向け
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ーと通信する 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 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)

ポート	プロトコル	サービス	用途
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オペレーティング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	<ul style="list-style-type: none"> <li>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)</li> <li>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。</li> </ul>



### 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が、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、応答が **あれば** IP アドレスは割り当てられません。

## 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ことから、クライアントが Satellite との通信用に接続できるように、Satellite Server でファイアウォールルールが設定されていること。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. 以下のコマンドを入力します。

```
# 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オペレーティングシステムを 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オペレーティング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オペレーティングシステムのリポジトリと、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に関する詳しい情報は、[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 のベースオペレーティング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 のベースオペレーティング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オペレーティングシステムには、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である。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オペレーティングシステムに、**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 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 のベースオペレーティングシステムを Capsule が所属する環境に登録します。アクティベーションキーを使用すると、その環境を簡単に指定できます。アクティベーションキーの詳細は、[コンテンツ管理ガイドの アクティベーションキーの管理](#) を参照してください。

```
# 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ップ

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ップ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 (RHSC) 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 ナレッジベースのソリューション [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割り当てられていない場合には、利用可能として分類されます。利用可能な Satellite サブスクリプション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します。

### 手順

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

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

このコマンドを実行すると、以下のような出力が表示されます。

```
Subscription Name: Red Hat Satellite Infrastructure Subscription
Provides:          Red Hat Satellite
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat CodeReady Linux Builder for x86_64
                  Red Hat 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
```

- 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。
- Capsule Server の実行先のベースオペレーティングシステムに、Satellite Infrastructure サブスクリプションを割り当てます。

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

このコマンドを実行すると、以下のような出力が表示されます。

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

- 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を有効にします。

### 手順

必要なリポジトリを設定するには、以下の手順を実行します。

- すべてのリポジトリを無効にします。

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

- 以下のリポジトリを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-satellite-capsule-6.9-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-tools-6.9-rpms \
--enable=rhel-server-rhsc-7-rpms \
--enable=rhel-7-server-ansible-2.9-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** リポジトリも有効にして、RHV ゲストエージェントとドライバーをインストールする必要があります。詳細は **Virtual Machine Management Guide** の [Installing the Guest Agents and Drivers on Red Hat Enterprise Linux](#) を参照してください。

3. メタデータを消去します。

```
# yum clean all
```

4. オプション: 必要なりポジトリが有効になっていることを確認します。

```
# yum repolist enabled
```

## 2.4. 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、ベースオペレーティング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
```

## 2.5. 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オペレーティングシステムのシステムクロックを Network Time Protocol (NTP) サーバーと同期する必要があります。ベースオペレーティングシステムのクロックが正しく設定されていない場合には、証明書の検証に失敗する可能性があります。

**chrony** スイートに関する詳細は、**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 設定](#) を参照してください。

### 手順

1. **chrony** パッケージをインストールします。

```
# 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
```

## 2.6. 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。

## 2.7. SATELLITE WEB UI での適切な組織および場所の CAPSULE SERVER への割り当て

Capsule Server パッケージのインストール後に、組織または場所が複数ある場合には、Satellite Web UI で Capsule に正しい組織と場所を割り当てて Capsule が表示されるようにする必要があります。

### 手順

1. Satellite Web UI にログインします。
2. 画面左上にある **組織** リストから、**任意の組織** を選択します。
3. 画面左上にある **場所** リストから、**任意の場所** を選択します。
4. **ホスト > すべてのホスト** に移動し、Capsule Server を選択します。
5. **アクションの選択** リストから、**組織の割り当て** を選択します。
6. **組織の選択** リストから、この Capsule を割り当てる組織を選択します。
7. **Fix Organization on Mismatch (組織の不一致についての修正)** をクリックします。
8. **送信** をクリックします。
9. Capsule Server を選択します。**アクションの選択** リストから、**場所の割り当て** を選択します。
10. **場所** リストから、この Capsule を割り当てる場所を選択します。
11. **Fix Location on Mismatch (場所の不一致についての修正)** をクリックします。
12. **送信** をクリックします。
13. **管理 > 組織** の順に移動して、Capsule を割り当てた組織をクリックします。
14. **Capsules** タブをクリックし、Capsule Server が **選択項目** リストに表示されていることを確認し、**送信** をクリックします。
15. **管理 > 場所** の順に移動して、Capsule を割り当てた場所をクリックします。

16. **Capsules** タブをクリックし、Capsule Server が **選択項目** リストに表示されていることを確認し、**送信** をクリックします。

## 結果

必要に応じて、Capsule Server が Satellite Web UI に正しく表示されているかどうかを検証できるようになりました。

1. **組織** タブで、組織を選択します。
2. **場所** リストから場所を選択します。
3. **ホスト** > **すべてのホスト** に移動します。
4. **インフラストラクチャー** > **Capsules** に移動します。

### 2.7.1. 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

```
output omitted
satellite-installer \
--scenario capsule \
--certs-tar-file "/root/capsule_cert/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_cert/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

証明書のデプロイ後に、証明書のアーカイブファイル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。

## 2.7.2. 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」](#)

### 2.7.2.1. 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 Capsule Server 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 Server と Capsule Server 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 ] ❶
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 ❷

[ 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 ❸
```

❶ **[ req\_distinguished\_name ]** セクションに、貴社の組織の情報を入力します。

❷ 証明書の共通ネーム **CN** を、Capsule Server またはワイルドカードの \* 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。ワイルドカードの値を設定しており、**katello-certs-check** コマンドを使用する場合には、**-t capsule** オプションを追加する必要があります。

- 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。同じ証明局を使用して Satellite Server と Capsule Server の証明書を署名する必要がある。  
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。

### 2.7.2.2. 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 と Capsule が以下の条件を満たすことを確認してください。

- 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 \
```



```
-t capsule -c /root/capsule_cert/capsule_cert.pem \ ❶
-k /root/capsule_cert/capsule_cert_key.pem \ ❷
-b /root/capsule_cert/ca_cert_bundle.pem ❸
```

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

`/root/capsule_cert/openssl.cnf` 設定ファイルの証明書の共通ネーム **CN =** に、\* のワイルドカードの値を設定した場合には、**katello-certs-check** コマンドに **-t capsule** オプションを追加する必要があります。

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 **\$CAPSULE** を Capsule Server の FQDN に変更します。
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

```
output omitted
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" \
```

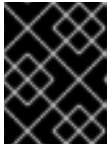


```
--foreman-proxy-trusted-hosts      "satellite.example.com"\
--foreman-proxy-trusted-hosts      "capsule.example.com"\
--foreman-proxy-oauth-consumer-key  "s97QxvUAgFNAQZNGg4F9zLq2biDsxM7f"\
--foreman-proxy-oauth-consumer-secret "6bpzAdMpRAfYaVZtaepYetomgBVQ6ehY"\
--puppet-server-foreman-url          "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_cert/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

証明書のデプロイ後に、証明書のアーカイブファイル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。

## 2.7.2.3. 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。

### 手順

- 各ホストに **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してください。



#### 注記

Katello エージェントは非推奨で、今後の Satellite のバージョンで削除されます。プロセスを移行して、リモート実行機能を使用してクライアントをリモートで更新します。詳細は、[ホストの管理ガイドの Goferd および Katello エージェントを使用しないホスト管理](#) を参照してください。

**katello-agent** パッケージは、**goferd** service を提供する **gofer** パッケージによって異なります。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 6.9 リポジトリを有効化しておく。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 6.9 リポジトリの有効化](#) を参照してください。
- Satellite Server で、Satellite Tools 6.9 リポジトリを同期しておく。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 6.9 リポジトリの同期](#) を参照してください。
- クライアントで Satellite Tools 6.9 リポジトリを有効にしておく。

#### 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. **katello-agent** 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. 外部 CAPSULE での OPENSAP 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
```

## 3.3. 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。

**ライブラリー** ライフサイクル環境は、CDN がリポジトリを更新するたびに自動的に Capsule が同期をトリガーするため、Capsule Server に割り当てないでください。自動で同期される場合、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. 左側のメニューから、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 **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。  
同期の各タイプの定義については、**コンテンツ管理ガイド**の [リポジトリの復旧](#) 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 をメモします。

2. ID を使用して、Capsule の詳細を確認します。

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

3. 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、ID と組織名を書き留めます。

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

4. 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" \
--lifecycle-environment-id lifecycle-environment_id
```

Capsule Server に追加するライフサイクル環境ごとに繰り返します。

#### 5. 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、以下のコマンドを入力します。

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

- Satellite Server から Capsule Server に特定のライフサイクル環境を同期するには、以下のコマンドを入力します。

```
# hammer capsule content synchronize --id external_capsule_id \
--lifecycle-environment-id lifecycle-environment_id
```

### 3.4. 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イスが必要である。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 \(BMC\) インターフェイスの追加](#) を参照してください。

#### 手順

- BMC を有効にするには、以下のコマンドを入力します。

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

### 3.5. CAPSULE SERVER での DNS、DHCP、および TFTP の設定

DNS、DHCP および TFTP サービスを Capsule Server で設定するには、お使いの環境に適したオプションを指定して **satellite-installer** コマンドを使用します。設定可能なオプションの全リストを表示するには、**satellite-installer --scenario satellite --help** コマンドを入力します。

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。

代わりに外部の DNS、DHCP および TFTP サービスを使用するには、[4章 外部サービスを使用した Capsule Server の設定](#) を参照してください。

#### Multihomed DHCP の詳細の追加

Multihomed DHCP を使用する場合は、ネットワークインターフェイスファイルの更新が必要です。

1. `/etc/systemd/system/dhcpd.service.d/interfaces.conf` ファイルで、以下の行を編集して Multihomed DHCP を追加します。

```
[Service]
ExecStart=/usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid eth0
eth1 eth2
```

このファイルがまだ存在しない場合は作成します。

2. 以下のコマンドを入力して、デーモンのリロードを実行します。

```
# systemctl --system daemon-reload
```

3. 以下のコマンドを入力して、**dhcpd** サービスを再起動します。

```
# systemctl restart dhcpd.service
```

### 前提条件

- 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イス名 (**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 サービスの設定に関する情報は、[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. MONGODB 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、ファイアウォール設定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、ファイアウォール設定を更新します。

```
# 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 外部サービスを使用した CAPSULE SERVER の設定

Capsule Server で DNS、DHCP、および TFTP サービスを設定しない場合は、外部 DNS、DHCP、および TFTP サービスと連携させる Capsule Server の設定のセクションを使用します。

### 4.1. 外部 DNS を使用した CAPSULE SERVER の設定

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

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

#### 前提条件

- 外部 DNS サーバーが設定されている必要がある。

#### 手順

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

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

6. **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
```

7. foreman-proxy サービスを再起動します。

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

8. Satellite Server Web UI にログインします。
9. インフラストラクチャー > Capsules に移動し、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。
10. 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 設定とリースファイルを 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。



#### 注記

外部の DHCP サーバーとして dnsmasq を使用する場合には、**dhcp-no-override** の設定を有効にします。Satellite は **grub2/** サブディレクトリーの配下にある TFTP サーバーに設定ファイルを作成するので、この設定を必ず有効にしてください。**dhcp-no-override** 設定が無効な場合には、クライアントは root ディレクトリーからブートローダーと設定をフェッチするのでエラーが発生する可能性があります。

#### 手順

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つのファイルで設定されるキーペアが現在のディレクトリーに作成されます。

3. キーからシークレットハッシュをコピーします。

```
# 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

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. インフラストラクチャー > **Capsules** に移動し、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。
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

1. NFS 用に TFTP ディレクトリーを作成します。

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

2. **/etc/fstab** ファイルで以下の行を追加します。

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

3. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

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

```
# satellite-installer --foreman-proxy-tftp=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、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. 外部 IDM DNS を使用した CAPSULE SERVER の設定

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

Capsule Server 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。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。

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

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

内部 DNS サービスに戻すには、次の手順を使用します。

- [「内部 DNS サービス使用への復元」](#)



### 注記

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

### 4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

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

をインストールする必要があります。

## 前提条件

- IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されている。詳細は [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。
- IdM サーバーの管理者に問い合わせて、IdM サーバーでゾーンを作成するパーミッションが割り当てられた、IdM サーバーのアカウントを取得する。
- デプロイメントに DNS サービスを提供するように Satellite Server または Capsule Server が設定されていることを確認する。
- デプロイメントの DNS サービスを管理する Satellite または Capsule のいずれかのベースオペレーティングシステムで DNS、DHCP および TFTP サービスを設定する必要がある。
- 応答ファイルのバックアップを作成しておく。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、[Satellite Server の設定](#) を参照してください。

## 手順

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

### IdM サーバーでの Kerberos プリンシパルの作成

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

```
# kinit idm_user
```

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

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

### IdM クライアントのインストールおよび設定

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

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

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

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

3. Kerberos チケットを取得します。

```
# kinit admin
```

4. 既存の **keytab** を削除します。

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

5. このシステムの **keytab** を取得します。

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



#### 注記

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

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

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

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

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

### IdM Web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。

- a. **Network Services (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。(例: **example.com**)
- c. **Add and Edit** をクリックします。
- d. 設定タブをクリックして **BIND 更新ポリシー** 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** を **True** に設定します。
- f. **Allow PTR sync** を有効にします。
- g. **送信** をクリックして、変更を保存します。

2. 逆引きゾーンを作成して設定します。

- a. **Network Services (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** をクリックします。
- c. **Reverse zone IP network** を選択して、CIDR 形式でネットワークアドレスを追加し、逆引き参照を有効にします。
- d. **Add and Edit** をクリックします。

- e. **設定** タブの **BIND 更新ポリシー** 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** を **True** に設定します。
- g. **送信** をクリックして、変更を保存します。

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

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

- Satellite で以下のコマンドを入力します。

```
satellite-installer --scenario satellite \
--foreman-proxy-dns=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 で、以下のコマンドを実行します。

```
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
```

2. Satellite または Capsule のプロキシサービスを再起動します。

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

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

## Satellite Web UI での設定更新

1. **インフラストラクチャー** > **Capsules** に移動し、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。
2. 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。
3. 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 サーバーを使用して管理するドメインを選択します。
  - d. **Capsules** タブで、**Reverse DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
  - e. **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

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

##### 前提条件

- IdM サーバーがデプロイされ、ホストベースのファイアウォールが正確に設定されている。詳細は [Linux 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。
- IdM サーバーで **root** 権限を取得する必要があります。
- デプロイメントに DNS サービスを提供するように Satellite Server または Capsule Server が設定されていることを確認する。
- デプロイメントの DNS サービスを管理する Satellite または Capsule のいずれかのベースオペレーティングシステムで DNS、DHCP および TFTP サービスを設定する必要がある。
- 応答ファイルのバックアップを作成しておく。応答ファイルが破損した場合に、元の状態に戻せるように、バックアップを使用できます。詳細は、[Satellite Server の設定](#) を参照してください。

##### 手順

TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

##### IdM サーバーの DNS ゾーンに対する外部更新の有効化

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

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

2. **named** サービスをリロードして、変更を有効にします。

```
# systemctl reload named
```

3. IdM Web UI で、**ネットワークサービス > DNS > DNS ゾーン** に移動して、ゾーンの名前をクリックします。設定 タブで、以下の変更を適用します。

- a. **BIND update policy (BIND 更新ポリシー)** ボックスで以下の内容を追加します。

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

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

- c. **更新** をクリックして変更を保存します。

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

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

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

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

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

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

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

```
# satellite-installer --scenario satellite \
--foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="IdM_Server_IP_Address" \
--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更新のテスト

1. **bind-utils** ユーティリティー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キーファイルと同じであることを確認します。

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

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

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

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

#### 手順

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

#### DNS サーバーとしての Satellite または Capsule の設定

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

**# satellite-installer**

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

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

### Satellite Web UI での設定更新

1. インフラストラクチャー > Capsules に移動します。
2. 更新する各 Capsule で、アクション リストから 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 に移動して、設定するドメイン名をクリックします。
  - b. ドメイン タブで、DNS Capsule を、サブネットの接続先の Capsule に設定します。
4. サブネットを設定します。
  - a. 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 で管理するドメインを選択します。
  - d. Capsules タブで、Reverse DNS Capsule を、サブネットの接続先の Capsule に設定します。
  - e. 送信 をクリックして変更を保存します。

## 付録A CAPSULE SERVER のスケーラビリティに関する考慮事項

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 では、ある時点で同時実行される Puppet エージェントの上限数が 100 となっています。100 を超える Puppet エージェントを同時実行すると、503 HTTP エラーが発生します。

たとえば、実行が終了してから、次の実行が開始されるまでの任意のタイミングで、同時に実行される Puppet エージェントが 100 台未満で、Puppet エージェントの実行が均等に分散されていると仮定した場合に、CPU が 4 つ割り当てられた Capsule Server は最大で 1250-1600 台の Puppet クライアントに対応し、各 Puppet クライアントに、中程度のワークロードである 10 個の 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より、Satellite のインストールは、Capsule Server の数をスケールアウトしてサポートします。

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。

- 外部 Puppet クライアントには、Satellite 6 統合 Capsule に直接報告するものではありません。
- 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 エージェント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、Passenger の要求キューがいっぱいにな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の制限を示しています。

表A.14 CPU を使用した場合の Puppet のスケーラビリティ

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

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