Red Hat Satellite 6.7

オンラインネットワークからの Satellite Server のインストール

オンラインネットワークからの Red Hat Satellite Server のインストール

Last Updated: 2024-02-14

Red Hat Satellite 6.7 オンラインネットワークからの Satellite Server のインストール

オンラインネットワークからの Red Hat Satellite Server のインストール

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書では、オンラインネットワークから Red Hat Satellite Server のインストール方法、初期設定の実行方法、および外部サービスの設定方法を説明します。

目次

第1章 インストールのための環境準備

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

1.1. システム要件

ネットワーク接続されたベースのオペレーティングシステムには、以下の要件が適用されます。

- x86_64 アーキテクチャー
- Red Hat Enterprise Linux 7 Server の最新バージョン
- 最低 4 コア 2.0 GHz CPU
- 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 をインストールする前に、環境がインストール要件を満たしていることを確認する必要があります。

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

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

注記

Red Hat Satellite Server と Capsule Server のバージョンは同じでなければなりません。たとえば、Satellite 6.7 Server は、以前のバージョンの Capsule Server を実行できません。Satellite Server と Capsule Server のバージョンが一致しないと、Capsule Server が警告なしで失敗します。

認定ハイパーバイザー

Satellite Server は、Red Hat Enterprise Linux の実行をサポートするハイパーバイザーで稼働する物理システムおよび仮想マシン両方で完全にサポートされます。認定ハイパーバイザーに関する詳細は、「Which hypervisors are certified to run Red Hat Enterprise Linux? 」を参照してください。

FIPS モード

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

1.2. ストレージ要件

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

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

表1.1 Satellite Server インストールのストレージ要件

ディレクトリー	インストールサイズ	ランタイムサイズ
/var/cache/pulp/	1M バイト	20 GB
/var/lib/pulp/	1 MB	300 GB
/var/lib/mongodb/	3.5 GB	50 GB
/var/lib/qpidd/	25 MB	適用外
/var/log/	10 MB	10 GB
/var/lib/pgsql/	100 MB	10 GB
/var/spool/squid/	0 MB	10 GB
/usr	3 GB	適用外
/opt	3 GB	適用外
/opt/puppetlabs	500 MB	適用外

1.3. ストレージのガイドライン

Satellite Server をインストールして効率性を向上させる場合は、以下のガイドラインを考慮してください。

- **/tmp** ディレクトリーを別のファイルシステムとしてマウントする場合は、 **/etc/fstab** ファイルの **exec** マウントオプションを使用する必要があります。**/tmp** が、**noexec** オプションを指定してすでにマウントされている場合は、オプションを **exec** に変更して、ファイルシステムを再マウントする必要があります。これは、**puppetserver** サービスが機能するために必要です。
- Satellite Server データの多くは **/var** ディレクトリーに格納されるため、LVM ストレージに **/var** をマウントして、システムがスケーリングできるようにしてください。
- **/var/cache/pulp/** と **/var/lib/pulp/** ディレクトリーに同じボリュームを使用することで、同期後に **/var/cache/pulp/** から **/var/lib/pulp/** にコンテンツを移動する時間を短縮できます。
- **/var/lib/qpidd/** ディレクトリーでは、**goferd** サービスが管理するコンテンツホスト 1 つに対して使用される容量は 2 MB を少し超えます。たとえば、コンテンツホストの数が 10,000 個の場合、**/var/lib/qpidd/** に 20 GB のディスク容量が必要になります。
- **/var/lib/pulp/** ディレクトリーと **/var/lib/mongodb/** ディレクトリーには、高帯域幅で低レイテンシーのストレージの使用をお勧めします。Red Hat Satellite には I/O を大量に使用する操作が多数あるため、高レイテンシーで低帯域幅のストレージを使用すると、パフォーマンス低下の問題が発生します。インストールに、毎秒 60 - 80 メガバイトのスピードがあることを確認してください。**fio** ツールを使用すると、このデータが取得できます。**fio** ツールの詳細な使用方法は、Red Hat ナレッジベースのソリューション「Impact of Disk Speed on Satellite Operations」を参照してください。

ファイルシステムのガイドライン

- XFS ファイルシステムは、**ext4** では存在する inode の制限がないため、Red Hat Satellite 6 では XFS ファイルシステムを使用してください。Satellite Server は多くのシンボリックリンクを使用するため、**ext4** とデフォルトの数の inode を使用する場合は、システムで inode が足りなくなる可能性が高くなります。
- MongoDB は従来の I/O を使用してデータファイルにアクセスしないので、MongoDB では NFS を使用しないでください。また、NFS でデータファイルとジャーナルファイルの両方がホストされている場合にはパフォーマンスの問題が発生します。NFS を使用する必要がある場合は、**/etc/fstab** ファイルで **bg**、**nolock**、および **noatime** のオプションを使用してボリュームをマウントします。
- Pulp データストレージに NFS を使用しないでください。Pulp に NFS を使用すると、コンテンツの同期のパフォーマンスが低下します。
- 入出力レイテンシーが高すぎるため、GFS2 ファイルシステムは使用しないでください。

ログファイルのストレージ

ログファイルは、**/var/log/messages/**、**/var/log/httpd/**、および **/var/lib/foreman-proxy/openscap/content/** に書き込まれます。**logrotate** を使って、これらのファイルのサイズを管理できます。詳細は『**Red Hat Enterprise Linux 7 システム管理者のガイド**』の「ログローテーション」を参照してください。

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

NFS マウントを使用する場合の SELinux の考慮事項

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

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

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

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

重複パッケージ

同じパッケージが異なるリポジトリーで重複して存在する場合には、ディスク上に一度しか保存されません。そのため、重複するパッケージを別のリポジトリーに追加するときに必要な追加ストレージが少なくて済みます。ストレージの多くは、**/var/lib/mongodb/** および **/var/lib/pulp/** ディレクトリーにあります。これらのエンドポイントは手動で設定できません。ストレージの問題を回避するために、ストレージが **/var** ファイルシステムで利用可能であることを確認してください。

一時的なストレージ

/var/cache/pulp/ ディレクトリーは、同期中に、コンテンツを一時的に保存するために使用します。同期タスクがすべて完了したら、コンテンツは **/var/lib/pulp/** ディレクトリーに移動されます。

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

ソフトウェアコレクション

ソフトウェアコレクションは、**/opt/rh/** ディレクトリーと **/opt/theforeman/** ディレクトリーにインストールされます。

/opt ディレクトリーへのインストールには、root ユーザーによる書き込みパーミッションおよび実行パーミッションが必要です。

シンボリックリンク

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

同期された RHEL ISO

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

1.4. サポート対象オペレーティングシステム

オペレーティングシステムは、ディスク、ローカル ISO イメージ、キックスタート、または Red Hat がサポートする方法であれば他の方法でもインストールできます。Red Hat Satellite Server は、Satellite Server 6.7 のインストール時に入手可能な Red Hat Enterprise Linux 7 Server の最新バージョンでのみサポートされています。EUS または z-stream を含む以前の Red Hat Enterprise Linux バージョンはサポートされません。

Red Hat Satellite Server には、**@Base** パッケージグループが含む Red Hat Enterprise Linux インス

トールが必要です。他のパッケージセットの変更や、サーバーの運用に直接必要でないサードパーティーの構成やソフトウェアは含めないようにしてください。この制限は、ハード化や Red Hat 以外の他社のセキュリティーソフトウェアが該当します。インフラストラクチャーにこのようなソフトウェアが必要な場合は、Satellite Server が完全に機能することを最初に確認し、その後でシステムのバックアップを作成して、Red Hat 以外のソフトウェアを追加します。

新しくプロビジョニングされたシステムに Satellite Server をインストールします。

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

1.5. サポート対象ブラウザー

Satellite は、最新版の Firefox および Google Chrome ブラウザーをサポートします。

Satellite 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。

1.6. ポートとファイアウォールの要件

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

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

統合 Capsule

Satellite Server には Capsule が統合されており、Satellite Server に直接接続されたホストは、以下のセクションのコンテキストでは Satellite のクライアントになります。これには、Capsule Server が実行されているベースオペレーティングシステムが含まれます。

Capsule のクライアント

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

使用している設定に応じて、必要なポートは変わることがあります。

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

以下の表は、宛先ポートとネットワークトラフィックの方向を示しています。

表1.2 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 Server に Red Hat CDN へのアクセス権がある。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の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1.3 Satellite へのブラウザーベースユーザーインターフェース向けポート

ポート	プロトコル	サービス	用途
443	TCP	HTTPS	Satellite へのブラウザーベース UI アクセス
80	TCP	HTTP	Satellite に Web UI 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1.4 クライアントが Satellite と通信するためのポート

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
5646	TCP	AMQP	Capsule の Qpid ディスパッチルーターから Satellite の Qpid ディスパッチルーターへの通信
5647	TCP	AMQP	Satellite の Qpid ディスパッチルーターと通信する Katello エージェント
8000	TCP	HTTP	キックスタートテンプレートをホストにダウンロードする Anaconda、iPXE ファームウェアのダウンロード向け
8140	TCP	HTTPS	マスター接続に対する Puppet エージェント

ポート	プロトコル	サービス	用途
9090	TCP	HTTPS	プロビジョニング時の検出イメージや、リモート実行 (Rex) 設定の SSH キーをコピーするための Satellite Server との通信で使用するために、統合 Capsule で SCAP レポートを送信
7	TCP および UDP	ICMP	IP アドレスが解放されていることを確認するためのクライアント上の外部 DHCP から Satellite ネットワークと ICMP ECHO (オプション)
53	TCP および UDP	DNS	Satellite の統合 Capsule の DNS サービスへのクライアント DNS クエリー (オプション)
67	UDP	DHCP	Satellite の統合 Capsule ブロードキャストと、Satellite 統合 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 Satellite の統合 Capsule から PXE ブートイメージファイルをダウンロードするクライアント (オプション)
5000	TCP	HTTPS	Docker レジストリーのための Katello への接続 (オプション)

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

表1.5 Capsule に通信する Satellite 向けポート

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

表1.6 オプションのネットワークポート

ポート	プロトコル	サービス	用途
22	TCP	SSH	Remote Execution (Rex) および Ansible 向けの Satellite および Capsule からの通信
443	TCP	HTTPS	vCenter のコンピュートリソースに対する Satellite からの通信
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ーに対する Satellite からの通信
22, 16514	TCP	SSH、SSL/TLS	libvirt のコンピュートリソースに対する Satellite からの通信
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 からの通信
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 からの通信

1.7. 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。

以下の手順を使用して、Satellite のインストール先の Red Hat Enterprise Linux 7 システムでホストベースのファイアウォールを設定し、クライアントからの受信接続を有効にして、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。

手順

1. クライアントから Satellite の通信用のポートを開放するには、Satellite をインストールするベースオペレーティングシステムで以下のコマンドを入力します。

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

2. 変更を永続化します。

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

1.8. ファイアウォール設定の確認

この手順を使用して、ファイアウォール設定への変更を検証します。

手順

ファイアウォールの設定を検証するには、以下の手順を実行します。

1. 以下のコマンドを入力します。

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

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

1.9. DNS 解決の検証

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

手順

1. 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
```

2. 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場合には、コンテンツ管理、サブスクリプション管理、プロビジョニングなどのタスクに失敗します。

第2章 SATELLITE SERVER のインストール

オンラインネットワークから Satellite Server をインストールする場合は、Red Hat コンテンツ配信ネットワークから直接パッケージと更新を取得できます。

注記

Satellite Server に自己登録することはできません。

以下の手順を使用して、Satellite Server をインストールし、初期設定を実行して、サブスクリプションマニフェストをインポートします。サブスクリプションマニフェストに関する詳細は、『**コンテンツ管理ガイド**』の「**サブスクリプションの管理**」を参照してください。

Satellite 6 インストールスクリプトは Puppet をベースとするので、インストールスクリプトを複数回実行すると、手動での設定変更を上書きする可能性がある点に注意してください。これを回避し、今後どの変更を適用するか判断するには、インストールスクリプトの実行時に **--noop** の引数を使用します。この引数では、実際の変更は加えられません。今後変更される可能性のある内容は **/var/log/foreman-installer/satellite.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、不要な変更を元に戻すことができます。たとえば、foreman-installer ログで Filebucket に関する以下のようなエントリーが確認できます。

```
/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
```

2.1. RED HAT サブスクリプション管理への登録

Red Hat サブスクリプション管理にホストを登録すると、ユーザーが利用可能なサブスクリプションにホストを登録して、サブスクリプションのコンテンツを使用できるようになります。これには、Red Hat Enterprise Linux、Red Hat Software Collection (RHSCL)、Red Hat Satellite などのコンテンツが含まれます。

手順

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

  ```
  # subscription-manager register
  ```

  このコマンドを実行すると、以下のような出力が表示されます。

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

2.2. SATELLITE INFRASTRUCTURE サブスクリプションのアタッチ

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

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

サブスクリプションがシステムに割り当てられていない場合には、利用可能として分類されます。利用可能な Satellite サブスクリプションを見つけることができない場合は、Red Hat ナレッジベースソリューション「How do I figure out which subscriptions have been consumed by clients registered under Red Hat Subscription Manager?」を参照してスクリプトを実行し、サブスクリプションが別のシステムで使用されているかどうかを確認します。

手順

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

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

このコマンドを実行すると、以下のような出力が表示されます。

```
Subscription Name:   Red Hat Satellite Infrastructure Subscription
Provides:            Red Hat Satellite
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat CodeReady Linux Builder for x86_64
                     Red Hat Ansible Engine
                     Red Hat Enterprise Linux Load Balancer (for RHEL Server)
                     Red Hat
                     Red Hat Software Collections (for RHEL Server)
                     Red Hat Enterprise Linux Server
                     Red Hat Satellite Capsule
                     Red Hat Enterprise Linux for x86_64
                     Red Hat Enterprise Linux High Availability for x86_64
                     Red Hat Satellite
                     Red Hat Satellite 5 Managed DB
                     Red Hat Satellite 6
                     Red Hat Discovery
SKU:                 MCT3719
Contract:            11878983
Pool ID:             8a85f99968b92c3701694ee998cf03b8
Provides Management: No
Available:           1
Suggested:           1
Service Level:       Premium
Service Type:        L1-L3
Subscription Type:   Standard
Ends:                03/04/2020
System Type:         Physical
```

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

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

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

このコマンドを実行すると、以下のような出力が表示されます。

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

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

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

2.3. リポジトリーの設定

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

手順

1. すべてのリポジトリーを無効にします。

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

2. 以下のリポジトリーを有効にします。

```
# subscription-manager repos --enable=rhel-7-server-rpms \
--enable=rhel-7-server-satellite-6.7-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-ansible-2.8-rpms
```

注記

Red Hat Virtualization (RHV) がホストする仮想マシンとして、Satellite Server をインストールする場合は、**Red Hat Common** リポジトリーを有効にして、RHV ゲストエージェントとドライバーもインストールする必要があります。詳細は『**Virtual Machine Management Guide**』の「Installing the Guest Agents and Drivers on Red Hat Enterprise Linux」を参照してください。

3. メタデータを消去します。

```
# yum clean all
```

4. オプション: 必要なリポジトリーが有効になっていることを確認します。

```
# yum repolist enabled
```

2.4. SATELLITE SERVER パッケージのインストール

Satellite Server 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。

手順

1. すべてのパッケージを更新します。

```
# yum update
```

2. Satellite Server パッケージをインストールします。

```
# yum install satellite
```

2.5. CHRONYD とシステムクロックの同期

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

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

手順

1. **chrony** パッケージをインストールします。

```
# yum install chrony
```

2. **chronyd** サービスを起動して、有効にします。

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

2.6. ベース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

ベースオペレーティングシステムに **sos** パッケージをインストールし、Red Hat Enterprise Linux システムから設定および診断情報を取得できるようにします。このパッケージを使用すると、Red Hat テクニカルサポートへのサービスリクエストの起票時に必要な初期システム分析を提示できます。**sos** の使用方法に関する詳細は、カスタマーポータルのナレッジベースソリューション「What is a sosreport and how to create one in Red Hat Enterprise Linux 4.6 and later?」を参照してください。

手順

1. **sos** パッケージをインストールします。

```
# yum install sos
```

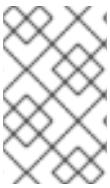
2.7. SATELLITE SERVER の設定

satellite-installer インストールスクリプトを使用して Satellite Server をインストールします。以下の手法から 1 つ選択します。

- 「Satellite の手動設定」.この手法では、1 つまたは複数のコマンドオプションを指定して、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。必要なオプションの設定に、必要に応じてスクリプトは何回でも実行することができます。

- 「応答ファイルを使用した Satellite の自動設定」.この手法では、インストールスクリプトの実行時に設定プロセスを自動化する応答ファイルを使用します。デフォルトの Satellite の応答ファイルは **/etc/foreman-installer/scenarios.d/satellite-answers.yaml** です。使用中の応答ファイルは、**/etc/foreman-installer/scenarios.d/satellite.yaml** 設定ファイルの **answer_file** ディレクティブで設定します。



注記

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

2.7.1. Satellite の手動設定

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。初期設定では、MongoDB および PostgreSQL データベースも同じサーバーにインストールします。

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** または **tmux** などのユーティリティーを使用してください。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。または、詳しくは **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。

手動設定に関する考慮事項

- **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、意味を持つ値を指定します。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、各サービスに関連するファイルが Puppet で管理されないようにします。他のサービスにカスタム設定を適用する方法は、付録A *Red Hat Satellite へのカスタム設定の適用* を参照してください。

手順

1. 使用する追加オプションを指定し、以下のコマンドを入力します。

```
-
```

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

このスクリプトは、進捗を表示し、**/var/log/foreman-installer/satellite.log** にログを記録します。

2.7.2. 応答ファイルを使用した Satellite の自動設定

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** スクリプトの初回実行後に、インストール向けの標準的なパラメーター値が入力されます。いつでも Satellite Server の設定は変更できます。

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。

手順

1. デフォルトの応答ファイル **/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. 応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、カスタム応答ファイルを参照する応答ファイルエントリーを編集します。

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

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

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

2.8. 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

以下の手順を使用して、サブスクリプションマニフェストを Satellite Server にインポートします。

前提条件

- カスタマーポータルから、サブスクリプションマニフェストファイルをエクスポートしておくこと。詳細は、『**Red Hat Subscription Management の使用**』ガイドの「マニフェストの使用」を参照してください。

手順

1. Satellite Web UI で、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** > **サブスクリプション** に移動して、**マニフェストの管理** をクリックします。
3. マニフェストの管理ウィンドウで、**参照**をクリックします。
4. サブスクリプションマニフェストファイルが保存されている場所に移動して、**表示** をクリックします。マニフェストの管理ウィンドウが自動的に終了しない場合は、**終了** をクリックしてサブスクリプションウィンドウに戻ります。

CLI をご利用の場合

1. サブスクリプションマニフェストファイルをクライアントから Satellite Server にコピーします。

```
$ scp ~/manifest_file.zip root@satellite.example.com:~/.
```

2. Satellite Server に **root** ユーザーとしてログインし、サブスクリプションマニフェストファイルをインポートします。

```
# hammer subscription upload \
--file ~/manifest_file.zip \
--organization "organization_name"
```

第3章 SATELLITE SERVER での追加設定の実行

3.1. SATELLITE SERVER での RED HAT INSIGHTS の使用

Red Hat Insights を使用すると、セキュリティー違反、パフォーマンスの低下、および安定性の消失に関連するシステムとダウンタイムを診断できます。ダッシュボードを使用して、安定性、セキュリティー、およびパフォーマンスの主要なリスクを素早く特定できます。また、カテゴリー別に分類したり、影響度および解決方法の詳細を表示したり、影響を受けたシステムを調べたりすることができます。

サブスクリプションマニフェストに Red Hat Insights のエンタイトルメントを追加する必要がない点に注意してください。Satellite および Red Hat Insights の詳細は、「Satellite で管理される Red Hat Enterprise Linux (RHEL) 上の Red Hat Insights」を参照してください。

Satellite Server を保守し、Satellite で発生する可能性のある問題を監視および診断する能力を向上させるには、Satellite Server に Red Hat Insights をインストールし、Satellite Server を Red Hat Insights に登録します。

insights-client のスケジューリング

Satellite に **insights-client.timer** を設定することで、デフォルトの **insights-client** 実行スケジュールを変更できる点に留意してください。詳細は、『**Red Hat Insights のクライアント設定ガイド**』の「insights-client スケジュールの変更」を参照してください。

手順

1. Satellite Server で Red Hat Insights をインストールするには、以下のコマンドを入力します。

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

2. Satellite Server を Red Hat Insights に登録するには、以下のコマンドを入力します。

```
# insights-client --register
```

3.2. SATELLITE TOOLS 6.7 リポジトリーの有効化

Satellite Tools 6.7 リポジトリーでは、Satellite Server に登録したクライアントに **katello-agent**、**katello-host-tools**、および **puppet** パッケージが提供されます。

手順

1. Satellite Web UI で、**コンテンツ** > **Red Hat リポジトリー** に移動します。
2. 検索フィールドを使用して **Satellite Tools 6.7（RHEL 7 Server 用）(RPM)**のリポジトリー名を入力します。
3. 利用可能なリポジトリーペインで、**Satellite Tools 6.7（RHEL 7 Server 用）(RPM)**をクリックして、リポジトリーセットを展開します。
   Satellite Tools 6.7の項目が表示されない場合は、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その項目が含まれないことが原因として考えられます。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ーを追加し、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. **x86_64** エントリーでは、**有効化** アイコンをクリックして、リポジトリーを有効にします。

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 6.7 リポジトリーを有効にします。Red Hat リポジトリーの有効後に、このリポジトリーの製品が自動的に作成されます。

CLI をご利用の場合

- **hammer repository-set enable** コマンドを使用して、Satellite Tools 6.7 リポジトリーを有効化します。

```
# hammer repository-set enable --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--basearch='x86_64' \
--name 'Red Hat Satellite Tools 6.7 (for RHEL 7 Server) (RPMs)'
```

3.3. SATELLITE TOOLS 6.7 リポジトリーの同期

本セクションを使用して、Red Hat コンテンツ配信ネットワーク(CDN)から Satellite に Satellite Tools 6.7 リポジトリーを同期します。このリポジトリーは、Satellite Server に登録したクライアントに **katello-agent**、**katello-host-tools**、および **puppet** パッケージを提供します。

手順

1. Satellite Web UI で、**コンテンツ** > **同期の状態** に移動します。
同期可能な製品リポジトリーのリストが表示されます。
2. **Red Hat Enterprise Linux Server**製品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. **Satellite Tools 6.7（RHEL 7 Server 用）RPMs x86_64**を選択します。
4. **今すぐ同期** をクリックします。

CLI をご利用の場合

- **hammer repository synchronize** コマンドを使用して、Satellite Tools 6.7 リポジトリーを同期します。

```
# hammer repository synchronize --organization "initial_organization_name" \
--product 'Red Hat Enterprise Linux Server' \
--name 'Red Hat Satellite Tools 6.7 for RHEL 7 Server RPMs x86_64' \
--async
```

3.4. HTTP プロキシーを使用した SATELLITE SERVER の設定

以下の手順を使用して、HTTP プロキシーで Satellite を設定します。

3.4.1. デフォルトの HTTP プロキシーの Satellite への追加

ネットワークで HTTP プロキシーを使用している場合は、Red Hat コンテンツ配信ネットワーク (CDN) または別のコンテンツソースへの要求送信に HTTP プロキシーを使用するように 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。

以下の手順では、Satellite のコンテンツダウンロード専用のプロキシーを設定します。

手順

1. Satellite Web UI で、**インフラストラクチャー** > **HTTP プロキシー** に移動します。
2. **新しい HTTP プロキシー** をクリックします。
3. **名前** フィールドで、HTTP プロキシーの名前を入力します。
4. **Url** フィールドで、**https://proxy.example.com:8080** の形式で HTTP プロキシーの URL を入力します。
5. オプション: 認証が必要な場合には、**Username** フィールドに認証に使用するユーザー名を入力します。
6. オプション: 認証が必要な場合には、**Password** フィールドに認証に使用するパスワードを入力します。
7. プロキシーへの接続をテストするには、**テスト接続** ボタンをクリックします。
8. **送信** をクリックします。
9. **管理** > **設定** に移動して、**コンテンツ** タブをクリックします。
10. 作成した HTTP プロキシーに **Default HTTP Proxy** 設定を指定します。

CLI をご利用の場合

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

```
# unset http_proxy
# unset https_proxy
# unset no_proxy
```

2. HTTP プロキシーエントリーを Satellite に追加します。

```
# hammer http-proxy create --name=myproxy \
--url http://myproxy.example.com:8080  \
--username=proxy_username \
--password=proxy_password
```

3. Satellite がデフォルトでこの HTTP プロキシーを使用するように設定します。

```
# hammer settings set --name=content_default_http_proxy --value=myproxy
```

3.4.2. Red Hat CDN に接続するための HTTP プロキシーの設定

Satellite が Red Hat CDN に接続し、リポジトリーを同期できることを確認します。

手順

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

ホスト名	ポート	プロトコル
subscription.rhsm.redhat.com	443	HTTPS
cdn.redhat.com	443	HTTPS
*.akamaiedge.net	443	HTTPS
cert-api.access.redhat.com (Red Hat Insights を使っている場合)	443	HTTPS
api.access.redhat.com (Red Hat Insights を使っている場合)	443	HTTPS

Satellite Server は、SSL を使用して Red Hat CDN との通信のセキュリティーを確保します。SSL インターセプションプロキシーを使用すると、この通信を干渉します。これらのホストはプロキシーでホワイトリスト化する必要があります。

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の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。

2. 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 = myproxy.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 =
```

3.4.3. カスタムポートでの Satellite へのアクセスを確保するように SELinux を設定する手順

SELinux を使用すると、Red Hat Satellite 6 と Red Hat Subscription Manager は、特定のポートにしかアクセスできます。HTTP キャッシュの場合には、TCP ポートは 8080、8118、8123、および 10001 - 10010 を使用できます。SELinux タイプが **http_cache_port_t** のポートを使用する場合には、以下の手順を実行してください。

手順

1. Satellite で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
[output truncated]
```

2. 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) を許可するよう設定します。

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

3.4.4. 全 Satellite HTTP 要求での HTTP プロキシーの使用

Satellite Server は、HTTP および HTTPS をブロックするファイアウォールの内側に設定する必要がある場合に、コンピュートリソースなどの外部システムとの通信に使用するプロキシーを設定してください。

プロビジョニングにコンピュートリソースを使用し、コンピュートリソースと、異なる HTTP プロキシーを併用する場合には、コンピュートリソースに設定したプロキシーではなく、Satellite 通信すべてに設定したプロキシーが優先されます。

手順

1. Satellite Web UI で、**管理** > **設定**に移動します。
2. **HTTP(S)プロキシー** 行で、隣接する **Value** 列を選択し、プロキシー URL を入力します。
3. チェックのアイコンをクリックして変更を保存します。

CLI をご利用の場合

- 以下のコマンドを入力します。

```
# hammer settings set --name=http_proxy --value=Proxy_URL
```

3.4.5. プロキシー化された要求を受信しないようにホストを除外する手順

Satellite HTTP または HTTPS 要求に HTTP プロキシーを使用する場合は、プロキシー経由で通信しないように、特定のホストを除外できます。

手順

1. Satellite Web UI で、**管理** > **設定**に移動します。
2. **HTTP(S) proxy except hosts**の行で、隣接する **Value** の列を選択して、プロキシー要求から除外する、1 つまたは複数のホストの名前を入力します。
3. チェックのアイコンをクリックして変更を保存します。

CLI をご利用の場合

- 以下のコマンドを入力します。

```
# hammer settings set --name=http_proxy_except_list --value=[hostname1.hostname2...]
```

3.4.6. HTTP プロキシーのリセット

現在の HTTP プロキシーの設定をリセットする場合には、**Default HTTP Proxy** 設定を解除します。

手順

1. **管理** > **設定** に移動して、**コンテンツ** タブをクリックします。
2. **Default HTTP Proxy** の設定を **no global default** に指定します。

CLI をご利用の場合

- **content_default_http_proxy** の設定を空の文字列に設定します。

```
# hammer settings set --name=content_default_http_proxy --value=""
```

3.5. 管理対象ホスト上での電源管理の有効化

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

前提条件

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

手順

- BMC を有効にするには、以下のコマンドを入力します。

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

3.6. SATELLITE SERVER での DNS、DHCP および TFTP の設定

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

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

代わりに外部の DNS、DHCP および TFTP サービスを使用するには、*4章 外部サービスでの Satellite Server の設定* を参照してください。

Multihomed DHCP の詳細の追加

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

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

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

このファイルがまだ存在しない場合は作成します。

2. 以下のコマンドを入力して、デーモンのリロードを実行します。

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

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

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

前提条件

- 以下の情報が利用可能であることを確認する。
  - 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。
- ネットワーク管理者に連絡して正しい設定が行われていることを確認する。

手順

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

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

プロンプトに表示される **satellite-installer** コマンドの進行状況を監視できます。**/var/log/foreman-installer/satellite.log** でログを表示できます。**/etc/foreman-installer/scenarios.d/satellite-answers.yaml** ファイルで、使用されている設定 (**initial_admin_password** パラメーターなど) を表示できます。

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

3.7. 管理対象外ネットワークに対する DNS、DHCP、および TFTP の無効化

TFTP、DHCP および DNS サービスを手動で管理する場合には、Satellite がオペレーティングシステム上でこれらのサービスを管理しないようにし、オーケストレーションを無効にして、DHCP および DNS バリデーションエラーを回避する必要があります。ただし、Satellite ではオペレーティングシステムのバックエンドサービスは削除されません。

手順

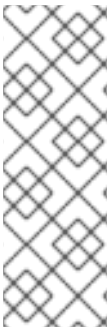
1. Satellite Server で以下のコマンドを入力します。

```
# satellite-installer --foreman-proxy-dhcp false \
--foreman-proxy-dns false \
--foreman-proxy-tftp false
```

2. Satellite Web UI で、**インフラストラクチャー** > **Capsule** に移動し、サブネットを選択します。
3. **Capsules** タブで、**DHCP Capsule**、**TFTP Capsule**、および **逆引き DNS Capsule** を選択します。
4. **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。
5. **DNS Capsule** フィールドの内容を消去します。
6. 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、想定の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、今後、ホストの削除に失敗しないように、既存の Satellite 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。

3.8. SATELLITE SERVER での送信メールの設定

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おく。以下に例を示します。

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

手順

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	**user@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	値例
配信方法	SMTP
SMTP アドレス	smtp.gmail.com
SMTP 認証	plain

名前	値例
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に関する詳細は、『**Red Hat Satellite の管理**』の「メール通知 の設定」を参照してください。

3.9. カスタムの SSL 証明書を使用した SATELLITE SERVER の設定

デフォルトでは、Red Hat Satellite 6 は自己署名の SSL 証明書を使用して、Satellite Server、外部の Capsule Server および全ホストの間で暗号化した通信ができるようにします。Satellite 自己署名の証明書を使用できない場合には、外部の証明局で署名した SSL 証明書を使用するように Satellite Server を設定できます。

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

1. 「Satellite Server 向けのカスタム SSL 証明書の作成」
2. 「カスタムの SSL 証明書の Satellite Server へのデプロイ」
3. 「ホストへの カスタム SSL 証明書のデプロイ」
4. Satellite Server に外部の Capsule Server を登録した場合には、カスタムの SSL 証明書を使用して設定する必要があります。詳細は、『Capsule Server の **インストール』の「カスタム SSL 証明書を使用した** Capsule Server の設定」を参照してください。

3.9.1. Satellite Server 向けのカスタム SSL 証明書の作成

この手順を使用して、Satellite Server 用にカスタムの SSL 証明書を作成します。Satellite Server 用のカスタムの SSL 証明書がある場合にはこの手順は省略してください。

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

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

手順

カスタムの SSL 証明書を作成するには、以下の手順を実行します。

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

```
# mkdir /root/satellite_cert
```

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

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

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

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

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

```
[ req_distinguished_name ] (1)
C  = Country Name (2 letter code)
ST = State or Province Name (full name)
L  = Locality Name (eg, city)
O  = Organization Name (eg, company)
OU = The division of your organization handling the certificate
CN = satellite.example.com (2)

[ v3_req ]
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth, codeSigning, emailProtection
subjectAltName = @alt_names

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

[ alt_names ]
DNS.1 = satellite.example.com (3)
```

(1) **[req_distinguished_name]** セクションに、貴社の組織の情報を入力します。

(2) 証明書のコモンネーム **CN** を、Satellite Server の完全修飾ドメイン名 (FQDN) と一致するように設定します。FQDN を確認するには、対象の Satellite Server で **hostname -f** コマンドを入力します。これは、**katello-certs-check** コマンドが証明書を正しく検証することを確認するために必要です。

(3) サブジェクトの別名 (SAN: Subject Alternative Name) **DNS.1** を、お使いのサーバーの完全修飾ドメイン名 (FQDN) に一致する用に設定します。

4. 証明書署名要求 (CSR) を作成します。

```
# openssl req -new \
-key /root/satellite_cert/satellite_cert_key.pem \ (1)
-config /root/satellite_cert/openssl.cnf \ (2)
-out /root/satellite_cert/satellite_cert_csr.pem (3)
```

(1) 秘密鍵へのパス

(2) 設定ファイルへのパス

(3) 生成する CSRへのパス

5. 証明局に証明書署名要求を送信します。同じ証明局を使用して Satellite Server と Capsule Server の証明書を署名する必要がある。

要求を送信する場合は、証明書の有効期限を指定してください。証明書要求を送信する方法は異なるため、推奨の方法について認証局にお問い合わせください。要求への応答で、認証局バンドルと署名済み証明書を別々のファイルで受け取ることになります。

3.9.2. カスタムの SSL 証明書の Satellite Server へのデプロイ

この手順を使用して、Satellite Server が、認証局で署名されたカスタムの SSL 署名書を使用するように設定します。**katello-certs-check** コマンドは、入力した証明書ファイルを検証して、Satellite Server にカスタムの SSL 証明書をデプロイするのに必要なコマンドを返します。

手順

Satellite Server にカスタムの証明書をデプロイするには、以下の手順を実行します。

1. カスタムの SSL 証明書入力ファイルを検証します。**katello-certs-check** コマンドが正しく実行されるには、証明書のコモンネーム (CN) が Satellite Server の FQDN と一致する必要があることに注意してください。

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

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

このコマンドに成功すると、2 つの **satellite-installer** コマンドが返されます。1 つは、Satellite Server に証明書をデプロイするのに使用する必要があります。

katello-certs-check の出力例

```
Validation succeeded.

To install the Red Hat Satellite Server with the custom certificates, run:

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

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

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

2. 要件に合わせて **katello-certs-check** コマンドの出力から、**satellite-installer** コマンドを入力し、カスタムの SSL 証明書で新しい Satellite をインストールするか、現在実行中の Satellite の証明書を更新します。

実行するコマンドが不明な場合には、**/etc/foreman-installer/scenarios.d/.installed** が存在するかをチェックし、Satellite がインストールされていることが確認できます。ファイルが存在する場合には、2 番目の **satellite-installer** コマンドを実行すると証明書が更新されます。

重要

証明書のデプロイ後に、証明書のアーカイブファイルを削除しないでください。Satellite Server のアップグレード時などに必要です。

3. Satellite Server にネットワークでアクセスできるコンピューターで、この URL (**https://satellite.example.com**) に移動します。

4. ブラウザーで、証明書の詳細を表示して、デプロイした証明書を確認します。

3.9.3. ホストへの カスタム SSL 証明書のデプロイ

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

手順

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

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

3.10. SATELLITE での外部データベースの使用

Red Hat Satellite のインストールプロセスの一部として、**satellite-installer** コマンドは MongoDB および PostgreSQL のデータベースを Satellite と同じサーバー上にインストールします。Satellite のデプロイメントによっては、デフォルトのローカルにあるデータベースの代わりに外部データベースを使用すると、サーバーの負荷を軽減される場合があります。外部データベースに MongoDB と PostgreSQL のどちらのデータベースが使用できるか (または両方使用できるか) については、要件によって異なります。

Red Hat では、外部データベースのメンテナンスのサポートやそのためのツールは提供していません。これにはバックアップ、アップグレード、データベースのチューニングが含まれます。外部データベースをサポートし、管理する自社のデータベース管理者が必要です。

外部データベースとして MongoDB を使用する際の注意点 および 外部データベースとして PostgreSQL を使用する際の注意点 を参照して、Satellite デプロイメントに外部データベースを使用するかどうかを決定します。

Satellite 用に外部データベースを作成して使用するには、以下の手順を実行します。

1. 「外部データベース用のホストの準備」: 外部データベースをホストするように Red Hat Enterprise Linux 7 サーバーを準備します。

2. 「MongoDB のインストール」.**pulp_database** を所有する **pulp** ユーザーで MongoDB を準備します。

3. 「PostgreSQL のインストール」.Satellite および Candlepin のデータベースおよびそれらを所有する専用ユーザーで PostgreSQL を準備します。

4. 「外部データベースを使用するための Satellite の設定」.新規データベースを参照するように **satellite-installer** のパラメーターを編集し、**satellite-installer** を実行します。

3.10.1. 外部データベースとして MongoDB を使用する際の注意点

Pulp は MongoDB データベースを使用します。MongoDB を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判別してください。Satellite は MongoDB バージョン 3.4 をサポートしています。

外部 MongoDB の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- Satellite 操作にマイナスの影響をもたらすことなく MongoDB サーバーのシステムを調整する柔軟性が得られます。

外部 MongoDB のマイナス点

- デプロイメントの複雑性が増し、問題解決がより困難になります。
- 外部 MongoDB サーバーの場合は、パッチおよびメンテナンス対象に新たなシステムが加わることになります。
- Satellite または Mongo データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite と外部データベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出る可能性があります。

FIPS 関連の制限

- FIPS モードの Satellite で外部 MongoDB を使用することはできません。

3.10.2. 外部データベースとして PostgreSQL を使用する際の注意点

Foreman、Katello、および Candlepin は PostgreSQL データベースを使用します。PostgreSQL を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判別してください。Satellite は PostgreSQL バージョン 9.2 をサポートします。

外部 PostgreSQL の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- PostgreSQL データベースで **shared_buffers** を高い値に設定しても、Satellite 上の他のサービスの妨げるリスクがありません。
- Satellite 操作にマイナスの影響をもたらすことなく PostgreSQL サーバーのシステムを調整する柔軟性が得られます。

外部 PostgreSQL のマイナス点

- デプロイメントの複雑性が増し、問題解決がより困難になります。
- 外部 PostgreSQL サーバーの場合は、パッチおよびメンテナンス対象に新たなシステムが加わることになります。

- Satellite または PostgreSQL データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite Server とデータベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出ます。

お使いの Satellite 上の PostgreSQL データベースが原因でパフォーマンスの低下が生じている可能性がある場合は、「Satellite 6: How to enable postgres query logging to detect slow running queries 」を参照して時間のかかっているクエリーがあるかどうか判定します。1 秒以上かかるクエリーがある場合は、通常、大規模インストールのパフォーマンスが原因であることが多く、外部データベースに移行しても問題解決が期待できません。時間のかかっているクエリーがある場合は、Red Hat サポートチームまでお問い合わせください。

3.10.3. 外部データベース用のホストの準備

新しくプロビジョニングされたシステムに最新の Red Hat Enterprise Linux 7 サーバーをインストールして、外部データベースをホストします。

Red Hat Software Collections および Red Hat Enterprise Linux のサブスクリプションでは、外部データベースと Satellite を併用する場合に、正しいサービスレベルアグリーメントが提供されません。外部データベースに使用するベースオペレーティングシステムにも、Satellite サブスクリプションをアタッチする必要があります。

前提条件

- Red Hat Enterprise Linux 7 サーバーは Satellite の ストレージ要件 を満たしている必要があります。

手順

1. 「Satellite Infrastructure サブスクリプションのアタッチ」の 手順に従い、サーバーに Satellite サブスクリプションをアタッチします。
2. すべてのリポジトリーを無効にし、以下のリポジトリーのみを有効にします。

```
# subscription-manager repos --disable '*'
# subscription-manager repos --enable=rhel-server-rhscl-7-rpms \
--enable=rhel-7-server-rpms
```

3.10.4. MongoDB のインストール

インストール可能な MongoDB は、内部データベースのインストール中に **satellite-installer** ツールでインストールされたものと同じバージョンの MongoDB のみになります。MongoDB はサポート対象のバージョンであれば、Red Hat Software Collections (RHSCL) リポジトリーからまたは外部ソースからインストールすることが可能です。Satellite は MongoDB バージョン 3.4 をサポートしています。

手順

1. MongoDB をインストールするには、以下のコマンドを入力します。

```
# yum install rh-mongodb34 rh-mongodb34-syspaths
```

2. **rh-mongodb34** サービスを起動して有効にします。

```
# systemctl start rh-mongodb34-mongod
# systemctl enable rh-mongodb34-mongod
```

3. **pulp_database** データベース用に、MongoDB に Pulp ユーザーを作成します。

```
# mongo pulp_database \
--eval "db.createUser({user:'pulp',pwd:'pulp_password',roles:[{role:'dbOwner',
db:'pulp_database'},{ role: 'readWrite', db: 'pulp_database'}]})"
```

4. **/etc/opt/rh/rh-mongodb34/mongod.conf** ファイルでバインド IP を指定します。

```
bindIp: your_mongodb_server_bind_IP,::1
```

5. **/etc/opt/rh/rh-mongodb34/mongod.conf** ファイルを編集して **security** セクションの認証を有効にします。

```
security:
  authorization: enabled
```

6. **rh-mongodb34-mongod** サービスを再起動します。

```
# systemctl restart rh-mongodb34-mongod
```

7. MongoDB にポート 27017 を開きます。

```
# firewall-cmd --add-port=27017/tcp
# firewall-cmd --runtime-to-permanent
```

8. Satellite Server から、データベースにアクセスできることをテストします。接続が成功すると、コマンドから **1** が返ります。

```
# scl enable rh-mongodb34 " mongo --host mongo.example.com \
-u pulp -p pulp_password --port 27017 --eval 'ping:1' pulp_database"
```

3.10.5. PostgreSQL のインストール

インストール可能な PostgreSQL は、内部データベースのインストール中に **satellite-installer** ツールでインストールされたものと同じバージョンの PostgreSQL のみになります。PostgreSQL はサポート対象のバージョンであれば、Red Hat Enteprise Linux Server 7 リポジトリーからまたは外部ソースからインストールすることが可能です。Satellite は PostgreSQL バージョン 9.2 をサポートします。

手順

1. PostgreSQL をインストールするには、以下のコマンドを入力します。

```
# yum install postgresql-server
```

2. PostgreSQL サービスを初期化して起動し、有効にするには、以下のコマンドを実行します。

```
# postgresql-setup initdb
# systemctl start postgresql
# systemctl enable postgresql
```

3. **/var/lib/pgsql/data/postgresql.conf** ファイルを編集します。

```
# vi /var/lib/pgsql/data/postgresql.conf
```

4. **#** を削除して、着信接続をリッスンするようにします。

```
listen_addresses = '*'
```

5. **/var/lib/pgsql/data/pg_hba.conf** ファイルを編集します。

```
# vi /var/lib/pgsql/data/pg_hba.conf
```

6. 以下の行をファイルに追加します。

```
host  all   all   Satellite_ip/24   md5
```

7. PostgreSQL サービスを再起動して、変更を適用します。

```
# systemctl restart postgresql
```

8. 外部 PostgreSQL サーバーで **postgresql** ポートを開きます。

```
# firewall-cmd --add-service=postgresql
# firewall-cmd --runtime-to-permanent
```

9. **postgres** ユーザーに切り替え、PostgreSQL クライアントを起動します。

```
$ su - postgres -c psql
```

10. Satellite と Candlepin 用にそれぞれ、データベース、および専用ロールを作成します。

```
CREATE USER "foreman" WITH PASSWORD 'Foreman_Password';
CREATE USER "candlepin" WITH PASSWORD 'Candlepin_Password';
CREATE DATABASE foreman OWNER foreman;
CREATE DATABASE candlepin OWNER candlepin;
```

11. **postgres** ユーザーをログアウトします。

```
# \q
```

12. Satellite Server から、データベースにアクセスできることをテストします。接続に成功した場合には、コマンドは **1** を返します。

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com  -p 5432 -U
foreman -d foreman -c "SELECT 1 as ping"
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U
candlepin -d candlepin -c "SELECT 1 as ping"
```

3.10.6. 外部データベースを使用するための Satellite の設定

satellite-installer コマンドを使用して Satellite が外部の MongoDB と PostgreSQL データベースに接続するように設定します。

前提条件

- Red Hat Enterprise Linux サーバーに MongoDB および PostgreSQL データベースをインストールおよび設定していること。

手順

1. Satellite の外部データベースを設定するには以下のコマンドを入力します。

```
satellite-installer --scenario satellite \
  --foreman-db-host postgres.example.com \
  --foreman-db-password Foreman_Password \
  --foreman-db-database foreman \
  --katello-candlepin-db-host postgres.example.com \
  --katello-candlepin-db-name candlepin \
  --katello-candlepin-db-password Candlepin_Password \
  --katello-candlepin-manage-db false \
  --katello-pulp-db-username pulp \
  --katello-pulp-db-password pulp_password \
  --katello-pulp-db-seeds mongo.example.com:27017 \
  --katello-pulp-db-name pulp_database
```

2. データベースのステータスを確認します。

- PostgreSQL の場合は、以下のコマンドを実行します。

```
# satellite-maintain service status --only postgresql
```

- MongoDB の場合は、以下のコマンドを実行します。

```
# satellite-maintain service status --only rh-mongodb34-mongod
```

3.11. MONGOD へのアクセスの制限

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

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

手順

1. 以下のコマンドを入力して、ファイアウォール構成を更新します。

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

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

2. 変更を永続化します。

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

3.12. 事前定義済みプロファイルを使用した SATELLITE SERVER の調整

Satellite のデプロイメントに 5000 台を超えるホストが含まれる場合には、事前定義済みのチューニングプロファイルを使用して Satellite Server のパフォーマンスを向上できます。

Satellite が管理するホストの数と利用可能なハードウェアリソースに応じて、プロファイルの 1 つを選択できます。tuning プロファイルは、**/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes** ディレクトリーにあります。

default

管理対象ホスト数: 0-5000
RAM: 20G

CPU コア数: 4

medium

管理対象ホスト数: 5001-10000
RAM: 32G

CPU コア数: 8

large

管理対象ホスト数: 10001-20000
RAM: 64G

CPU コア数: 16

extra-large

管理対象ホスト数: 20001-60000
RAM: 128G

CPU コア数: 32

extra-extra-large

管理対象ホスト数: 60000+
RAM: 256G

CPU コア数: 48+

手順

- Satellite デプロイメントのチューニングプロファイルを設定するには、**--tuning** オプションを指定して **satellite-installer** コマンドを入力します。たとえば、medium tuning プロファイル設定を適用するには、以下のコマンドを入力します。

```
# satellite-installer --tuning medium
```

第4章 外部サービスでの SATELLITE SERVER の設定

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

4.1. 外部 DNS を使用した SATELLITE SERVER の設定

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

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

前提条件

- 外部 DNS サーバーが構成されている必要がある。

手順

1. **bind-utils** パッケージをインストールしておく。

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

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

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

3. 所有者、パーミッション、SELinux コンテキストを設定します。

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

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

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

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

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

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

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

7. foreman-proxy サービスを再起動します。

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

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

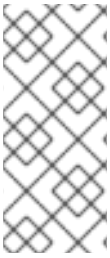
4.2. 外部 DHCP を使用した SATELLITE SERVER の設定

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

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

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

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



注記

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

手順

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

```
# yum install dhcp bind
```

2. セキュリティートークンを生成します。

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

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

3. キーからシークレットハッシュをコピーします。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

11. DHCP サービスを起動します。

```
# systemctl start dhcpd
```

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

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

13. NFS を使用してエクスポートする DHCP 設定ファイルとリースファイルのディレクトリーを作成します。

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

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

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

15. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

16. **/etc/exports** に以下の行があることを確認します。

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

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

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

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

17. NFS サーバーをリロードします。

```
# exportfs -rva
```

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

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

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

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

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

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

前提条件

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

手順

1. **nfs-utils** ユーティリティーをインストールします。

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

2. NFS 用の DHCP ディレクトリーを作成します。

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

3. ファイルの所有者を変更します。

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

4. NFS サーバーとの通信とリモートプロシージャコール (RPC: Remote Procedure Call) 通信パスを検証します。

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

5. **/etc/fstab** ファイルに以下の行を追加します。

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

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

6. **/etc/fstab** でファイルシステムをマウントします。

```
# mount -a
```

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

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

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

```
# satellite-installer --foreman-proxy-dhcp=true \
--foreman-proxy-dhcp-provider=remote_isc \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-config /mnt/nfs/etc/dhcp/dhcpd.conf \
--foreman-proxy-plugin-dhcp-remote-isc-dhcp-leases /mnt/nfs/var/lib/dhcpd/dhcpd.leases \
--foreman-proxy-plugin-dhcp-remote-isc-key-name=omapi_key \
--foreman-proxy-plugin-dhcp-remote-isc-key-
secret=jNSE5YI3H1A8Oj/tkV4...A2ZOHb6zv315CkNAY7DMYYCj48Umw== \
--foreman-proxy-plugin-dhcp-remote-isc-omapi-port=7911 \
--enable-foreman-proxy-plugin-dhcp-remote-isc \
--foreman-proxy-dhcp-server=DHCP_Server_FQDN
```

9. **foreman-proxy** サービスを再起動します。

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

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

4.3. 外部 TFTP での SATELLITE SERVER の設定

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

手順

1. NFS 用に TFTP ディレクトリーを作成します。

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

2. **/etc/fstab** ファイルで以下の行を追加します。

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

3. **/etc/fstab** のファイルシステムをマウントします。

```
# mount -a
```

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

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

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

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

6. Satellite Server Web UI にログインします。

7. **インフラストラクチャー** > **Capsules** に移動し、Satellite Server の場所を特定して、 **Actions** コラムの一覧から、**Refresh** を選択します。

8. TFTP サービスに適切なサブネットとドメインを関連付けます。

4.4. 外部 IDM DNS を使用した SATELLITE SERVER の設定

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

Satellite Server は、 Red Hat Identity Management (IdM) サーバーを使って DNS サービスを提供するように設定できます。Red Hat Identity Management の詳細は、『Linux ドメイン ID、認証、およびポリシーガイド』を参照してください。

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

- 「GSS-TSIG 認証を使用した動的 DNS 更新の設定」
- 「TSIG 認証を使用した動的 DNS 更新の設定」

内部 DNS サービスに戻すには、次の手順を使用します。

- 「内部 DNS サービス使用への復元」

注記

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

4.4.1. GSS-TSIG 認証を使用した動的 DNS 更新の設定

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

トをインストールする必要があります。

前提条件

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

手順

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

IdM サーバーでの Kerberos プリンシパルの作成

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

```
# kinit idm_user
```

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

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

IdM クライアントのインストールおよび設定

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

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

   - Capsule Server で以下のコマンドを入力します。

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

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

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

3. Kerberos チケットを取得します。

```
# kinit admin
```

4. 既存の **keytab** を削除します。

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

5. このシステムの **keytab** を取得します。

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

注記

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

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

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

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

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

IdM Web UI での DNS ゾーンの設定

1. 管理するゾーンを作成して、設定します。
   a. **Network Services (ネットワークサービス)** > **DNS** > **DNS Zones (DNS ゾーン)** に移動します。
   b. **追加** を選択し、ゾーン名を入力します。(例: **example.com**)
   c. **Add and Edit** をクリックします。
   d. 設定タブをクリックして **BIND アップデートポリシー** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

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

   e. **Dynamic update** を **True** に設定します。
   f. **Allow PTR sync** を有効にします。
   g. **Save** をクリックして、変更を保存します。
2. 逆引きゾーンを作成して設定します。

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

b. **Add** をクリックします。

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

d. **Add and Edit** をクリックします。

e. **設定** タブの **BIND アップデートポリシー** ボックスで、以下のようにセミコロン区切りのエントリーを追加します。

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

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

g. **Save** をクリックして、変更を保存します。

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

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

- Satellite で以下のコマンドを入力します。

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

- Capsule で、以下のコマンドを実行します。

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

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

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

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

Satellite Web UI での設定更新

1. **インフラストラクチャー** > **Capsules** に移動し、Satellite Server の場所を特定して、 **Actions** コラムの一覧から、**Refresh** を選択します。
2. ドメインを設定します。
   a. **インフラストラクチャー** > **ドメイン** に移動し、ドメイン名を選択します。
   b. **ドメイン** タブで、**DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
3. サブネットを設定します。
   a. **インフラストラクチャー** > **サブネット** に移動し、サブネット名を選択します。
   b. **サブネット** タブで、**IPAM** を **None** に設定します。
   c. **ドメイン** タブで、IdM サーバーを使用して管理するドメインを選択します。
   d. **Capsules** タブで、**Reverse DNS Capsule** が、サブネットが接続されている Capsule に設定されていることを確認します。
   e. **送信** をクリックして変更を保存します。

4.4.2. TSIG 認証を使用した動的 DNS 更新の設定

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

前提条件

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

手順

TSIG 認証で動的 DNS 更新を設定するには、以下の手順を実行します。

IdM サーバーの DNS ゾーンに対する外部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。

```
include "/etc/rndc.key"; controls { inet IdM_Server_IP_Address port 953 allow {
Satellite_IP_Address; } keys { "rndc-key"; }; };
```

2. **named** サービスをリロードして、変更を有効にします。

```
# systemctl reload named
```

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

   a. **BIND update policy (BIND 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** を **True** に設定します。

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

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

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

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

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

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

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

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

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

IdM サーバーの DNS ゾーンに対する外部アップデートのテスト

1. **bind-utils** ユーティリティーをインストールします。

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

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

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

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

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

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

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

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

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

6. 正常に解決されたら、テスト DNS エントリーを削除します。

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

7. DNS エントリーが削除されたことを確認します。

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

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

4.4.3. 内部 DNS サービス使用への復元

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

手順

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

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

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

```
# satellite-installer
```

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

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

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

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

Satellite Web UI での設定更新

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

付録A 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、設定する場合には、**--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して、DNS および DHCP 設定ファイルが Puppet で管理されないように指定してください。これらのフラグがインストーラーの初回実行時に指定されていない場合には、アップグレードの目的で再実行する場合など、インストーラーを再実行すると、手動で変更した内容がすべて上書きされます。変更が上書きされた場合には、復元の手順を実行して手動の変更を復元する必要があります。詳細は、付録B *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設定するには、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** 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する場合は、Red Hat サポートにお問い合わせください。

付録B 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。

手順

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があるか、ファイル間で比較できます。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。以下に例を示します。

```
# 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。