



Red Hat Satellite 6.3

インストールガイド

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

Red Hat Satellite 6.3 インストールガイド

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

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

法律上の通知

Copyright © 2018 Red Hat, Inc.

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

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

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

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

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

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

The OpenStack ® 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 Server および Capsule Server のインストール方法、初期設定の実行方法、および外部サービスの設定方法を説明します。

目次

第1章 SATELLITE SERVER と CAPSULE SERVER の機能	5
第2章 インストールのための環境準備	6
2.1. システム要件	6
2.2. ストレージの要件と推奨事項	6
2.3. サポート対象オペレーティングシステム	11
2.4. サポート対象ブラウザ	11
2.5. ポートとファイアウォールの要件	12
2.6. クライアントから SATELLITE SERVER への接続の有効化	16
2.7. CAPSULE SERVER から SATELLITE SERVER への接続の有効化	17
2.8. SATELLITE SERVER およびクライアントから CAPSULE SERVER への接続の有効化	17
2.9. ファイアウォール設定の確認	18
2.10. DNS 解決の検証	18
2.11. デフォルトの SELINUX ポートの変更	19
第3章 SATELLITE SERVER のインストール	21
3.1. 接続済みネットワークからの SATELLITE SERVER のインストール	21
3.1.1. Red Hat Subscription Management への登録	21
3.1.2. Satellite サブスクリプションを識別してホストへの割り当て	22
3.1.3. リポジトリの設定	24
3.1.4. Satellite サーバーパッケージのインストール	25
3.2. 切断されたネットワークからのダウンロードおよびインストール	25
3.2.1. バイナリー DVD イメージのダウンロード	26
3.2.2. オフラインリポジトリでベースシステムの設定	26
3.2.3. オフラインリポジトリからのインストール	28
3.2.4. パッケージの手動ダウンロード	29
3.3. 初期設定の実行	29
3.3.1. 時間の同期	29
3.3.2. ホストオペレーティングシステムへの SOS パッケージのインストール	30
3.3.3. インストールオプションの指定	30
3.3.3.1. 手動による初期設定	31
3.3.3.2. 応答ファイルを使用した初期設定の自動実行	32
3.3.4. カスタマーポータルでサブスクリプション割り当ての作成	33
3.3.5. 割り当てへのサブスクリプションの追加	33
3.3.6. カスタマーポータルからのサブスクリプションマニフェストのエクスポート	33
3.3.6.1. Satellite Server へのサブスクリプションマニフェストのインポート	34
3.4. 追加設定の実行	34
3.4.1. Satellite Tools リポジトリのインストール	34
3.4.2. HTTP プロキシを使用した Satellite Server の設定	35
3.4.3. 管理対象ホスト上での電源管理の有効化	36
3.4.4. Satellite Server で DNS、DHCP、および TFTP の設定	37
3.4.5. 管理対象外ネットワークに対して DNS、DHCP、および TFTP の無効化	38
3.4.6. Satellite Server で送信メールの設定	39
3.4.7. カスタムサーバー証明書を使用した Satellite Server の設定	41
3.4.7.1. Satellite Server 向けの SSL 証明書を取得	42
3.4.7.2. Satellite Server の SSL 証明書の検証	43
3.4.7.3. カスタム証明書パラメーターを使用した Satellite インストーラーの実行	45
3.4.7.4. Satellite Server に接続されたすべてのホストへの新しい証明書のインストール	46
3.4.8. mongod へのアクセスの制限	46
第4章 CAPSULE SERVER のインストール	48
4.1. SATELLITE SERVER への CAPSULE SERVER の登録	48

4.2. CAPSULE SERVER サブスクリプションの識別と割り当て	48
4.3. リポジトリの設定	49
4.4. 時間の同期	50
4.5. CAPSULE SERVER のインストール	51
4.6. CAPSULE SERVER の初期設定の実行	51
4.6.1. デフォルトのサーバー証明書を使用した Capsule Server の設定	51
4.7. CAPSULE SERVER での追加設定の実行	52
4.7.1. katello エージェントのインストール	52
4.7.2. Capsule Server でリモート実行の有効化	53
4.7.3. Capsule Server へのライフサイクル環境の追加	53
4.7.4. 管理対象ホスト上での電源管理の有効化	54
4.7.5. Capsule Server での DNS と DHCP の設定	55
4.7.6. カスタムサーバー証明書を使用した Capsule Server の設定	56
4.7.6.1. Capsule Server 向けの SSL 証明書の取得	56
4.7.6.2. Capsule Server の SSL 証明書の検証	58
4.7.6.3. Capsule サーバーの証明書アーカイブファイルの作成	59
4.7.6.4. Capsule Server のカスタム証明書のインストール	60
4.7.6.5. すべてのホストへの Capsule Server の新しい証明書のインストール	61
4.7.7. mongod へのアクセスの制限	61
第5章 外部サービスの設定	64
5.1. 外部 DNS を使用した SATELLITE の設定	64
5.2. DNS サービスの開始と起動	66
5.3. CAPSULE SERVER での外部 DNS の設定	66
5.4. SATELLITE SERVER での外部 DHCP の設定	67
5.5. CAPSULE SERVER での外部 DHCP の設定	71
5.6. SATELLITE SERVER での外部 TFTP の設定	73
5.6.1. ファイアウォールでの TFTP への外部アクセスの設定	74
5.7. CAPSULE SERVER での外部 TFTP の設定	74
5.8. SATELLITE での外部 IDM DNS の設定	75
5.8.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定	75
5.8.2. TSIG 認証を使用した動的 DNS 更新の設定	79
5.8.3. 内部 DNS サービス使用への復元	81
第6章 SATELLITE SERVER および CAPSULE SERVER のアンインストール	83
6.1. SATELLITE SERVER のアンインストール	83
6.2. CAPSULE SERVER のアンインストール	84
第7章 詳細情報の提供元	86
第8章 AMAZON WEB SERVICES 上での RED HAT SATELLITE の実行	87
8.1. ユースケースにおける留意点	87
8.1.1. 機能するユースケース	87
8.1.2. 機能しないユースケース	88
8.2. デプロイメントシナリオ	88
8.3. 前提条件	91
8.3.1. Amazon Web Service の前提条件	91
8.3.2. Red Hat Cloud の前提条件	92
8.3.3. Red Hat Satellite 固有の前提条件	92
8.3.4. Red Hat Satellite のインストール準備	93
8.4. AWS での SATELLITE SERVER のインストール	93
8.5. AWS での CAPSULE のインストール	93
8.6. ブートストラップスクリプトを使ったホストの SATELLITE への登録	93

付録A 大規模デプロイメントに関する考慮事項	94
付録B CAPSULE SERVER のスケーラビリティに関する考慮事項	98
付録C RED HAT SATELLITE へのカスタム設定の適用	100
C.1. PUPPET 実行で上書きされた手動変更の復元	100

第1章 SATELLITE SERVER と CAPSULE SERVER の機能

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

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

Capsule Server により中央サーバーの負荷が減少し、冗長性が増加し、帯域幅の使用率が低下します。詳細は『[Capsule Server の概要](#)』を参照してください。

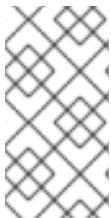
第2章 インストールのための環境準備

2.1. システム要件

ネットワーク接続されたベースシステムには、以下の要件が適用されます。

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

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



注記

Red Hat Satellite Server と **Capsule Server** のバージョンは一致する必要があります。たとえば、**Satellite 6.2 Server** は **6.3 Capsule Server** を実行できず、**Satellite 6.3 Server** は **6.2 Capsule Server** を実行できません。**Satellite Server** と **Capsule Server** のバージョンが一致しないと、警告なしで **Capsule Server** が失敗します。



注記

自己登録の **Satellites** はサポートされません。

大量のコンテンツホストがある場合は「[付録A 大規模デプロイメントに関する考慮事項](#)」を参照して、環境が適切に設定されていることを確認してください。

Capsule Server のスケーリングの詳細は「[付録B 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。

2.2. ストレージの要件と推奨事項

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

異なるリポジトリ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に必要な追加ストレージが少なくなります。ストレージの多くは、**/var/lib/mongodb/** ディレクトリおよび **/var/lib/pulp/** ディレクトリに使用されま

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。

/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** を実行できます (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 形式のコンテンツの場合、1 つの同期タスクに対する ISO ファイルはすべて、タスクが完了するまで **/var/cache/pulp/** に格納されます (タスクの完了後は、**/var/lib/pulp/** ディレクトリーに移動します)。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。同期する ISO ファイルの数を考慮してください (これらのファイルに必要な一時ディスク容量は通常 RPM コンテンツのサイズを超えます)。

/var/lib/qpidd/ ディレクトリーでは、1 つのコンテンツホスト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。

ストレージ要件

以下の表には、特定のディレクトリーに推奨されるストレージ要件が詳述されています。これらの値は、期待されるユースケースシナリオに基づき、個別の環境に応じて異なることがあります。**Satellite Server** にはデフォルトで統合 **Capsule** が含まれるので、**Capsule Server** の表は **Satellite Server** にも適用されます。表を参照する際には、ご自分のユースケースに留意してください。たとえば、**Capsule Server** で **Pulp** を有効にしていない場合、**/var/lib/pulp/** のような **Pulp** に関連するディレクトリーのストレージ要件については、ここに記載されているものと同一である必要はありません。

以下の 2 つの表では、ランタイムサイズは Red Hat Enterprise Linux 5、6、および 7 のリポジトリと同期して測定されています。

表2.1 Satellite Server インストールのストレージ要件

フォルダー	インストールサイズ	ランタイムサイズ	留意事項
/var/cache/pulp/	1M バイト	10 GB (接続インストールの最小値)	本項の概要にある記述を参照してください。
/var/cache/pulp/	1M バイト	30 GB (非接続インストールの最小値)	本項の概要にある記述を参照してください。
/var/lib/pulp/	1 MB	500 GB	<ul style="list-style-type: none"> コンテンツが Satellite Server に追加されると、継続的に増加します。長期にわたる増加を計画してください。 シンボリックリンクは使用できません。

フォルダー	インストールサイズ	ランタイムサイズ	留意事項
/var/lib/mongodb/	3.5 GB	50 GB	<ul style="list-style-type: none"> • コンテンツが Satellite Server に追加されると、継続的に増加します。長期にわたる増加を計画してください。 • シンボリックリンクは使用できません。 • MongoDB では NFS は推奨されません。
/var/log/	10 MB	250 MB	なし
/var/lib/pgsql/	100 MB	10 GB	<p>/var/lib/pgsql/ に最小 2 GB の利用可能なストレージがあること。さらに、データストレージ要件の増加に伴ってこのディレクトリーを含むパーティションを拡張できること。</p> <p>PostgreSQL で NFS を使用することは推奨されません。</p>
/usr	3 GB	適用外	なし
/opt	500 MB (接続されたインストール)	適用外	<p>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 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。</p>

フォルダー	インストールサイズ	ランタイムサイズ	留意事項
/opt	3 GB (非接続インストール)	適用外	<ul style="list-style-type: none"> ソフトウェアコレクションは、/opt/rh/ディレクトリーと/opt/theoforeman/ディレクトリーにインストールされます。/optディレクトリーへのインストールには、rootによる書き込みパーミッションおよび実行パーミッションが必要です。 インストールに使用されるリポジトリのコピーは、このディレクトリーに格納されます。

表2.2 Capsule Server インストールのストレージ要件

フォルダー	インストールサイズ	ランタイムサイズ	留意事項
/var/cache/pulp/	1M バイト	10 GB (最小)	本項の概要にある記述を参照してください。
/var/lib/pulp/	1 MB	500 GB	<ul style="list-style-type: none"> コンテンツが追加されると、継続的に増加します。長期にわたる増加を計画してください。 シンボリックリンクは使用できません。

フォルダー	インストールサイズ	ランタイムサイズ	留意事項
/var/lib/mongodb/	3.5 GB	50 GB	<ul style="list-style-type: none"> コンテンツが追加されると、継続的に増加します。長期にわたる増加を計画してください。 シンボリックリンクは使用できません。 MongoDB では NFS は推奨されません。

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『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。

ストレージの推奨事項

- ほとんどの **Satellite Server** データと **Capsule Server** データは `/var` ディレクトリーに格納されるため、システムがスケーラブルになるよう `/var` を LVM ストレージにマウントすることを強くお勧めします。
- `/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 を大量に使用する多くの操作があるため、高レイテンシーで低帯域幅のストレージを使用すると、パフォーマンス低下の問題が発生します。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 6 Operations](#)」を参照してください。
- MongoDB はデータファイルにアクセスするために通常の I/O を使用せず、データファイルとジャーナルファイルが NFS でホストされた場合にパフォーマンスの問題が発生するため、MongoDB とともに NFS を使用することは推奨されません。NFS を使用する必要がある場合は、`/etc/fstab` ファイルで `bg`、`noatime`、および `noatime` のオプションを使用してボリュームをマウントします。
- 入出力レイテンシーが高すぎるため、GFS2 ファイルシステムは使用しないでください。
- パフォーマンスを向上させるには、HDD (Hard Disk Drive) ではなく SSD (Solid State Drive) を使用します。
- XFS ファイルシステムは、`ext4` では存在する `inode` の制限がないため、Red Hat Satellite 6 に推奨されます。Satellite は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。
- 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
```

2.3. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする他の任意の方法でインストールできます。Red Hat Satellite Server と Red Hat Satellite Capsule Server は、Satellite 6.3 のリリース時に利用可能な Red Hat Enterprise Linux 7 Server の最新バージョンでのみサポートされています。EUS または z-stream を含む Red Hat Enterprise Linux の以前のバージョンはサポートされません。

Red Hat Satellite Server および Red Hat Satellite Capsule Server には、@Base パッケージグループを含む Red Hat Enterprise Linux インストールが必要です。他のパッケージセットの変更や、サーバーの直接的な運用に直接必要でないサードパーティーの構成やソフトウェアは含めないようにしてください。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、Satellite Server が完全に機能することを最初に確認し、その後でシステムのバックアップを作成して、Red Hat 以外のソフトウェアを追加します。

Satellite Server システムは新しくプロビジョニングすることが推奨されます。また、Capsule Server も新しくプロビジョニングし、Red Hat CDN に登録されていないシステムであることが推奨されます。Satellite を実行する以外の目的でシステムを使用することはサポートされません。

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

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

2.4. サポート対象ブラウザー

以下の Web ブラウザーは完全にサポートされます。

- Firefox バージョン 39 以降
- Chrome バージョン 28 以降

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

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



注記

Satellite Server の Web UI とコマンドラインインターフェースは、英語、ポルトガル語、中国語 (簡体)、中国語 (繁体)、韓国語、日本語、イタリア語、スペイン語、ロシア語、フランス語、ドイツ語に対応しています。

2.5. ポートとファイアウォールの要件

Satellite アーキテクチャーのコンポーネントが通信できるようにするには、特定のネットワークポートがベースオペレーティングシステムでオープンかつフリーの状態であり、ネットワークベースファイアウォールでオープンである必要があります。本項の表は、ポートの用途を説明しています。ホストベースのファイアウォールに対応するファイアウォールコマンドは、以下の項に記載されています。インストール開始前に Satellite Server と Capsule Server 間のポートを開いておかないと、Capsule Server のインストールに失敗します。

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



注記

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下の表のコンテキストでは Satellite のクライアントになります。これには、Capsule Server が実行されているベースシステムが含まれます。ネットワークベースのファイアウォール設定を計画している場合は、このことを考慮してください。

Satellite と統合された Capsule ではなく、Capsule のクライアントであるシステムには、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 アーキテクチャーガイド』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。

表2.3 Red Hat CDN 通信に対する Satellite のポート

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

Satellite がネットワークから切断されている場合を除き、Satellite Server には Red Hat CDN へのアクセスが必要になります。Red Hat CDN (cdn.redhat.com) で使用されている IP アドレスの一覧は、Red Hat カスタマーポータルナレッジベース記事「[Red Hat が公開している CIDR の一覧](#)」を参照してください。

表2.4 Satellite へのブラウザーベースユーザーインターフェース向けポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Satellite へのブラウザーベース UI アクセス

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Satellite に Web UI でアクセスするための HTTPS へのリダイレクション (オプション)

表2.5 Satellite に通信するクライアント向けポート

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Anaconda、yum、Katello 証明書およびテンプレートの取得向け、iPXE ファームウェアのダウンロード向け
443	TCP	HTTPS	サブスクリプション管理サービス、yum、Telemetry サービス、Katello エージェントへの接続向け
5647	TCP	amqp	Satellite の Qpid ディスパッチャーと通信する Katello エージェント
8000	TCP	HTTPS	キックスター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
9090	TCP	HTTPS	統合 Capsule のスマートプロキシへの SCAP レポートの送信、プロビジョニング中の検出イメージ向け
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続

Satellite Server に直接接続された管理対象ホストは、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2.6 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

ポート	プロトコル	サービス	用途
5647	TCP	amqp	Capsule の Qpid ディスパッチルータと通信する Katello エージェント
8000	TCP	HTTPS	キックスターテンプレート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への SCAP レポートの送信、プロビジョニング中の検出イメージ向け
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続
53	TCP および UDP	DNS	Capsule の DNS サービスに Capsule DNS を問い合わせるクライアント (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)

表2.7 Satellite に通信する Capsule 向けポート

ポート	プロトコル	サービス	用途
80	TCP	HTTP	Anaconda、yum、および Katello 証明書アップデートの取得向け
443	TCP	HTTPS	Katello、Foreman、Foreman API、および Pulp
5646	TCP	amqp	Capsule の Qpid ディスパッチルータから Satellite の Qpid ディスパッチルータへの通信

ポート	プロトコル	サービス	用途
5647	TCP	amqp	Satellite の Qpid ディスパッチルーターと通信する Katello エージェント
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続

Capsule Server が稼働しているベースシステムは、Satellite に統合されている Capsule のクライアントとなります。「[Satellite に通信するクライアント向けポート](#)」の表を参照してください。

表2.8 Capsule に通信する Satellite 向けポート

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

表2.9 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2.10 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	libvirt のコンピュートリソースに対する Satellite による通信
443	TCP	HTTPS	vCenter のコンピュートリソースに対する Satellite による通信
7911	TCP	DHCP	<ul style="list-style-type: none"> DHCP レコードのオーケストレーションのための実行元が Capsule のコマンド (ローカルまたは外部) DHCP が外部サービスにより提供された場合は、外部サーバーでポートを開く必要があります。
5000	TCP	HTTP	OpenStack のコンピュートリソースまたは実行中のコンテナに対する Satellite による通信
22, 16514	TCP	SSH、SSL/TLS	libvirt のコンピュートリソースに対する Satellite による通信
389、636	TCP	LDAP、LDAPS	LDAP およびセキュアな LDAP 認証ソースに対する Satellite による通信
5900～5930	TCP	SSL/TLS	ハイパーバイザー向け Web UI の NoVNC コンソールに対する Satellite による通信



注記

DHCP Capsule は IP アドレスが空であることを確認するために ICMP ECHO を送信し、応答なしが期待されます。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、いかなる応答でも IP アドレスの割り当てが妨げられます。

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

Satellite Server の内部 Capsule のクライアントである Capsule とコンテンツホストは、Satellite のホストベースのファイアウォールとすべてのネットワークベースのファイアウォールを介したアクセスを必要とします。

本セクションでは、Satellite をインストールする Red Hat Enterprise Linux 7 システム上のホストベースのファイアウォールの設定と、クライアントからの受信接続を有効にし、これらの設定をシステムの再起動後にも保持する方法について説明します。使用するポートの詳細は「[ポートとファイアウォールの要件](#)」を参照してください。

ファイアウォールの設定

1. クライアントと **Satellite** 間の通信に必要なポートを開きます。

```
# firewall-cmd \
--add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="9090/tcp"
```

2. **--permanent** オプションを追加してコマンドを再度実行し、設定を永続化します。

```
# firewall-cmd --permanent \
--add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="9090/tcp"
```

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

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

前提条件

Capsule Server のベースシステムは、Satellite Server のクライアントであるため、「[クライアントから Satellite Server への接続の有効化](#)」を最初に完了する必要があります。この手順により、外部の Capsule Server が必要とする追加ポートが開かれます。

使用するポートの詳細は「[ポートとファイアウォールの要件](#)」を参照してください。

1. ファイヤーウォールを設定します。

```
# firewall-cmd --add-port="5000/tcp" --add-port="5646/tcp"
```

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

```
# firewall-cmd --permanent --add-port="5000/tcp" --add-port="5646/tcp"
```

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

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

使用されるポートの詳細は「[ポートとファイアウォールの要件](#)」を参照してください。

1. ファイヤーウォールを設定します。

```
# firewall-cmd --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

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

```
# firewall-cmd --permanent --add-port="53/udp" --add-port="53/tcp" \
--add-port="67/udp" --add-port="69/udp" \
--add-port="80/tcp" --add-port="443/tcp" \
--add-port="5000/tcp" --add-port="5647/tcp" \
--add-port="8000/tcp" --add-port="8140/tcp" \
--add-port="8443/tcp" --add-port="9090/tcp"
```

2.9. ファイアウォール設定の確認

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

ファイアウォール設定の確認

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

詳細は『**Red Hat Enterprise Linux 7 セキュリティーガイド**』の「[firewall-cmd コマンドラインツールを使用したファイアウォールの設定](#)」を参照してください。

2.10. DNS 解決の検証

完全修飾ドメイン名を使用して完全な正引きおよび逆引き DNS 解決を検証すると、**Satellite** のインストール中の問題を回避できます。

ホスト名とローカルホストが正しく解決されることを確認します。

```
# ping -c1 localhost
# ping -c1 `hostname -f` # my_system.domain.com
```

名前解決に成功すると、以下のような出力が表示されます。

```
# ping -c1 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.043 ms

--- localhost ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.043/0.043/0.043/0.000 ms

# ping -c1 `hostname -f`
PING hostname.gateway (XX.XX.XX.XX) 56(84) bytes of data.
64 bytes from hostname.gateway (XX.XX.XX.XX): icmp_seq=1 ttl=64 time=0.019
ms
```

```
--- localhost.gateway ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.019/0.019/0.019/0.000 ms
```

静的および一時的なホスト名との不一致を避けるには、次のコマンドを入力して、システム上のすべてのホスト名を設定します。

```
# hostnamectl set-hostname name
```

詳細は、『**Red Hat Enterprise Linux 7 ネットワークガイド**』の「[hostnamectl を使ったホスト名の設定](#)」を参照してください。



警告

Satellite 6 の運用には名前解決が非常に重要です。**Satellite** が完全修飾ドメイン名を適切に解決できないと、多くのオプションが失敗します。これらのオプションには、コンテンツ管理、サブスクリプション管理、およびプロビジョニングがあります。

2.11. デフォルトの SELINUX ポートの変更

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

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

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

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

作業開始前の準備

- **Satellite** をインストールする前に、SELinux を有効にし、**Permissive** または **Enforcing** モードで実行する必要があります。詳細は「[SELinux ユーザーおよび管理者のガイド](#)」を参照してください。

デフォルトのポートの、ユーザー指定のポートへの変更

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

デフォルトのポート	SELinux コマンド
80、443、8443	<code>semanage port -a -t http_port_t -p tcp 99999</code>
8080	<code>semanage port -a -t http_cache_port_t -p tcp 99999</code>
8140	<code>semanage port -a -t puppet_port_t -p tcp 99999</code>
9090	<code>semanage port -a -t websm_port_t -p tcp 99999</code>
69	<code>semanage port -a -t tftp_port_t -p udp 99999</code>
53 (TCP)	<code>semanage port -a -t dns_port_t -p tcp 99999</code>
53 (UDP)	<code>semanage port -a -t dns_port_t -p udp 99999</code>
67、68	<code>semanage port -a -t dhcpd_port_t -p udp 99999</code>
5671	<code>semanage port -a -t amqp_port_t -p tcp 99999</code>
8000	<code>semanage port -a -t soundd_port_t -p tcp 99999</code>
7911	<code>semanage port -a -t dhcpd_port_t -p tcp 99999</code>
5000 (Red Hat Enterprise Linux 7 の場合)	<code>semanage port -a -t complex_main_port_t -p tcp 99999</code>
22	<code>semanage port -a -t ssh_port_t -p tcp 99999</code>
16514 (libvirt)	<code>semanage port -a -t virt_port_t -p tcp 99999</code>
389、636	<code>semanage port -a -t ldap_port_t -p tcp 99999</code>
5910～5930	<code>semanage port -a -t vnc_port_t -p tcp 99999</code>

2. 以前使用したポート番号とポートタイプの関連付けを解除します。

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


第3章 SATELLITE SERVER のインストール

本章では、Red Hat Satellite Server のインストール、初期設定、マニフェストの作成およびインストール、および追加設定の実行について説明します。

Red Hat Satellite 6.3 はデフォルトで Puppet 3 を使用しますが、インストールスクリプトの実行前に Puppet 4 アップグレードリポジトリを有効にすると、以下のインストール手順の一部でオプションとして Puppet 4 をインストールすることもできます。インストール後に Puppet 4 にアップグレードする手順と Puppet モジュールのアップグレードに関する情報は、『RED HAT SATELLITE のアップグレードおよびアップデート』の「[Puppet のアップグレード](#)」を参照してください。

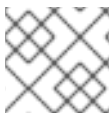
Satellite Server は、以下の 2 つのインストール方法があります。

接続インストール

Satellite Server のインストールに必要なパッケージは、Red Hat Content Delivery Network (CDN) から直接取得できます。CDN を使用すると、システムは常に最新のアップデートを受信できます。

非接続インストール

外部のコンピューターを使用してパッケージの ISO イメージをダウンロードして、それを Satellite Server のインストール先のシステムにコピーする必要があります。非接続環境が必要な場合にのみ、ISO イメージを使用してください。ISO イメージには最新のアップデートが含まれていない場合があります。



注記

Satellite Server をそれ自体に登録することはできません。

3.1. 接続済みネットワークからの SATELLITE SERVER のインストール

接続済みネットワークから Satellite Server をインストールする場合には、Red Hat Content Delivery Network から直接パッケージとアップデートを取得できます。

Satellite 6 インストールスクリプトは Puppet をベースとしていることに留意してください。つまり、インストールスクリプトを複数回実行すると、手動での設定変更が上書きされることがあります。この問題を回避し、適用する変更を特定するには、インストールスクリプトを実行時に `--noop` 引数を使用します。この引数により、実際の変更は行われません。潜在的な変更は `/var/log/katello-installer.log` に書き込まれます。

ファイルは常にバックアップされるため、不要な変更は復元することができます。たとえば、`katello-installer` ログには、Filebucket に関する以下のようなエントリーが示されます。

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

以前のファイルは以下のように復元できます。

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

3.1.1. Red Hat Subscription Management への登録

Red Hat サブスクリプション管理にホストを登録すると、ホストはユーザーが利用可能なサブスクリプションに対するコンテンツをサブスクライブし、使用できます。これには、Red Hat Enterprise Linux、Red Hat Software Collection (RHSC)、Red Hat Satellite などのコンテンツが含まれます。

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

```
# subscription-manager register
```

このコマンドを実行すると、以下のような出力が表示されます。

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

3.1.2. Satellite サブスクリプションを識別してホストへの割り当て

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

1. Satellite サブスクリプションの特定

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

このコマンドは、利用可能なすべてのサブスクリプションのフィールドに対して小文字と大文字を区別しない検索を実行します (**Subscription Name** と **Provides** を含み、**Red Hat Satellite** のすべてのインスタンスに一致)。システムにすでに割り当てられていないサブスクリプションが、利用可能として分類されます。検索文字列には、1 つ文字、またはゼロ個以上の文字にそれぞれ一致するワイルドカード `?` または `*` を含めることもできます。ワイルドカード文字はバックスラッシュでエスケープして、リテラルの疑問符またはアスタリスクを表すことができます。

利用可能な Satellite サブスクリプションを見つけることができない場合は、Red Hat ナレッジベースソリューション「[How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager?](#)」を参照して、スクリプトを実行し、サブスクリプションが別のシステムで使用されているかどうかを確認できます。

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

- a. 実行する **subscription-manager** コマンドの形式に関係なく、出力は以下のようになります。

```
Subscription Name: Red Hat Satellite
Provides:          Red Hat Satellite 6
                  Red Hat Enterprise Linux Server
                  Red Hat Satellite
                  Red Hat Enterprise Linux Load Balancer (for
RHEL Server)
```

```

SKU:                MCT0370
Pool ID:            8a85f9874152663c0541943739717d11
Available:         3
Suggested:         1
Service Level:     Premium
Service Type:      L1-L3
Multi-Entitlement: No
Ends:              10/07/2014
System Type:       Physical

```

2. 後で **Satellite** ホストに割り当てるために、プール ID をメモします。実際に使用するプール ID は、この例で使用されているものとは異なります。
3. **Satellite** サーバーにサブスクリプションを割り当てるには、実際のプール ID を指定して以下のコマンドを実行します。

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

出力は以下のようになります。

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

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

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

この出力では、以下のような内容が表示されます。

```

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

```

Status Details:

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

3.1.3. リポジトリの設定

1. すべての既存のリポジトリを無効にします。

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

```

2. 必要なりポジトリを有効にします。

- Red Hat Satellite、Red Hat Enterprise Linux、Red Hat Software Collections、および Puppet 4 のリポジトリを有効にするには、以下のコマンドを使用します。

```
# subscription-manager repos \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-satellite-6.3-rpms \
--enable=rhel-7-server-satellite-6.3-puppet4-rpms

```

- もしくは、以下のコマンドを使用して、Red Hat Satellite、Red Hat Enterprise Linux、Red Hat Software Collections、および Puppet 3 のリポジトリを有効にします。

```
# subscription-manager repos \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-satellite-6.3-rpms

```

**注記**

Satellite 6.3 は、Puppet 3 でサポートされる最後のリリースです。Puppet 3 から Puppet 4 へのアップグレードは、Satellite 6.3 以前を Satellite 6.4 にアップグレードする前に行う必要があります。Satellite 6.4 は Puppet 5 だけをサポートしますが、Puppet 5 へのアップグレードは、Satellite のアップグレード時に行われます。

**注記**

Red Hat Satellite を Red Hat Virtualization (RHV) でホストされる仮想マシンとしてインストールする場合は、**Red Hat Common** リポジトリを有効にして、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. Red Hat Subscription Manager が特定のオペレーティングシステムリリースを使用しないようにします。

```
# subscription-manager release --unset

```

4. Red Hat 以外の yum リポジトリからのすべてのメタデータを消去します。

```
# yum clean all
```

5. リポジトリが有効になっていることを確認します。

```
# yum repolist enabled
```

以下のような出力が表示されます。

```
Loaded plugins: product-id, subscription-manager
repo id                                repo name
status
!rhel-7-server-rpms/x86_64             Red Hat Enterprise Linux
7 Server (RPMs)                        9,889
!rhel-7-server-satellite-6.3-rpms/x86_64 Red Hat Satellite 6.3
(for RHEL 7 Server) (RPMs)            545
!rhel-server-rhsc1-7-rpms/x86_64       Red Hat Software
Collections RPMs for Red Hat Enterprise Linux 7 Server 4,279
repolist: 14,713
```

3.1.4. Satellite サーバーパッケージのインストール

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

1. すべてのパッケージを更新します。

```
# yum update
```

2. インストールパッケージをインストールします。

```
# yum install satellite
```

3. 「[初期設定の実行](#)」に移動して、インストーラスクリプトを実行し、Satellite Server の初期設定を行います。

3.2. 切断されたネットワークからのダウンロードおよびインストール

Red Hat Satellite Server のホストがオフライン環境にある場合は、ISO イメージを使用して Satellite Server をインストールできます。ISO イメージには最新のアップデート、バグフィックス、および機能が含まれないことがあるため、この方法はこの環境以外では推奨されません。



注記

ベースシステムが Red Hat CDN から更新されなかった場合、パッケージの依存関係エラーが発生することがあります。必要なパッケージの最新バージョンは手動でダウンロードしてインストールしてください。詳細は「[パッケージの手動ダウンロード](#)」を参照してください。

作業開始前の準備

- インストールで使用されたりポジトリのコピーは **/opt/** ディレクトリーに格納されます。このファイルシステムとディレクトリーのために最低 **3GB** の領域を確保してください。

3.2.1. バイナリー DVD イメージのダウンロード

1. [Red Hat カスタマーポータル](#) に移動し、ログインします。
2. **ダウンロード** をクリックします。
3. **Red Hat Enterprise Linux** を選択します。
4. 製品とバージョンがご使用の環境に適切であることを確認します。
 - **Product Variant (製品のバリエーション)** は **Red Hat Enterprise Linux Server** に設定されます。
 - **Version (バージョン)** は、ベースシステムとして使用する予定の製品の最新マイナーバージョンに設定されます。
 - **Architecture (アーキテクチャー)** は **64 ビットバージョン** に設定されます。
5. **Product Software (製品ソフトウェア)** タブで、最新の Red Hat Enterprise Linux Server バージョン向けのバイナリー DVD イメージをダウンロードします。
6. **DOWNLOADS (ダウンロード)** をクリックし、**Red Hat Satellite** を選択します。
7. 製品とバージョンがご使用の環境に適切であることを確認します。
 - **Product Variant (製品のバリエーション)** は **Red Hat Satellite** に設定されます。
 - **Version (バージョン)** は、ベースシステムとして使用する予定の製品の最新マイナーバージョンに設定されます。
 - **Architecture (アーキテクチャー)** は **64 ビットバージョン** に設定されます。
8. **Product Software (製品ソフトウェア)** タブで、最新の Red Hat Satellite バージョン向けのバイナリー DVD イメージをダウンロードします。
9. ISO ファイルを Satellite ベースシステムまたは他のアクセス可能なストレージデバイスにコピーします。

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

3.2.2. オフラインリポジトリでベースシステムの設定

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

```
# mkdir /media/rhel7-server
```

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

```
# mount -o loop rhel7-Server-DVD.iso /media/rhel7-server
```

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

-

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

3. ISO ファイルのリポジトリデータファイルをコピーします。

```
# cp /media/rhel7-server/media.repo /etc/yum.repos.d/rhel7-
server.repo
```

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

```
baseurl=file:///media/rhel7-server/
```

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

```
# vi /etc/yum.repos.d/rhel7-server.repo
[InstallMedia]
name=Red Hat Enterprise Linux 7.2
mediaid=1446216863.790260
metadata_expire=-1
gpgcheck=0
cost=500
baseurl=file:///media/rhel7-server/
enabled=1
```

5. リポジトリが設定されたことを確認します。

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

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

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

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

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

以下の例では、Red Hat Enterprise Linux 7 向け Red Hat Satellite 6.3.0 を使用した ISO のマウントを示しています。

```
# mount -o loop satellite-6.3.0-rhel-7-x86_64-dvd.iso /media/sat6
mount: /dev/loop1 is write-protected, mounting read-only
```

8. Red Hat Satellite 6.3 はデフォルトで Puppet 3 を使用しますが、Puppet 4 を使用することも可能で、その場合は必要なパッケージにアクセスするためのローカルリポジトリを作成し、そこに以下のコンテンツを追加します。

```
# vi /etc/yum.repos.d/satellite-puppet4.repo
[satellite-puppet4]
name=satellite-puppet4
baseurl=file:///media/sat6/addons/Puppet4
enabled=1
gpgcheck=1
```

3.2.3. オフラインリポジトリからのインストール

1. Red Hat Enterprise Linux Server と Red Hat Satellite の ISO イメージがマウントされていることを確認します。

```
# findmnt -t iso9660
```

2. Red Hat GPG キーをインポートします。

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

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

```
# yum update
```

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

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

5. マウントされたディレクトリでインストールスクリプトを実行します。

```
# ./install_packages
This script will install the foreman packages on the current
machine.
- Ensuring we are in an expected directory.
- Copying installation files.
- Creating a Repository File
- Creating RHSCS Repository File
- Checking to see if Foreman is already installed.
- Importing the gpg key.
- Foreman is not yet installed, installing it.
- Installation repository will remain configured for future
package installs.
- Installation media can now be safely unmounted.

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

パッケージが見つからない、または古いためにスクリプトが失敗する場合は、これらをダウンロードして個別にインストールする必要があります。手順は「[パッケージの手動ダウンロード](#)」を参照してください。

インストール済みパッケージが必要なものよりも新しいためにスクリプトが失敗する場合は、**yum distribution-synchronization** を実行してインストール済みパッケージを Red Hat Enterprise Linux ISO からのバージョンにダウングレードしてから、インストールスクリプトを再度実行します。リポジトリのソースが Red Hat Enterprise Linux ISO 以外のものに設定されている場合にのみ、これが発生します。このようなりポジトリの使用はサポート対象外になります。

3.2.4. パッケージの手動ダウンロード

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

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

3.3. 初期設定の実行

本セクションでは、Red Hat Satellite Server インストール時のホストオペレーティングシステムの初期設定について説明します。時間の同期、**sos** パッケージのインストール、インストールオプションの指定などが含まれます。

作業を進める前に、使用している環境に適切なマニフェストまたはパッケージを確認します。マニフェストについての詳細は『[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。

3.3.1. 時間の同期

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

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

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

NTP を使用した時間の同期

1. ntp をインストールします。

```
# yum install ntp
```

2. NTP サーバーが利用可能であることを確認します。

```
# ntpdate -q ntp_server_address
```

3. システム時刻を設定します。

```
# ntpdate ntp_server_address
```

chronyd を使用した時間の同期

1. chronyd をインストールします。

```
# yum install chrony
```

2. chronyd サービスを起動して、有効にします。

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

3.3.2. ホストオペレーティングシステムへの SOS パッケージのインストール

ホストオペレーティングシステムには **sos** パッケージをインストールする必要があります。**sos** パッケージを使用すると、Red Hat Enterprise Linux システムから設定と診断情報を収集できます。また、Red Hat テクニカルサポートでサービスリクエストを開く際に必要な初期システム分析を提供することもできます。**sos** の使用の詳細は、カスタマーポータルナレッジベース「[Red Hat Enterprise Linux 4.6 以降における sosreport の役割と取得方法](#)」を参照してください。

sos パッケージをインストールします。

```
# yum install sos
```

3.3.3. インストールオプションの指定

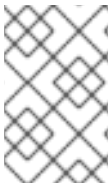
Satellite Server は **satellite-installer** インストールスクリプトを使用してインストールし、初期設定の一部として自動または手動で **Satellite** を設定します。

以下のいずれかの設定方法を選択します。

- 自動設定: この方法は、インストールスクリプトの実行時に応答ファイルを使用して設定プロセスを自動化することで実行します。応答ファイルとは、コマンドやスクリプトによって読み込まれるパラメーター一覧が含まれているファイルです。デフォルトの **Satellite** 応答ファイルは、**/etc/foreman-installer/scenarios.d/satellite-answers.yaml** です。使用する応答ファイルは、**/etc/foreman-installer/scenarios.d/satellite.yaml** 設定ファイル内の **answer_file** ディレクティブで設定します。
応答ファイルを使用したインストールスクリプトによる初期設定の実行法は「[応答ファイルを使用した初期設定の自動実行](#)」を参照してください。
- 手動設定: 1 つ以上のコマンドオプションが含まれるインストールスクリプトを実行します。コ

マンドオプションは、対応するデフォルトの初期設定オプションを上書きし、**Satellite** 応答ファイルに記録されます。必要なオプションを設定するために、スクリプトは何回でも実行することができます。

コマンドラインオプションのあるインストールスクリプトによる初期設定の実行法は「[手動による初期設定](#)」を参照してください。



注記

Satellite インストーラーの実行時に使用するオプションによっては、設定が完了するのに数分かかることがあります。管理者は、応答ファイルを見ることで、両方の方法でこれまでに使用されたオプションを確認できます。

3.3.3.1. 手動による初期設定

初期設定では、組織、場所、ユーザー名、およびパスワードが作成されます。初期設定後に、必要に応じて追加の組織と場所を作成できます。

インストールプロセスの完了には、数十分かかることがあります。システムにリモートで接続する場合は、リモートシステムから切断された場合にインストールの進捗を確認できるよう、通信セッションの一時中断または再接続を許可できる **screen** などのユーティリティーの使用を検討してください。[Red Hat ナレッジベースの記事「How to use the screen command」](#)には **screen** のインストールについて記載されています。詳細は **screen** の man ページを参照してください。インストールコマンドを実行しているシェルへの接続が切断された場合は、`/var/log/foreman-installer/satellite.log` のログを参照してプロセスが正常に完了したかどうかを確認します。

Satellite Server の手動設定

satellite-installer --scenario satellite --help コマンドを使用して利用可能なオプションとすべてのデフォルト値を表示します。値を指定しない場合は、デフォルト値が使用されます。

--foreman-initial-organization オプションには、意味のある値を指定することが推奨されます。たとえば会社名を指定できます。値に一致する内部ラベルが作成されますが、このラベルは後で変更できません。値を指定しない場合は、ラベルが **Default_Organization** の **Default Organization** という名前の組織が作成されます。組織名は変更できますが、ラベルは変更できません。

デフォルトでは、インストーラーが設定するすべての設定ファイルが **Puppet** によって管理されます。**satellite-installer** を実行すると、**Puppet** が管理するファイルに手動で加えられた変更が初期値で上書きされます。**Satellite Server** は、デフォルトでは、サービスとして実行している **Puppet** エージェントを使用してインストールされます。必要に応じて、**--puppet-runmode=none** オプションを使用して、**Satellite Server** で **Puppet** エージェントを無効にできます。

DNS ファイルと DHCP ファイルを手動で管理する場合には、**--foreman-proxy-dns-managed=false** オプションと **--foreman-proxy-dhcp-managed=false** オプションを使用して、**Puppet** が各サービスに関連するファイルを管理しないようにします。他のサービスにカスタム設定を適用する方法は「[付録C Red Hat Satellite へのカスタム設定の適用](#)」を参照してください。

```
# satellite-installer --scenario satellite \
--foreman-initial-organization "initial_organization_name" \
--foreman-initial-location "initial_location_name" \
--foreman-admin-username admin_user_name \
--foreman-admin-password admin_password \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

スクリプトが正常に完了すると、以下の出力が表示されます。

```

Installing                                Done
[100%] [.....]
Success!
* Satellite is running at https://rhel7-4-sat6-3.example.com
  Initial credentials are admin / changeme
* To install an additional Capsule on separate machine continue by
running:

    capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE" --certs-tar
"/root/$CAPSULE-certs.tar"
The full log is at /var/log/foreman-installer/satellite.log

```

切断環境でインストールしている場合は、ISO イメージをアンマウントします。

```

# umount /media/sat6
# umount /media/rhel7-server

```

3.3.3.2. 応答ファイルを使用した初期設定の自動実行

応答ファイルを使用すると、カスタマイズされたオプションでインストールを自動化できます。最初の応答ファイルには、部分的に情報が入力されます。応答ファイルには、**satellite-installer** の初回実行後に、インストール用の標準パラメーター値が入力されます。「[手動による初期設定](#)」の記載通りに **Satellite Server** を既にインストールしている場合は、この方法を使用する必要ありません。ただし、この方法を使用していつでも **Satellite Server** の設定に変更を加えることはできます。

ネットワークの変更の場合は、可能な限り、IP アドレスの代わりに **FQDN** を使用する必要があります。

応答ファイルを使用した **Satellite Server** の自動設定

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

```

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

```

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

```

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

```

5. **satellite-installer** スクリプトを実行します。

```

# satellite-installer --scenario satellite

```

6. 切断環境でインストールしている場合は、ISO イメージをアンマウントします。

```
# umount /media/sat6  
# umount /media/rhel7-server
```

3.3.4. カスタマーポータルでサブスクリプション割り当ての作成

サブスクリプション情報は、Red Hat カスタマーポータルでアクセスできます。また、カスタマーポータルでは、**subscription allocation** を使用して、Red Hat Satellite サーバーなどのオンプレミス管理アプリケーションで使用するサブスクリプションを割り当てることができます。

1. ブラウザーで <https://access.redhat.com/> を開き、Red Hat アカウントでログインします。
2. カスタマーポータルの左上にある **サブスクリプション** に移動します。
3. **サブスクリプション割り当て** に移動します。
4. **新規サブスクリプションの割り当てを作成** をクリックします。
5. **名前** フィールドに名前を入力します。
6. **タイプ** の一覧からお使いの **Satellite Server** に一致するタイプとバージョンを選択します。
7. **作成** をクリックします。

3.3.5. 割り当てへのサブスクリプションの追加

以下の手順では、サブスクリプションを割り当てに追加する方法を説明します。

1. **サブスクリプション割り当て** に移動します。
2. 変更するサブスクリプションの名前を選択します。
3. **サブスクリプションタブ** をクリックします。
4. **サブスクリプションの追加** をクリックします。
5. Red Hat 製品サブスクリプションの一覧が表示されます。各製品に対する**エンタイトルメントの数量**を入力します。
6. **送信** をクリックして割り当てを完了します。

割り当てにサブスクリプションを追加したら、マニフェストファイルをエクスポートします。

3.3.6. カスタマーポータルからのサブスクリプションマニフェストのエクスポート

少なくとも1つのサブスクリプションがあるサブスクリプション割り当てを表示する間に、以下のいずれかでマニフェストをエクスポートできます。

- **サブスクリプションセクションの詳細** タブから **マニフェストのエクスポート** ボタンをクリックします。
- **サブスクリプションタブ** から **マニフェストのエクスポート** ボタンをクリックします。

マニフェストをエクスポートすると、カスタマーポータルにより、選択したサブスクリプション証明書がエンコードされ、.zip アーカイブが作成されます。作成した .zip アーカイブはサブスクリプションのマニフェストで、Satellite サーバーにアップロードできます。

3.3.6.1. Satellite Server へのサブスクリプションマニフェストのインポート

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

Web UI を使用する場合

1. コンテキストが、使用する組織に設定されていることを確認します。
2. コンテンツ > **Red Hat サブスクリプション** に移動します。
3. マニフェストの管理 をクリックして、組織のマニフェストページを表示します。
4. ファイルの選択 をクリックしてサブスクリプションマニフェストを選択し、アップロード をクリックします。

CLI を使用する場合

Red Hat Satellite 6 CLI を使用するには、マニフェストが **Satellite Server** 上にある必要があります。ローカルクライアントシステムで、マニフェストを **Satellite Server** にコピーします。

```
[user@client ~]$ scp ~/manifest_file.zip root@satellite.example.com:~/.
```

次に、以下のコマンドを使用してインポートします。

```
[root@satellite ~]# hammer subscription upload \
--file ~/manifest_file.zip \
--organization "organization_name"
```

数分後に、CLI により、正常なマニフェストのインポートが報告されます。

上記の手順を完了すると、リポジトリを有効にして **Red Hat** コンテンツをインポートできるようになります。これは、後に続くいくつかの手順での前提条件になります。詳細は、『**Red Hat Satellite コンテンツ管理ガイド**』の「[Red Hat コンテンツのインポート](#)」を参照してください。

3.4. 追加設定の実行

3.4.1. Satellite Tools リポジトリのインストール

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

Satellite Tools リポジトリのインストール手順:

1. Satellite Web UI で、コンテンツ > **Red Hat リポジトリ** に移動し、**RPM** タブを選択します。
2. Red Hat Enterprise Linux Server 項目を見つけ、展開します。
3. Red Hat Satellite Tools 6.3 (Red Hat Enterprise Linux 7 Server 用) (RPM) 項目を見つけ、展開します。
Red Hat Satellite Tools 6.3 項目が非表示の場合は、その項目がカスタマーポータルから取得したサブスクリプションマニフェストに含まれないことが原因である場合があります。この問題を修正するには、カスタマーポータルにログインし、これらのリポジトリを追加し、サブス

クリプションマニフェストをダウンロードして、**Satellite** にインポートします。

4. **Satellite 6.3 Tools** リポジトリの名前の横にある **Enabled** チェックボックスをオンにします。

ホストで実行している **Red Hat Enterprise Linux** の各サポート対象メジャーバージョンに対して **Satellite Tools** リポジトリを有効にします。**Red Hat** リポジトリの有効後に、このリポジトリの製品が自動的に作成されます。

Satellite Tools リポジトリの同期方法:

1. **Content (コンテンツ) > Sync Status (同期ステータス)** に移動します。
同期可能な製品リポジトリのリストが表示されます。
2. 製品コンテンツの横にある矢印をクリックして利用可能なコンテンツを表示します。
3. 同期するコンテンツを選択します。
4. **Synchronize Now (今すぐ同期)** をクリックします。

3.4.2. HTTP プロキシを使用した **Satellite Server** の設定

ネットワークで HTTP プロキシを使用している場合は、それを使用するように **Satellite Server** を設定できます。ネットワークの変更が原因で接続が失われるのを回避するために、可能な限り IP の代わりに FQDN を使用します。

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

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

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

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

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

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

ホスト名	ポート	プロトコル
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS

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

Satellite Server は、SSL で安全に Red Hat CDN と通信します。SSL インターセプトプロキシを使用すると、この通信が妨害されます。これらのホストは、プロキシでホワイトリスト化されている必要があります。

Red Hat CDN (cdn.redhat.com) で使用されている IP アドレスの一覧は、Red Hat カスタマーポータルナレッジベース記事「[Red Hat が公開している CIDR の一覧](#)」を参照してください。

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

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

# port for http proxy server
proxy_port = 8080

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

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

4. SELinux を使用すると、Red Hat Satellite 6 と Red Hat Subscription Manager のアクセスが、特定ポートに限定されます。HTTP キャッシュの TCP ポートは、8080、8118、8123、および 10001 ~ 10010 になります。SELinux のタイプ `http_cache_port_t` がないポートを使用する場合は、以下のステップを行います。

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

```
# semanage port -l | grep http_cache
http_cache_port_t      tcp      8080, 8118, 8123, 10001-10010
[出力を省略]
```

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

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

SELinux ポートの設定に関する詳細は「[デフォルトの SELinux ポートの変更](#)」を参照してください。

3.4.3. 管理対象ホスト上での電源管理の有効化

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

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

BMC サービスの詳細は『ホストの管理』の「[追加のネットワークインターフェースの設定](#)」を参照してください。

作業開始前の準備

- すべての管理対象ホストに **BMC** タイプのネットワークインターフェースが搭載されている必要があります。Satellite はこの NIC を使用して適切な認証情報をホストに渡します。

管理対象ホスト上での電源管理の有効化

1. オプションを使用してインストーラーを実行し、BMC を有効にします。

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

3.4.4. Satellite Server で DNS、DHCP、および TFTP の設定

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

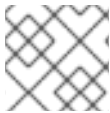
外部サービスを設定する場合は、「[5章 外部サービスの設定](#)」を参照してください。

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

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

作業開始前の準備

- ネットワーク管理者に連絡して正しい設定が行われていることを確認します。
- 以下の情報を用意する必要があります。
 - DHCP IP アドレス範囲
 - DHCP ゲートウェイ IP アドレス
 - DHCP ネームサーバー IP アドレス
 - DNS 情報
 - TFTP サーバー名
- ネットワークの変更の場合は、可能な限り、IP アドレスの代わりに FQDN を使用します。



注記

タスクの情報は例です。ご使用の環境情報を使用してください。

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

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

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

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

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

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

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



注記

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

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

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

1. Web UI で、インフラストラクチャー > サブネット に移動し、サブネットを選択します。

2. **Capsules (カプセル)** タブで、ドロップダウンリストを **None (なし)** に設定して、関連付けられている DHCP Capsule または TFTP Capsule がないことを確認します。
3. 正引きレコードオーケストレーションを無効にします。
 - a. インフラストラクチャー > **ドメイン** に移動し、ドメインを選択します。
 - b. **Domain (ドメイン)** タブで、**DNS Capsule (DNS カプセル)** ドロップダウンリストを **None (なし)** に設定します。
4. 逆引き (PTR) レコードオーケストレーションを無効にします。
 - a. インフラストラクチャー > **Subnets (サブネット)** に移動し、サブネットを選択します。
 - b. **Capsules (カプセル)** タブで、**Reverse DNS Capsule (逆引き DNS カプセル)** ドロップダウンリストを **None (なし)** に設定します。
5. オプション: サードパーティーが提供する DHCP サービスを使用する場合は、以下のオプションを渡すように DHCP サーバーを設定します。

```
Option 66: IP_address_of_Satellite_or_Capsule
Option 67: /pxelinux.0
```

DHCP オプションの詳細は「[RFC 2132](#)」を参照してください。



注記

Satellite 6 は、Capsule が該当するサブネットとドメインに設定されていない場合にオーケストレーションを実行しません。Capsule の関連付けを有効または無効にした場合に、期待されるレコードと設定ファイルが存在しないと、既存のホストのオーケストレーションコマンドが失敗することがあります。オーケストレーションを有効にするために Capsule を関連付ける場合は、将来ホストの削除に失敗することを回避するために、既存の Satellite 6 管理対象ホストに対して必要な DHCP レコード、DNS レコード、TFTP ファイルが所定の場所にあることを確認します。

3.4.6. Satellite Server で送信メールの設定

Satellite Server からメールメッセージを送信するには、SMTP サーバーまたは **sendmail** コマンドのいずれかを使用できます。

前提条件

前回のリリースからアップグレードしている場合は、設定ファイル **/usr/share/foreman/config/email.yaml** の名前を変更するか削除して、**httpd** サービスを再起動してください。例を示します。

```
# mv /usr/share/foreman/config/email.yaml \
/usr/share/foreman/config/email.yaml-backup
# systemctl restart httpd
```

Satellite Server で送信メールの設定

1. Satellite web UI で、**管理** → **設定** に移動します。
2. **Email** タブをクリックして、希望する配信方法に一致する設定オプションを設定します。変更は即座に反映されます。

- a. 以下の例は、SMTP サーバーを使用する場合の設定オプションの例を示しています。

表3.1 配信方法に SMTP サーバーを使用する例

名前	値の例
配信方法	SMTP
SMTP アドレス	smtp.example.com
SMTP 認証	ログイン
SMTP HELO/EHLO ドメイン	example.com
SMTP パスワード	パスワード
SMTP ポート	25
SMTP ユーザー名	satellite@example.com

SMTP ユーザー名 と SMTP パスワード では、SMTP サーバーのログイン認証情報を指定します。

- b. 以下の例では、gmail.com が SMTP サーバーとして使用されています。

表3.2 gmail.com を SMTP サーバーとして使用する例

名前	値の例
配信方法	SMTP
SMTP アドレス	smtp.gmail.com
SMTP 認証	plain
SMTP HELO/EHLO ドメイン	smtp.gmail.com
SMTP enable StartTLS auto	あり
SMTP パスワード	パスワード
SMTP ポート	587
SMTP ユーザー名	user@gmail.com

- c. 以下の例では、sendmail コマンドが配信方法として使用されています。

表3.3 配信方法に sendmail を使用する例

名前	値の例
配信方法	Sendmail
Sendmail の引数	-i -t -G

Sendmail の引数では、**sendmail** コマンドに渡すオプションを指定します。デフォルト値は、**-i -t**です。詳細は、**sendmail 1** の **man** ページを参照してください。

3. TLS 認証を使用する SMTP サーバーで電子メールを送信する場合は、以下のいずれかの手順を実行してください。

- SMTP サーバーの CA 証明書を信頼済みとしてマークします。このようにマークするには、**Satellite Server** で以下のコマンドを実行します。

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

ここで、**mailca.crt** は SMTP サーバーの CA 証明書です。

- 別の方法では、web UI の **SMTP enable StartTLS auto** オプションを **No** に設定します。
4. **Test email** をクリックしてユーザーのメールアドレスにテストメッセージを送信し、設定が機能していることを確認します。メッセージの送信に失敗する場合は、web UI でエラーが表示されます。詳細については、**/var/log/foreman/production.log** のログを確認してください。



注記

個々のユーザーまたはユーザーグループに対する電子メール通知の設定は、『**Red Hat Satellite の管理**』の「[電子メール通知の設定](#)」を参照してください。

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

SSL 証明書は、情報を保護し、通信を安全にするために使用されます。**Red Hat Satellite 6** は自己署名 SSL 証明書を作成し、**Satellite Server**、外部の **Capsule Server**、およびすべてのホスト間で暗号化された通信を有効します。必要に応じて、デフォルト証明書をカスタム証明書に置き換えることができます。これらの自己署名証明書を使用する代わりに、外部の信頼できる企業である認証局が発行したカスタム SSL 証明書をインストールすることもできます。たとえば、会社のセキュリティポリシーで、認証局から SSL 証明書を取得することが規定されている場合があります。証明書を取得するには、「[Satellite Server 向けの SSL 証明書を取得](#)」にあるように **Certificate Signing Request** を作成して認証局に送信します。すると、署名済み SSL 証明書が送られてきます。



注記

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

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

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

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

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



重要

SSL 証明書には、PEM エンコードのみを使用してください。



注記

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

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

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

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



注記

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

```
# openssl genrsa -out /root/sat_cert/satellite_cert_key.pem 4096
```

3. Certificate Signing Request (CSR) の作成

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



注記

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

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

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

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

Certificate Signing Request セッションの例

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

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

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

4. 証明書要求を認証局に送信します。
要求を送信する場合は、証明書のライフスパンを指定する必要があります。証明書要求を送信する方法は異なるため、推奨される方法について認証局にお問い合わせください。要求に対する応答で、認証局バンドルと署名済み証明書を別々のファイルで受け取ることになります。

3.4.7.2. Satellite Server の SSL 証明書の検証

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

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

```
# katello-certs-check \
  -c /root/sat_cert/satellite_cert.pem \ ❶
  -k /root/sat_cert/satellite_cert_key.pem \ ❷
  -r /root/sat_cert/satellite_cert_csr.pem \ ❸
  -b /root/sat_cert/ca_cert_bundle.pem ❹
```


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

katello-certs-check の出力例

```
Checking expiration of certificate: [OK]
Checking expiration of CA bundle: [OK]
Validating the certificate subject=
/C=AU/ST=Queensland/L=Brisbane/O=Example/OU=Sales/CN=satellite.example.com
/emailAddress=example@example.com
Checking to see if the private key matches the certificate: [OK]
Checking ca bundle against the cert file: [OK]
Checking for non ascii characters[OK]
```

Validation succeeded.

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

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

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

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

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

```
capsule-certs-generate --foreman-proxy-fqdn "$CAPSULE"\
  --certs-tar "~/ $CAPSULE-certs.tar"\
  --server-cert "/root/sat_cert/satellite_cert.pem"\
  --server-cert-req "/root/sat_cert/satellite_cert_csr.pem"\
  --server-key "/root/sat_cert/satellite_cert_key.pem"\
  --server-ca-cert "/root/sat_cert/ca_cert_bundle.pem"
```

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

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

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

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

このセクションのコマンドは、「[Satellite Server の SSL 証明書の検証](#)」で説明されたように **katello-certs-check** コマンドの出力を使用します。**katello-certs-check** の出力は、ターミナルにコピーアンドペーストできます。

1. インストールの状況に応じて、**satellite-installer** コマンドを実行します。
 - a. Satellite がすでにインストールされている場合は、Satellite サーバーで以下のコマンドを実行します。

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

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



注記

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

- b. Satellite を まだインストールしていない場合は、Satellite Server で以下のコマンドを実行します。

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



注記

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

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

3.4.7.4. Satellite Server に接続されたすべてのホストへの新しい証明書のインストール

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

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

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

2. ホストにカスタム SSL 証明書をインストールします。

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

3.4.8. mongod へのアクセスの制限

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

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

1. ファイヤーウォールを設定します。

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

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

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

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

第4章 CAPSULE SERVER のインストール

Capsule Server をインストールする前に、ご使用の環境がインストールの要件を満たしていることを確認する必要があります。Capsule Server のインストール要件は Satellite Server と同じですが、Red Hat CDN への接続にはプロキシを使用しない設定になっている必要があります。詳細は「[システム要件](#)」を参照してください。

4.1. SATELLITE SERVER への CAPSULE SERVER の登録

作業開始前の準備

- Satellite Server には、サブスクライブする組織の適切なリポジトリを使用してマニフェストがインストールされている必要があります。マニフェストには Capsule のベースシステムと Capsule に接続されたすべてのクライアント向けのリポジトリが含まれる必要があります。リポジトリは同期する必要があります。マニフェストとリポジトリの詳細は、『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。
- Satellite Server のベースシステムは、Capsule Server のベースシステムのホスト名を解決する必要があります。Capsule Server のベースシステムは Satellite Server のベースシステムのホスト名を解決する必要があります。
- Red Hat Satellite へのアクセスを妨げるプロキシの使用に関連した変更を元に戻す必要があります。
- ホストとネットワークベースのファイアウォールが設定済みである必要があります。詳細は「[ポートとファイアウォールの要件](#)」を参照してください。
- Satellite Server のユーザー名とパスワードが必要です。詳細は『Red Hat Satellite の管理』の「[外部認証の設定](#)」を参照してください。

Satellite Server への Capsule Server の登録

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

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

2. Capsule Server を組織に登録します。

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

4.2. CAPSULE SERVER サブスクリプションの識別と割り当て

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

1. Capsule Server のサブスクリプションを識別します。

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

このコマンドを実行すると、以下のような出力が表示されます。

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

Subscription Name: Red Hat Satellite Capsule Server
Provides:          Red Hat Satellite Proxy
                  Red Hat Satellite Capsule
                  Red Hat Software Collections (for RHEL Server)
                  Red Hat Satellite Capsule
                  Red Hat Enterprise Linux Server
                  Red Hat Enterprise Linux High Availability (for
RHEL Server)

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

2. 後で **Satellite** ホストに割り当てるために、プール ID をメモします。実際に使用するプール ID は、この例で使用されているものとは異なります。
3. プール ID を使用してサブスクリプションを **Capsule Server** に割り当てます。

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

この出力では、以下のような内容が表示されます。

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

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

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

4.3. リポジトリの設定

1. すべての既存のリポジトリを無効にします。

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

```

2. **Red Hat Satellite Capsule**、**Red Hat Enterprise Linux**、および **Red Hat Software Collections** のリポジトリを有効にします。

Red Hat Software Collections リポジトリは、リモート実行機能を含む一部の **Red Hat Satellite Capsule** 機能で必要な、新しいバージョンの **Ruby** を提供します。

```
# subscription-manager repos --enable rhel-7-server-rpms \
--enable rhel-7-server-satellite-capsule-6.3-rpms \
--enable rhel-server-rhsc1-7-rpms
```

3. お使いの **Puppet** モジュールが **Puppet 4** にアップグレードされている場合は、インストールプロセスの一部としてデフォルトの **Puppet 3** を **Puppet 4** に変更できます。 **Puppet 4** リポジトリを有効にするには、以下を実行します。

```
# subscription-manager repos \
--enable=rhel-7-server-satellite-capsule-6.3-puppet4-rpms
```

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

```
# yum clean all
```

5. リポジトリが有効になっていることを確認します。

```
# yum repolist enabled
```

以下のような出力が表示されます。

```
Loaded plugins: langpacks, product-id, subscription-manager
repo id                                     repo name
status
!rhel-7-server-rpms/7Server/x86_64        Red Hat
Enterprise Linux 7 Server (RPMs)          7,617
!rhel-7-server-satellite-capsule-6.3-rpms/x86_64 Red Hat Satellite
Capsule 6.3(for RHEL 7 Server) (RPMs)     176
repolist: 7,793
```

4.4. 時間の同期

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

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

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

NTP を使用した時間の同期

1. **ntp** をインストールします。

```
# yum install ntp
```

2. NTP サーバーが利用可能であることを確認します。

```
# ntpdate -q ntp_server_address
```

3. システム時刻を設定します。

```
# ntpdate ntp_server_address
```

chronyd を使用した時間の同期

1. chronyd をインストールします。

```
# yum install chrony
```

2. chronyd サービスを起動して、有効にします。

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

4.5. CAPSULE SERVER のインストール

1. インストールパッケージをインストールします。

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

4.6. CAPSULE SERVER の初期設定の実行

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

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

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

作業開始前の準備

- ホストとネットワークベースのファイアウォールが設定済みである必要があります。詳細は「[ポートとファイアウォールの要件](#)」を参照してください。
- **katello-ca-consumer-latest** パッケージがインストール済みである必要があります。詳細は、「[Satellite Server への Capsule Server の登録](#)」を参照してください。
- Capsule Server が Satellite Server に登録されている必要があります。
- 必要なサブスクリプションが Capsule Server に割り当てられている必要があります。

デフォルトのサーバー証明書を使用した Capsule Server の設定

1. Satellite Server で証明書アーカイブを作成します。

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

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

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



注記

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

4.7. CAPSULE SERVER での追加設定の実行

4.7.1. katello エージェントのインストール

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

作業開始前の準備

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

katello-agent のインストール手順:

1. システムにログインします。
2. このバージョンの Satellite 向け Satellite Tools リポジトリを有効にします。


```
# subscription-manager repos \
--enable=rhel-7-server-satellite-tools-6.3-rpms
```

3. パッケージをインストールします。

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

4.7.2. Capsule Server でリモート実行の有効化

Capsule Server のホストでコマンドを実行する場合は、リモート実行が有効である必要があります。



注記

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

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

4.7.3. Capsule Server へのライフサイクル環境の追加

Capsule Server でコンテンツ機能が有効な場合は、1 つ以上のライフサイクル環境を追加する必要があります。環境を追加すると、Capsule Server で Satellite Server のコンテンツを同期し、コンテンツをホストシステムに提供できます。

Red Hat では、1 つ以上のライフサイクル環境を作成して、Capsule Server に割り当てることを推奨しています。こうすることで、Capsule は各ライフサイクル環境にプロモートされたコンテンツビューに含まれているリポジトリのみから受け取ることになり、システムリソースの使用が最適化されます。



注記

ライブラリーライフサイクル環境を Capsule Server に割り当てると、リポジトリが 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、これは避けてください。自動同期では、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ます。

Capsule Server は、Satellite Server 上で Hammer CLI を使用するか、Web UI で設定できます。

Hammer CLI を使用して Capsule Server にライフサイクル環境の追加

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

```
# hammer capsule list
```

3. ID を使用して、Capsule Server の詳細を確認します。

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

4. 利用可能なライフサイクル環境を確認し、環境 ID をメモします。

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

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

5. ライフサイクル環境を **Capsule Server** に追加します。

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

6. **Capsule Server** に追加する各ライフサイクル環境に対して手順を繰り返します。

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

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

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

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

ライフサイクル環境での作業に関する詳細は、『**Red Hat Satellite コンテンツ管理ガイド**』の「[アプリケーションライフサイクルの作成](#)」を参照してください。

Web UI を使用して **Capsule Server** にライフサイクル環境の追加

1. **Satellite Web UI** で、インフラストラクチャー > カプセルに移動し、カプセルを選択します。
2. **編集** をクリックします。
3. ライフサイクル環境タブで **Env** を選択します。
4. カプセルのコンテンツを同期するには、概要タブの **同期** ボタンをクリックします。
5. 以下のオプションのいずれかを選択します。
 - **Optimized Sync**
 - **Complete Sync**

4.7.4. 管理対象ホスト上での電源管理の有効化

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

Satellite Capsule サーバー上の BMC サービスを使用すると、さまざまな電源管理タスクを実行できます。この機能の基礎となるプロトコルは IPMI です (BMC 機能とも呼ばれます)。IPMI は、ホストの CPU から独立して実行する専用プロセッサに接続された管理対象ハードウェア上で、特別なネット

ワークインターフェースを使用します。多くのインスタンスでは、**BMC** 機能はシャーシ管理の一部としてシャーシベースのシステムに組み込まれます (シャーシの専用モジュール)。

BMC サービスの詳細は『ホストの管理』の「[追加のネットワークインターフェースの設定](#)」を参照してください。

作業開始前の準備

- すべての管理対象ホストに **BMC** タイプのネットワークインターフェースが搭載されている必要があります。Satellite はこの NIC を使用して適切な認証情報をホストに渡します。

管理対象ホスト上での電源管理の有効化

1. オプションを使用してインストーラーを実行し、**BMC** を有効にします。

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

4.7.5. Capsule Server での DNS と DHCP の設定

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

Capsule Server が外部 DNS および DHCP サービスを使用するよう設定することもできます。詳細は「[5章 外部サービスの設定](#)」を参照してください。

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

作業開始前の準備

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

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

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

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

```
--foreman-proxy-dns-interface=eth0 \
--foreman-proxy-dns-zone=example.com

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

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

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

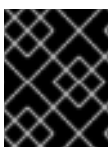
前提条件

- カスタム証明書が設定された Satellite サーバー。詳細は「[カスタムサーバー証明書を使用した Satellite Server の設定](#)」を参照してください。
- インストールされ Satellite Server に登録された Capsule サーバー。詳細は「[4章 Capsule Server のインストール](#)」を参照してください。

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

1. 「[Capsule Server 向けの SSL 証明書の取得](#)」
2. 「[Capsule Server の SSL 証明書の検証](#)」
3. 「[Capsule サーバーの証明書アーカイブファイルの作成](#)」
4. 「[Capsule Server のカスタム証明書のインストール](#)」
5. 「[すべてのホストへの Capsule Server の新しい証明書のインストール](#)」

4.7.6.1. Capsule Server 向けの SSL 証明書の取得



重要

この手順では、PEM エンコードの証明書が生成されます。SSL 証明書には、PEM エンコードのみを使用してください。



注記

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

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

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

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

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



注記

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

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

3. Certificate Signing Request (CSR) を作成します。

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



注記

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

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

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

- ① 証明書を署名するために使用される Capsule サーバーの秘密鍵
- ② Certificate Signing Request ファイル

Certificate Signing Request セッションの例

```
You are about to be asked to enter information that will be
incorporated
```

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

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

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

4. 証明書署名要求を認証局に送信します。

要求を送信する場合は、証明書のライフスパンを指定する必要があります。証明書署名要求を送信する方法は異なるため、推奨される方法について認証局にお問い合わせください。要求に対する応答で、認証局バンドルと署名済み証明書を別々のファイルで受け取ることになります。

4.7.6.2. Capsule Server の SSL 証明書の検証

Satellite サーバーで、**katello-certs-check** コマンドを使用して Capsule サーバーの証明書入力ファイルを検証します。このプロセスでは、Capsule Server の鍵、CSR、および SSL 証明書が Capsule Server から Satellite Server にコピーされている必要があります。

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

- ① 認証局により提供された Capsule サーバー証明書ファイル
- ② 証明書を署名するために使用される Capsule サーバーの秘密鍵
- ③ Capsule サーバーの証明書署名要求ファイル
- ④ 認証局により提供された認証局バンドル

証明書が正常に検証された場合、出力には以下の情報が含まれます。

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

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

「[Capsule サーバーの証明書アーカイブファイルの作成](#)」に進みます。

4.7.6.3. Capsule サーバーの証明書アーカイブファイルの作成

Capsule サーバーのインストーラーでは、サーバーの証明書がアーカイブファイルで必要になります。このファイルを作成するには、Satellite Server で **capsule-certs-generate** コマンドを使用します。

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

capsule-certs-generate コマンドに使用するパラメーターは、「[Satellite Server の SSL 証明書の検証](#)」で得た **katello-certs-check** コマンドの出力を使用します。

1. エディターで **capsule-certs-generate** コマンドのコピーを準備します。
2. Capsule Server の FQDN に一致するよう **--foreman-proxy-fqdn** の値を編集し、証明書アーカイブファイルのファイルパスおよび名前に一致するよう **--certs-tar** の値を編集します。
3. Capsule Server をインストールしていない場合は、**--certs-update-server** パラメーターを削除します。これは、既存の Capsule Server の証明書を更新するためにのみ使用されます。
4. 編集した **capsule-certs-generate** コマンドをテキストエディターから端末にコピーします。
5. 編集した **capsule-certs-generate** コマンドを実行します。

capsule-certs-generate コマンドの例

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

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

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

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

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

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



警告

この手順は、Capsule サーバーで完了してください。

Capsule サーバーのカスタム証明書をインストールするには、**satellite-installer** スクリプトをカスタムパラメーターで実行します。コマンドとパラメーターは、「[Capsule サーバーの証明書アーカイブファイルの作成](#)」の**capsule-certs-generate** コマンドで得た出力を使用します。

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

カスタム **satellite-installer** コマンドの例

```
# satellite-installer --scenario capsule \
--foreman-proxy-content-parent-fqdn "satellite.example.com" \
--foreman-proxy-register-in-foreman "true" \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-trusted-hosts "satellite.example.com" \
--foreman-proxy-trusted-hosts "capsule.example.com" \
--foreman-proxy-oauth-consumer-key
"FeQsbASvCjvvaqE6duKH6SoYZWg4jwjg" \
--foreman-proxy-oauth-consumer-secret
"7UhPXFDBongvdTbNixbsWR5WFZsKEgF" \
--foreman-proxy-content-pulp-oauth-secret
"VpQ9587tVmYeuY4Du6VitmZpZE5vy9ac" \
--foreman-proxy-content-certs-tar "/root/capsule_certs.tar"
```




注記

satellite-installer コマンドの値は、**capsule-certs-generate** コマンドの出力からも分かるように、各 Capsule Server に対して一意です。したがって、複数の Capsule Server で同じコマンドを使用しないでください。

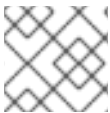
証明書が関連するすべてのホストにデプロイされた後であっても、証明書アーカイブファイル(.tar ファイル)は削除しないでください。このファイルは、たとえば、Capsule サーバーをアップグレードする際に必要になります。証明書アーカイブファイルがインストーラーによって検出されない場合は、以下のようなメッセージが出て失敗します。

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

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

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

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



注記

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

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

4.7.7. mongod へのアクセスの制限

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

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

1. ファイヤーウォールを設定します。

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

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

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

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

```
&& firewall-cmd --permanent --direct --add-rule ipv6 filter OUTPUT 1  
\  
-o lo -p tcp -m tcp --dport 28017 -j DROP
```

第5章 外部サービスの設定

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

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

5.1. 外部 DNS を使用した **SATELLITE** の設定

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

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

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

2. ドメインの設定ファイルを作成します。

以下の例では、ドメイン **virtual.lan** を1つのサブネット **192.168.38.0/24** として設定し、**capsule** という名前のセキュリティーキーを設定して、フォワーダーを Google のパブリック DNS アドレス (**8.8.8.8** および **8.8.4.4**) に設定します。**192.168.38.2** は DNS サーバーの IP アドレスで、**192.168.38.1** は、**Satellite Server** または **Capsule Server** の IP アドレスになります。

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

controls {
    inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { "capsule";
};
    inet 192.168.38.2 port 953 allow { 192.168.38.1; 192.168.38.2; }
keys { "capsule"; };
};

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

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

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

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

```
};
```

設定ファイルの **inet** 行は、1つの行として入力する必要があります。

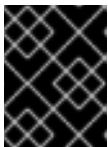
3. キーファイルを作成します。

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

このコマンドが完了するまで、しばらく時間がかかることがあります。

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

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



重要

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

5. ゾーンファイルを作成します。

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

$ORIGIN virtual.lan.
$TTL 86400      ; 1 day
capsule         A       192.168.38.1
service         A       192.168.38.2
```

6. 逆引きゾーンファイルを作成します。

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

```

                                3600          ; minimum (1 hour)
                                )
                                NS      service.virtual.lan.
$ORIGIN 38.168.192.in-addr.arpa.
$TTL 86400          ; 1 day
1          PTR      capsule.virtual.lan.
2          PTR      service.virtual.lan.

```

ASCII 以外の他の文字は使用しないでください。

5.2. DNS サービスの開始と起動

1. 構文を検証します。

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

2. サーバーを起動します。

```
# systemctl restart named
```

3. 新しいホストを追加します。

以下のコマンドでは、ホストの例 **192.168.38.2** を使用しています。この値は、ご使用の環境に合わせて変更してください。

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

4. DNS サービスが新しいホストを解決できることを確認します。

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

5. 必要な場合は、新しいエントリーを削除します。

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

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

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

5.3. CAPSULE SERVER での外部 DNS の設定

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

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

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

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

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

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

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

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

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

5. **satellite-installer** スクリプトを実行して、以下の永続的な変更を `/etc/foreman-proxy/settings.d/dns.yml` ファイルに加えます。

```
# satellite-installer --foreman-proxy-dns=true \
--foreman-proxy-dns-managed=false \
--foreman-proxy-dns-provider=nsupdate \
--foreman-proxy-dns-server="192.168.38.2" \
--foreman-proxy-keyfile=/etc/rndc.key \
--foreman-proxy-dns-ttl=86400
```

6. **foreman-proxy** サービスを再起動します。

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

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

5.4. SATELLITE SERVER での外部 DHCP の設定

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

DHCP サーバーの設定、DHCP 設定およびリースファイルの共有

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

```
# yum install dhcp bind
```

2. 空のディレクトリーでセキュリティトークンを生成します。

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

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

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

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

3. キーからシークレットハッシュをコピーします。

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

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

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

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

omapi-port 7911;
key omapi_key {
    algorithm HMAC-MD5;
    secret "jNSE5YI3H1A80j/tkV4...A2Z0Hb6zv315CkNAY7DMYYCj48Umw==";
};
omapi-key omapi_key;
```

5. 2 つのキーファイルを、それらを作成したディレクトリーから削除します。

6. **Satellite Server** で各サブネットを定義します。

競合を回避するために、リース範囲と予約範囲は別々に設定することが推奨されます。たとえば、リース範囲を **192.168.38.10** から **192.168.38.100** とし、予約範囲 (**Satellite Web UI** で定義済み) を **192.168.38.101** から **192.168.38.250** とします。ここでは、定義されたサブネットに **DHCP Capsule** を設定しないでください。

7. ファイアウォールで **DHCP** サーバーへの外部アクセスを設定します。

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

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


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

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

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

10. 設定ファイルを読み取り可能にするために、読み取りおよび実行フラグを復元します。

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

11. DHCP サービスを起動します。

```
# systemctl start dhcpd
```

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

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

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

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

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

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

15. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

16. **/etc/exports** に以下の行が存在することを確認します。

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

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

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

17. NFS サーバーをリロードします。

```
# exportfs -rva
```

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

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

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

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

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

Satellite Server の設定

1. **NFS** クライアントをインストールします。

```
# yum install nfs-utils
```

2. **NFS** 用の **DHCP** ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

3. ファイルの所有者を変更します。

```
# chown -R foreman-proxy /mnt/nfs
```

4. **NFS** サーバーとの通信と **RPC** 通信パスを検証します。

```
# showmount -e your_DHCP_server_FQDN
# rpcinfo -p your_DHCP_server_FQDN
```

5. **/etc/fstab** ファイルに以下の行を追加します。

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

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

6. **/etc/fstab** 上のファイルシステムをマウントします。

```
# mount -a
```

7. 関連するファイルを読み取ります。

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

8. **satellite-installer** スクリプトを実行して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/dhcp.yml** ファイルに加えます。

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config
/mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases
/mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A80j/tkV4...A2Z0Hb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=your_DHCP_server_FQDN
```

9. foreman-proxy サービスを再起動します。

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

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

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

12. DHCP サービスに適切なサブネットとドメインを関連付けます。

5.5. CAPSULE SERVER での外部 DHCP の設定

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

前提条件

- DHCP サーバーが設定済みで、DHCP 設定とリースファイルが NFS 経由で共有されています。詳細は「[Satellite Server での外部 DHCP の設定](#)」を参照してください。

1. NFS クライアントをインストールします。

```
# yum install nfs-utils
```

2. NFS 用の DHCP ディレクトリーを作成します。

```
# mkdir -p /mnt/nfs/etc/dhcp /mnt/nfs/var/lib/dhcpd
```

3. ファイルの所有者を変更します。

```
# chown -R foreman-proxy /mnt/nfs
```

4. NFS サーバーとの通信と RPC 通信パスを検証します。

```
# showmount -e your_DHCP_server_FQDN
# rpcinfo -p your_DHCP_server_FQDN
```

5. **/etc/fstab** ファイルに以下の行を追加します。

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

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

6. **/etc/fstab** 上のファイルシステムをマウントします。

```
# mount -a
```

7. 関連するファイルを読み取ります。

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

8. **satellite-installer** スクリプトを実行して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/dhcp.yml** ファイルに加えます。

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config
/mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases
/mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A80j/tkV4...A2Z0Hb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=your_DHCP_server_FQDN
```

9. **foreman-proxy** サービスを再起動します。

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

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

11. インフラストラクチャー > **Capsules** に移動します。適切な **Capsule Server** を見つけ、アクション ドロップダウンリストから **更新** を選択します。この結果、DHCP 機能が現れます。
12. DHCP サービスに適切なサブネットとドメインを関連付けます。

5.6. SATELLITE SERVER での外部 TFTP の設定

作業開始前の準備

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

Satellite Server での外部 TFTP の設定

1. TFTP サーバーをインストールし、有効にします。

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

2. **tftp.socket** ユニットを有効にし、アクティベートします。

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

3. PXELinux 環境を設定します。

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

4. SELinux ファイルコンテキストを復元します。

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

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

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

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

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

7. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

8. **/etc/exports** に以下の行があることを確認します。

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

```
/exports/var/lib/tftpboot
192.168.38.1(rw,async,no_root_squash,no_subtree_check,nohide)
```

最初の行は DHCP 設定に共通で、このシステムで以前の手順を完了すると作成されます。

9. NFS サーバーをリロードします。

```
# exportfs -rva
```

5.6.1. ファイアウォールでの TFTP への外部アクセスの設定

1. ファイアウォールを設定します (ポート 69 上の UDP)。

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

5.7. CAPSULE SERVER での外部 TFTP の設定

1. NFS を準備するために TFTP ディレクトリーを作成します。

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

2. **/etc/fstab** ファイルで以下の行を追加します。

```
192.168.38.2:/exports/var/lib/tftpboot /mnt/nfs/var/lib/tftpboot nfs
rw,vers=3,auto,nosharecache,context="system_u:object_r: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** スクリプトを実行して、以下の永続的な変更を **/etc/foreman-proxy/settings.d/tftp.yml** ファイルに加えます。

```
# satellite-installer --foreman-proxy-tftp=true \
--foreman-proxy-tftp-root /mnt/nfs/var/lib/tftpboot
```

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

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

この結果、すべての設定ファイルが新しい値で更新されます。

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

5.8. SATELLITE 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لم登録機能を使用して、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لم登録は、相互排他的になります。レلم登録の詳細は『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。

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

5.8.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

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

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

IdM サーバーの設定は以下のようになります。

ホスト名	idm1.example.com
ドメイン名	example.com

作業開始前の準備

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

IdM サーバー上で Kerberos プリンシパルの作成

1. Kerberos チケットがあることを確認します。

```
# kinit idm_user
```

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

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

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

IdM クライアントのインストールと設定

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

1. IdM クライアントパッケージをインストールします。

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

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

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

3. Kerberos チケットがあることを確認します。

```
# kinit admin
```

4. 既存の keytab を削除します。

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

5. このシステム用に作成された keytab を取得します。


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



注記

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

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

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

7. 必要に応じて、**keytab** が有効か確認します。

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

IdM web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。
 - a. **Network Services** (ネットワークサービス) > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。
 - b. **Add** を選択し、ゾーン名を入力します。この例では、**example.com** になります。
 - c. **Add and Edit** をクリックします。
 - d. 設定タブの **BIND update policy** ボックスで、以下のようにセミコロン区切りのエントリを追加します。

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

- e. **Dynamic update** が **True** に設定されていることを確認します。
 - f. **Allow PTR sync** を有効にします。
 - g. **Save** を選択して、変更を保存します。
2. 逆引きゾーンを作成、設定します。
 - a. **Network Services** (ネットワークサービス) > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。
 - b. **Add** を選択します。
 - c. **Reverse zone IP network** を選択して、CIDR 形式でネットワークアドレスを追加し、逆引き参照を有効にします。
 - d. **Add and Edit** をクリックします。

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

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

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

- g. **Save** を選択して、変更を保存します。

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

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

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

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

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

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

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

Satellite web UI での設定更新

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

1. インフラストラクチャー > **Capsules (スマートプロキシー)** に移動します。
2. 更新する **Capsule** で、アクション ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。

インフラストラクチャー > ドメイン > 移動 | ドメインを選択 | 更新

- d. **192.168.25.1**のサブネットに移動し、サブネット名を選択します。
 - b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている **Capsule** に選択されていることを確認します。
4. サブネットを設定します。
- a. **インフラストラクチャー > サブネット** に移動し、サブネット名を選択します。
 - b. **サブネット** タブで、**IPAM** を **None** に設定します。
 - c. **ドメイン** タブで、**IdM** サーバーが管理するドメインが選択されていることを確認します。
 - d. **Capsules** タブで、**Reverse DNS Capsule** 、サブネットが接続されている **Capsule** に選択されていることを確認します。
 - e. **送信** をクリックして変更を保存します。

5.8.2. 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

- 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。
- IdM** サーバーで **root** 権限を取得します。
- 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。
- 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。
- 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。
- 変更を元に戻す場合に備えて、応答ファイルのバックアップを作成します。詳細は「[インストールオプションの指定](#)」を参照してください。

IdM サーバーの DNS ゾーンに対する外部アップデートの有効化

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

```
// This was added to allow Satellite Server at 192.168.25.1 to make
DNS updates.
#####
####
include "/etc/rndc.key";
controls {
inet 192.168.25.2 port 953 allow { 192.168.25.1; } keys { "rndc-
key"; };
};
#####
####
```

2. `named` をリロードして、変更を有効にします。

```
# systemctl reload named
```

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

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

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

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

- c. **Update (更新)** をクリックして変更を保存します。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5. **IdM Web UI** でエントリーを参照するために、**Network Services (ネットワークサービス) > DNS > DNS Zones (DNS ゾーン)** に移動します。ゾーンの名前を選択し、名前でホストを検索します。
6. 正常に解決されたら、テスト **DNS** エントリーを削除します。

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

7. **DNS** エントリーが削除されたことを確認します。

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

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

5.8.3. 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。
応答ファイルについての詳細は「[インストールオプションの指定](#)」を参照してください。

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 と DHCP 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)** に移動します。
2. 更新する **Capsule** で、アクション ドロップダウンメニューから **更新** を選択します。
3. ドメインを設定します。
 - a. インフラストラクチャー > **ドメイン** に移動し、ドメイン名を選択します。
 - b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている **Capsule** に選択されていることを確認します。
4. サブネットを設定します。
 - a. インフラストラクチャー > **サブネット** に移動し、サブネット名を選択します。
 - b. **サブネット** タブで、**IPAM** を **DHCP** または **Internal DB** に設定します。
 - c. **ドメイン** タブで、**Satellite** または **Capsule** が管理するドメインが選択されていることを確認します。
 - d. **Capsules** タブで、**Reverse DNS Capsule** 、サブネットが接続されている **Capsule** に選択されていることを確認します。
 - e. **送信** をクリックして変更を保存します。

第6章 SATELLITE SERVER および CAPSULE SERVER のアンインストール

Satellite Server または Capsule Server は、不要になったらアンインストールできます。

6.1. SATELLITE SERVER のアンインストール

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

作業開始前の準備

katello-remove スクリプトを実行すると、2つの警告が出され、システムのすべてのパッケージと設定ファイルを削除する前に確認が求められます。



警告

このスクリプトは、以下の重要なパッケージを含む、多くのパッケージおよび設定ファイルを削除します。

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

Satellite Server のアンインストール

1. Satellite Server をアンインストールします。

```
# katello-remove
```

次のようなメッセージが表示されます。

```
Once these packages and configuration files are removed there is no  
going back.  
If you use this system for anything other than Katello and Foreman  
you probably  
do not want to execute this script.  
Read the source for a list of what is removed.  Are you sure(Y/N)? y
```

```
ARE YOU SURE?: This script permanently deletes data and
configuration.
Read the source for a list of what is removed.  Type [remove] to
continue? remove
Shutting down Katello services...
```

6.2. CAPSULE SERVER のアンインストール

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

作業開始前の準備

katello-remove スクリプトを実行すると、2つの警告が出され、システムのすべてのパッケージと設定ファイルを削除する前に確認が求められます。



警告

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

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

Capsule Server のアンインストール

1. Capsule Server をアンインストールします。

```
# katello-remove
```

次のようなメッセージが表示されます。

```
Once these packages and configuration files are removed there is no
going back.
If you use this system for anything other than Katello and Foreman
you probably
do not want to execute this script.
Read the source for a list of what is removed.  Are you sure(Y/N)? y
ARE YOU SURE?: This script permanently deletes data and
```


configuration.

Read the source for a list of what is removed. Type [remove] to
continue? remove

Shutting down Katello services...

第7章 詳細情報の提供元

最初のインストールおよびセットアップの最後に、追加設定を実行し、**Satellite** 環境をセットアップできます。詳細は、以下の **Satellite** ドキュメンテーションリソースを参照してください。

- [Hammer CLI Guide](#)
- [Administering Red Hat Satellite](#)
- [Managing Hosts](#)
- [Content Management Guide](#)
- [Puppet Guide](#)
- [Virtual Instances Guide](#)

第8章 AMAZON WEB SERVICES 上での RED HAT SATELLITE の実行

本章では、Red Hat Satellite Server および Capsules を Amazon Web Services (AWS) Elastic Compute Cloud (Amazon EC2) にインストールする準備について説明します。

AWS 上での Satellite および Capsule インストールで利用可能なアーキテクチャ設定は「[デプロイメントシナリオ](#)」を参照してください。

前提条件 セクションでは、Red Hat Satellite のインストールに際しての Red Hat と Amazon Web サービスの準備について説明しています。

サブスクリプション

すべてのサブスクリプションがパブリッククラウド環境で実行できるわけではありません。サブスクリプションを実行できるかどうかの適格性は、「[Cloud Access Page](#)」を参照してください。新たな組織を作成して、その組織に新たなマニフェストをインポートすることもできます。詳細は『[コンテンツ管理ガイド](#)』の「[組織の作成](#)」を参照してください。

8.1. ユースケースにおける留意点

Amazon Web サービスは画像のみのサービスなので、Satellite のユースケースによっては、Amazon Web サービス環境では動作しない、または追加設定が必要な場合があります。AWS で Satellite を使用する予定がある場合は、使用するユースケースのシナリオが AWS 環境で利用可能であることを確認してください。

8.1.1. 機能するユースケース

以下の Red Hat Satellite ユースケースは、AWS で機能することがわかっています。

- [Subscription Management](#)
- [Content Management](#)
- [Errata Management](#)
- [Configuring Hosts](#)
- [Red Hat Insights](#)
- [Managing Containers](#)
- [Realm Integration via IdM](#)
- [OpenSCAP](#)
- [Remote Execution](#)

マルチホームの Satellite および Capsule

Satellite で異なるホスト名を持つ複数のインタフェースを使用したい場合は、Satellite Server と Satellite Capsule Server の CA 証明書に追加設定を行う必要があります。この設定での Satellite のデプロイをご希望の場合は、Red Hat までご連絡ください。

Satellite Server または Capsule Server の内部および外部 DNS ホスト名が異なり、かつ Satellite Server と Capsule Server を配置するロケーション間にサイト間 VPN 接続がない場合は、この設定を実行する必要があります。

オンデマンドのコンテンツソース

オンデマンドダウンロードポリシーを使用すると、Satellite を実行する Red Hat Enterprise Linux サーバーのストレージフットプリントを削減できます。ダウンロードポリシーを **オンデマンド** に設定すると、コンテンツホストの要求時に、Satellite Server または Capsule Server にコンテンツが同期されます。

詳細は『**Red Hat Satellite コンテンツ管理ガイド**』の「[Red Hat コンテンツのインポート](#)」を参照してください。

8.1.2. 機能しないユースケース

AWS では、DHCP を管理できません。このため、Satellite Server のほとんどのキックスタートと PXE プロビジョニングモデルは使用できないことになります。これには、以下のものが含まれます。

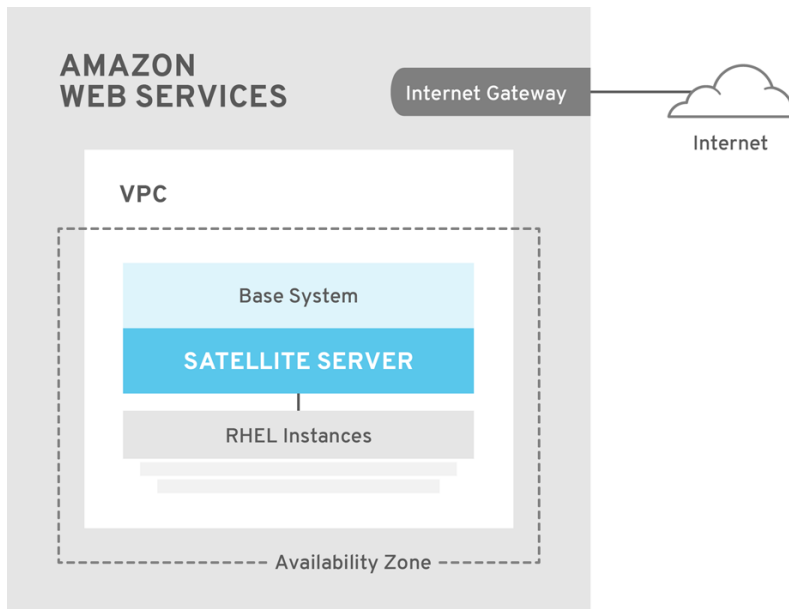
- PXE プロビジョニング
- 検出および検出ルール
- ISO プロビジョニングメソッド
 - PXE-Less Discovery (iPXE) (PXE を使用しない検出)
 - Per-host ISO
 - Generic ISO
 - Full-host ISO

8.2. デプロイメントシナリオ

Amazon Web Services では、以下の 3 つの Red Hat Satellite デプロイメントシナリオがあります。

1. 単一領域設定
2. オンプレミスと AWS 領域の接続
3. 異なる領域の接続

シナリオ 1: 単一領域設定



SATELLITE_465517_0118

Amazon Web Services における最もシンプルな Satellite Server 設定は、Satellite Server とコンテンツホストの両方が同一領域内でかつ仮想プライベートクラウド (VPC) 内にあるという構成です。

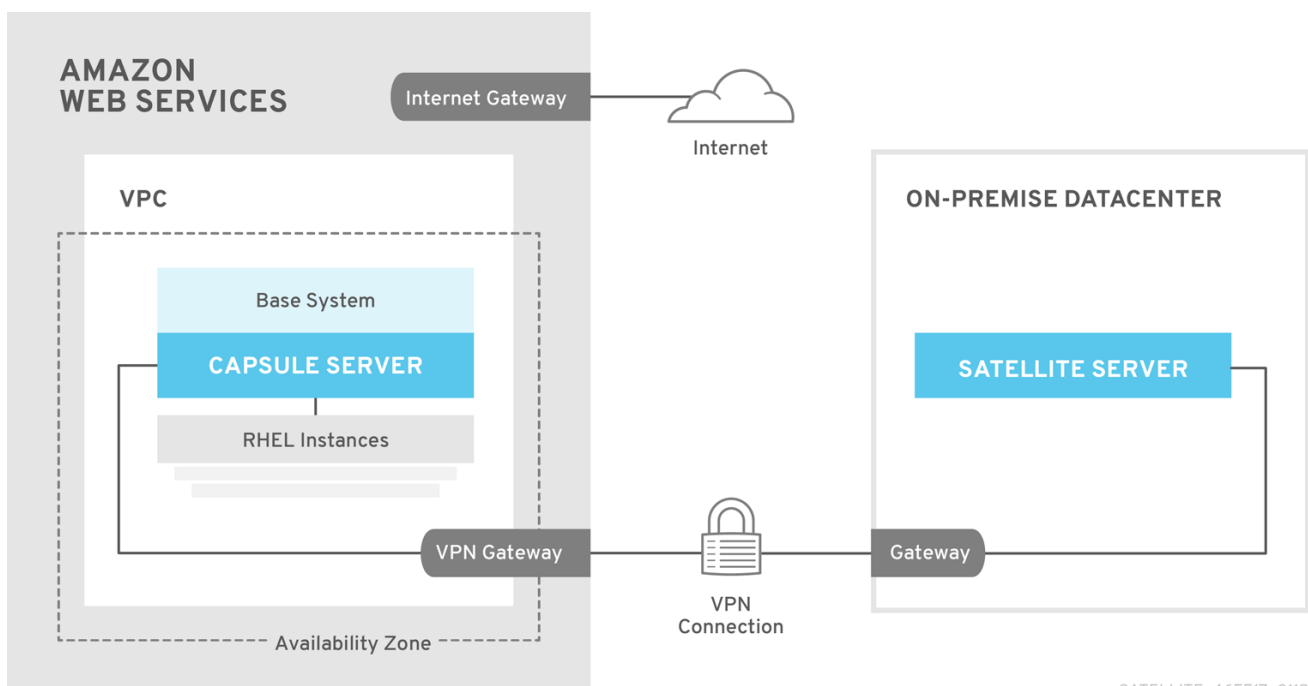
別のアベイラビリティゾーンを使用することもできます。

シナリオ 2: オンプレミスと AWS 領域の接続

この場合は、オンプレミスのロケーションと Capsule を設置する AWS 領域の間に VPN 接続を作成します。

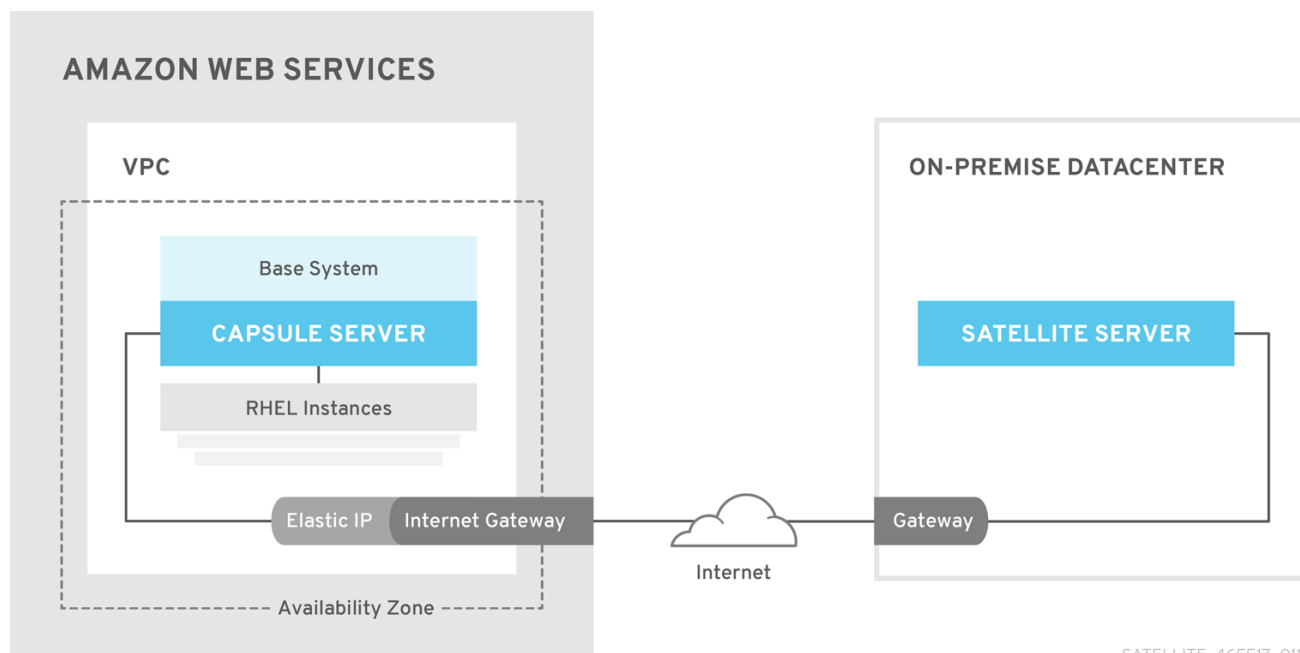
Capsule Server を実行するインスタンスを登録すれば、Satellite Server の外部のホスト名を使用することも可能です。

オプション 1: AWS 領域とオンプレミスデータセンターをつなぐサイト間 VPN 接続



SATELLITE_465517_0118

オプション 2: 外部 DNS ホスト名を使用した直接接続



シナリオ 3: 異なる領域の接続

この場合は、別の領域をつなぐサイト間 VPN 接続を作成し、**Capsule Server** を実行するインスタンスを **Satellite Server** に登録する際に、内部 DNS ホスト名を使用できるようにします。

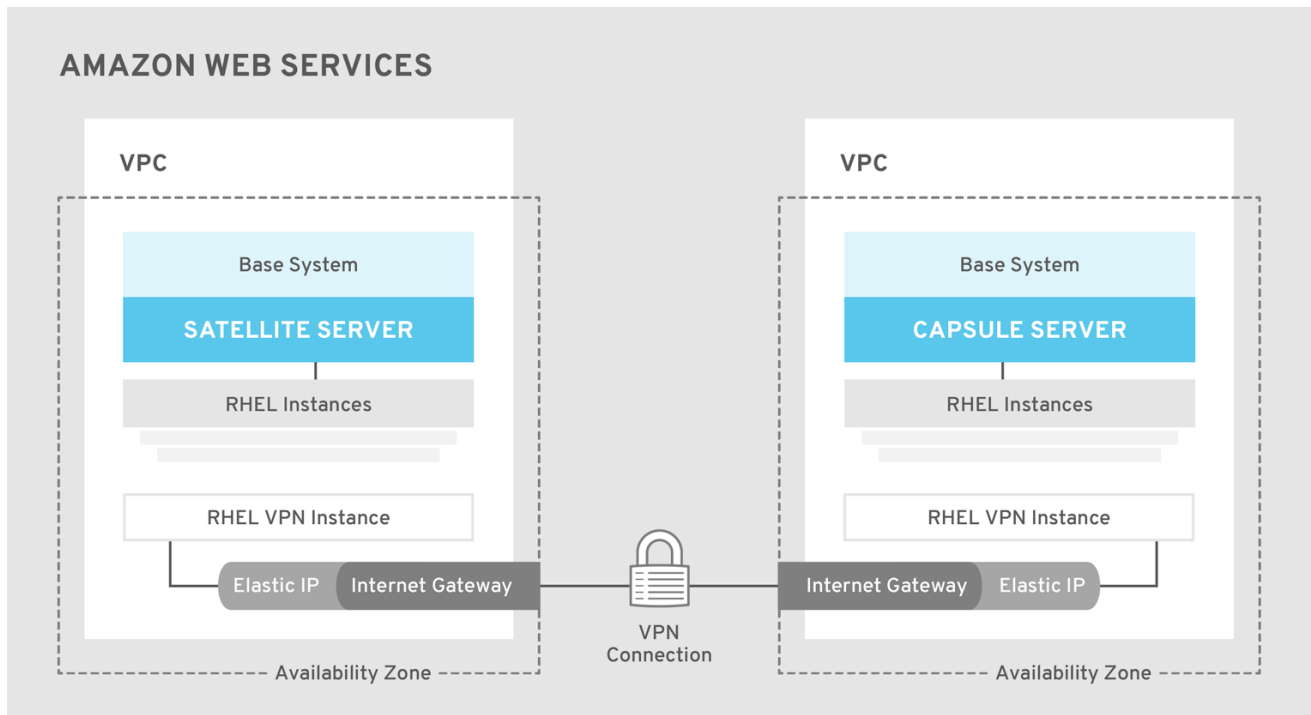
サイト間 VPN 接続を確立しない場合には、**Capsule Server** を実行するインスタンスを **Satellite Server** に登録する際に、外部 DNS ホスト名を使用します。



注記

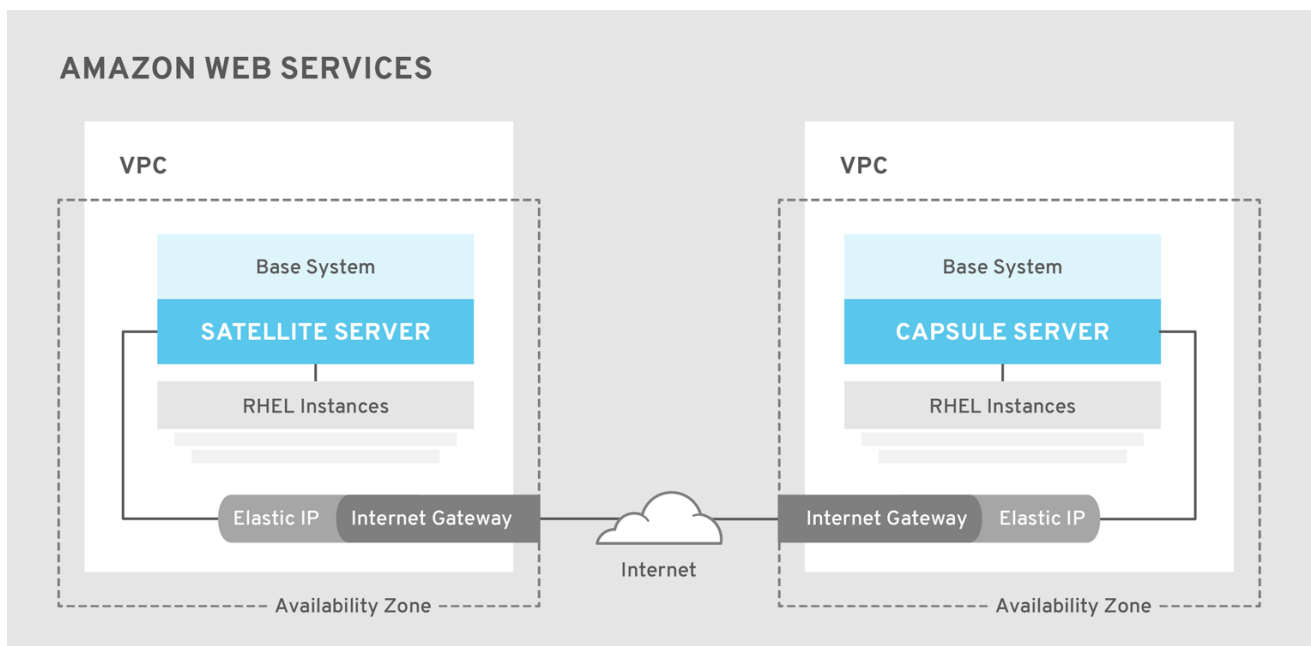
ほとんどのパブリッククラウドプロバイダーでは、領域へのデータ転送や、単一領域内でのアベイラビリティゾーン間でのデータ転送には課金されませんが、領域からインターネットへのデータ転送には課金されます。

オプション 1: AWS 領域間をつなぐサイト間 VPN 接続



SATELLITE_465517_0118

オプション 2: 外部 DNS ホスト名を使用した直接接続



SATELLITE_465517_0118

8.3. 前提条件

Red Hat Satellite および Capsule のインストールと登録前には、Amazon Web Services (AWS) でアカウントを設定し、AWS 上で Red Hat Enterprise Linux インスタンスを作成して起動する必要があります。

8.3.1. Amazon Web Service の前提条件

本ガイドでは、ユーザーが Amazon Web Services の以下の点について作業可能な知識を備えていることを前提としています。

- AWS での Red Hat Enterprise Linux イメージの作成とアクセス。
- AWS セキュリティーにおけるネットワークアクセスの編集。
- EC2 インスタンスの作成と EBS ボリュームの作成。
- インスタンスの開始。
- AWS での仮想マシンのインポートおよびエクスポート。
- AWS Direct Connect の使用。

Satellite と Capsule を AWS 環境にインストールするには、お使いの AWS 設定が「[システム要件](#)」を満たしている必要があります。

Amazon Web Services と関連用語に関する情報は、[Amazon Elastic Compute Cloud ドキュメント](#) を参照してください。

Amazon Web Services Direct Connect に関する詳細情報は、「[AWS Direct Connect とは何ですか](#)」を参照してください。

8.3.2. Red Hat Cloud の前提条件

本ガイドでは、以下のステップを完了していることが必要になります。

- Red Hat Cloud Access での登録
- 使用する Red Hat サブスクリプションの移行
- AWS インスタンスを作成し、Red Hat Enterprise Linux 仮想マシンをそのインスタンスにデプロイすること
- お使いのサブスクリプションが Red Hat Cloud に移行可能であることを確認。詳細は「[Red Hat Cloud Access プログラムの詳細](#)」を参照してください。

Red Hat Enterprise Linux の AWS でのデプロイに関する情報は、「[How to Locate Red Hat Cloud Access Gold Images on AWS EC2](#)」を参照してください。

8.3.3. Red Hat Satellite 固有の前提条件

- Amazon EC2 インスタンスのタイプが「[2章 インストールのための環境準備](#)」にあるシステムおよびストレージ要件を満たしていることを確認します。ベストパフォーマンスをお求めの場合は、AWS Storage Optimized インスタンスをお使いください。
- 「[2章 インストールのための環境準備](#)」を参照して、適切なストレージを把握して、AWS EBS ボリュームに割り当てます。
- ブートボリュームとは別の EBS ボリューム上に同期コンテンツを保存します。
- 同期した EBS ボリュームをオペレーティングシステム内で個別にマウントします。
- オプションで、**mongodb** ディレクトリーなどの他のデータを別個の EBS ボリュームに保存します。
- Satellite Server と Capsule Server が外部 DNS ホスト名を使用して通信するようにするには、インスタンスに関連付けられた AWS セキュリティーグループで、通信用に必要なポートを開きます。

8.3.4. Red Hat Satellite のインストール準備

お使いの AWS 環境で、以下のステップを完了させます。

1. Red Hat Enterprise Linux AMI の EC2 インスタンスを起動します。
2. 新規作成のインスタンスに接続します。

Red Hat Gold Image を使用している場合は、以下のように RHUI クライアントを削除して、**product-id.conf** の **enabled** パラメーターを **1** に設定します。

```
# yum -y remove rh-amazon-rhui-client*
# yum clean all
# cat << EOF > /etc/yum/pluginconf.d/product-id.conf
> [main]
> enabled=1
> EOF
```

8.4. AWS での SATELLITE SERVER のインストール

お使いの AWS 環境で、以下のステップを完了させます。

1. 新規インスタンスに接続します。
2. 「[3章 Satellite Server のインストール](#)」の手順に従って、Satellite Server をインストールします。

8.5. AWS での CAPSULE のインストール

お使いの AWS 環境で、以下のステップを完了させます。

1. 新規インスタンスに接続します。
2. 「[4章 Capsule Server のインストール](#)」の手順に従って、Capsule Server をインストールします。

8.6. ブートストラップスクリプトを使ったホストの SATELLITE への登録

Satellite Server および Capsule Server をインストールしたら、ブートストラップスクリプトを使用して、EC2 インスタンス上のコンテンツホストを Satellite に登録する必要があります。

ブートストラップスクリプトの使用方法は、『ユーザーガイド』の「[ブートストラップスクリプトを使ったホストの Satellite への登録](#)」を参照してください。

Katello Agent をインストールします。詳細は「[katello エージェントのインストール](#)」を参照してください。

付録A 大規模デプロイメントに関する考慮事項

Apache 向けファイル記述子の最大数の増加

登録されているコンテンツホストの数が 800 を超える場合、Apache では複数のシステムレベルの制限に到達し、新しいコンテンツホストの登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

1. `/etc/systemd/system/httpd.service.d/limits.conf` ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
```

3. Katello サービスを再起動します。

```
# katello-service restart
```

qpid 向けファイル記述子の最大数の増加

1100 を超えるコンテンツホストでエラータ更新のために `goferd` を実行すると、`qpid` でシステムレベルの制限に到達し、登録に失敗することがあります。この問題を回避するには、大量のコンテンツホストをデプロイする前に、ファイル記述子の制限を緩和する必要があります。

qpid 向けファイル記述子の最大数の増加

1. `/etc/systemd/system/qpid.service.d/limits.conf` ファイルを作成し、以下のテキストを挿入します。

```
[Service]
LimitNOFILE=65536
```

2. 変更をユニットに適用します。

```
# systemctl daemon-reload
# systemctl restart qpid.service
```

共有バッファと作業メモリの増加

`shared_buffer` と `work_mem` はそれぞれ、256M と 4M に増やすことができます。

1. Red Hat Enterprise Linux 7 の場合は、`/var/lib/pgsql/data/postgresql.conf` ファイルを作成し、以下のテキストを挿入します。

```
work_mem = 4MB
shared_buffers = 256MB
```

2. `postgresql` サービスを再起動します。

```
# systemctl restart postgresql
```

同時コンテンツホスト登録の増加

システムレベルの制限に到達しないように、最大 250 の同時コンテンツホストを処理するようグローバルパッセンジャーキュー制限を増加できます。詳細は『**Red Hat Satellite のチューニング**』の「**Passenger の設定**」を参照してください。以下の手順に従って、グローバルパッセンジャーキュー制限を増やします。

1. 最大パッセンジャープールサイズを、**Satellite Server** で利用可能な物理 CPU コアの 1.5 倍に調整します。
たとえば、**Satellite Server** にコアが 16 個ある場合、最大パッセンジャープールサイズは 24 になります。この数は例として使用しているため、実際にはご使用の環境に応じた数を使用する必要があります。
2. **/etc/httpd/conf.d/passenger.conf** ファイルを編集して以下のテキストに一致するよう **IfModule** スタンザを更新します。

```
<IfModule mod_passenger.c>
    PassengerRoot /usr/share/gems/gems/passenger-
4.0.18/lib/phusion_passenger/locations.ini
    PassengerRuby /usr/bin/ruby
    PassengerMaxPoolSize 24
    PassengerMaxRequestQueueSize 200
    PassengerStatThrottleRate 120
</IfModule>
```

3. **Foreman Passenger** アプリケーション設定ファイル **/etc/httpd/conf.d/05-foreman-ssl.conf** を編集して、以下のテキストに一致するよう **PassengerAppRoot** で始まるスタンザを更新します。

```
PassengerAppRoot /usr/share/foreman
PassengerRuby /usr/bin/tfm-ruby
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com
```

4. **Puppet Passenger** アプリケーション設定ファイル **/etc/httpd/conf.d/25-puppet.conf** を編集して以下のテキストを仮想ホスト定義の最後に追加します。

```
PassengerMinInstances 6
PassengerStartTimeout 90
PassengerMaxPreloaderIdleTime 0
PassengerMaxRequests 10000
PassengerPreStart https://example.com:8140
```

5. **/var/lib/pgsql/data/postgresql.conf** ファイルで最大接続数を変更します。

```
max_connections = 500
```

6. **postgresql** サービスを再起動します。

```
# systemctl restart postgresql
```

qdrouterd 向けオープンファイルの最大数の増加

登録されているコンテンツホストの数が1000を超える場合、**qdrouterd** はオープンファイルのデフォルトの最大数に到達することがあります。この問題を回避するには、**Satellite** サーバーとすべての外部 **Capsule** サーバーのオープンファイルの最大数を増加します。

1. 以下の式を使用して、オープンファイルの必要な最大数を計算します。

$$(3 \times \text{コンテンツホストの数}) + 100$$

たとえば、1020 のコンテンツホストの場合、新しい最大数は 3160 $((3 \times 1020) + 100)$ に設定します。

2. **/etc/systemd/system/qdrouterd.service.d/limits.conf** ファイルを作成し、以下のテキストを追加します。

```
[Service]
LimitNOFILE=maximum_number_of_files
```

- a. 変更をユニットに適用します。

```
# systemctl daemon-reload
```

- b. **Satellite** のサービスを再起動します。

```
# katello-service restart
```

ホスト登録の遅れを減少

Satellite と、登録した各ホストの間の接続は、証明書を使用することで保護されています。ホストを登録すると、**Satellite** は、識別証明書と **Puppet** 証明書の 2 つを作成します。各証明書を作成するのに使用されるアルゴリズムでは、**Red Hat Enterprise Linux** カーネルのランダムなデータが必要になります。ホストを登録する際に十分なエントロピーが利用できない場合は、エントロピーのレベルが十分になるまで遅延が発生します。環境が非常に大きく、10,000 台以上のホストを使用する場合は、エントロピーが足りなくなるため、ホスト登録にかかる速度はおそらく遅くなります。**Linux** カーネルで利用可能なエントロピーを増やし、**Satellite** のパフォーマンスをリスクを減らすための方法をいくつか使用できます。

デフォルトでは、**Linux** カーネルは、乱数のソースとして **/dev/random** デバイスを使用します。このデバイスはブロックデバイスで、乱数を適切に生成するためのエントロピーの数が少ないと判断すると、乱数を提供しません。それがこの待ち時間となり、ホストの登録に遅れが発生します。この問題を解決するには、非ブロックデバイスの **/dev/urandom** デバイスを使用します。

ハードウェアサーバーの中には、ハードウェアの乱数ジェネレーターが含まれるものもあります。その乱数ジェネレーターを **Red Hat Enterprise Linux** カーネルがサポートする場合は、それを乱数のソースとして使用できます。詳細は、ハードウェアベンダーのドキュメントを参照してください。

Satellite が仮想マシンにホストされている場合、ハイパーバイザーの中には、ハードウェアサーバーの乱数ジェネレーターを、それがホストする仮想マシンで利用できるようにするものもあります。**Red Hat Virtualization** には、**Red Hat Virtualization** ホストからのエントロピーへの KVM 仮想マシンアクセスを提供する **virtio RNG** (乱数ジェネレーター) デバイスがあります。**Red Hat Enterprise Linux 7.0** を実行しているゲストには、**rngd** をインストールして設定する必要があります。**Red Hat Enterprise Linux 7.1** 以降を実行しているゲストでは、ゲストのカーネルが、必要に応じてホストからエントロピーを取得します。ホストの乱数ジェネレーターをゲストと共有している場合は、ハードウェアの乱数ジェネレーターを使用することが推奨されます。

ゲストにおける乱数ジェネレーターの詳細は Red Hat Enterprise Linux 7 『仮想化の導入および管理ガイド』の「[乱数ジェネレーターデバイス](#)」を参照してください。その他のハイパーバイザーについては、ベンダーのドキュメントを参照してください。

乱数ジェネレーターデーモン (**rngd**) の詳細は Red Hat Enterprise Linux 7 の『セキュリティガイド』の「[乱数ジェネレーターの使用](#)」を参照してください。

付録B 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 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、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の制限を示しています。

表B.1 4 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B.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

Satellite 環境の安定性およびスケーラビリティを調整する方法は『[Red Hat Satellite のチューニング](#)』を参照してください。

付録C RED HAT SATELLITE へのカスタム設定の適用

satellite-installer を使用して初めて Satellite をインストールして設定する際には、**--foreman-proxy-dns-managed=false** と **--foreman-proxy-dhcp-managed=false** のインストーラーフラグを使用することで、DNS および DHCP 設定ファイルが Puppet に管理されないようにできます。インストーラーの初回実行時のこれらのフラグを指定しない場合は、インストーラーを再実行すると手動による変更がすべて上書きされるので、アップグレードする際に再実行できます。変更が上書きされても、復元手順を実行すると、手動での変更は復元できます。詳細は『インストールガイド』の「[Puppet 実行で上書きされた手動変更の復元](#)」を参照してください。

カスタム設定に利用可能なすべてのインストーラーフラグを表示するには、**satellite-installer --scenario satellite --full-help** を実行します。Puppet クラスには、Satellite インストーラーに公開されていないものもあります。これらのクラスを手動で管理して、インストーラーが値を上書きしないようにするには、設定ファイル **/etc/foreman-installer/custom-hiera.yaml** にエントリーを追加して設定値を指定します。この設定ファイルは YAML 形式で、**<puppet class>::<parameter name>: <value>** という形式を 1 行あたり 1 エントリーで記入します。このファイルで指定した設定値は、インストーラーを再起動しても維持されます。

一般的な例を示します。

- Apache で **ServerTokens** ディレクティブが製品名のみを返すようにするには、以下のようにします。

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

- Apache サーバー署名をオフにするには、以下のようにします。

```
apache::server_signature: Off
```

- Pulp で pulp ワーカーの数を設定するには、以下のようにします。

```
pulp::num_workers: 8
```

Satellite インストーラー用の Puppet モジュールは、**/usr/share/foreman-installer/modules** と **/usr/share/katello-installer-base/modules** に保存されています。クラス、パラメーター、および値を調べるには、**.pp** ファイル (例: **moduleName/manifests/example.pp**) を確認してください。別の方法では、**grep** コマンドでキーワード検索を実行します。

値の設定によっては、Red Hat Satellite のパフォーマンスや機能に影響が出る意図しない結果がもたらされる場合があります。設定を適用する前に変更の影響を考慮して、実稼働以外の環境で最初に変更をテストしてください。実稼働以外の Satellite 環境がない場合は、Satellite インストーラーを **--noop** と **--verbose** のオプションを追加して実行します。変更によって問題が発生する場合は、該当箇所を **custom-hiera.yaml** から削除し、Satellite インストーラーを再実行します。特定の値を変更することが安全かどうかを確認したい場合は、Red Hat サポートにお問い合わせください。

C.1. PUPPET 実行で上書きされた手動変更の復元

Puppet 実行で手動による設定が上書きされた場合でも、ファイルを元の状態に戻すことができます。以下の例では、Puppet 実行で上書きされた DHCP 設定ファイルを復元します。

1. 復元するファイルをコピーします。こうすることで、アップグレードに必要な変更をファイル間で確認できます。これは DNS や DHCP サービスでは一般的ではありません。


```
# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.backup
```

2. ログファイルを確認して、上書きされたファイルの **md5sum** をメモします。たとえば、以下のようになります。

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

3. 上書きされたファイルを復元します。

```
# puppet filebucket restore --local --bucket \
/var/lib/puppet/clientbucket /etc/dhcp/dhcpd.conf \
622d9820b8e764ab124367c68f5fa3a1
```

4. バックアップしたファイルと復元されたファイルを比べます。復元されたファイルに、アップグレードに必要な変更を追加します。