



## Red Hat Satellite 6.10

### Red Hat Satellite のアップグレードおよび更新

Red Hat Satellite Server および Capsule Server のアップグレードおよび更新



# Red Hat Satellite 6.10 Red Hat Satellite のアップグレードおよび更新

---

Red Hat Satellite Server および Capsule Server のアップグレードおよび更新

Red Hat Satellite Documentation Team

[satellite-doc-list@redhat.com](mailto:satellite-doc-list@redhat.com)

## 法律上の通知

Copyright © 2023 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, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

本ガイドでは、Red Hat Satellite Server、Capsule Server、およびホストのアップグレードおよび更新について説明します。

---

## 目次

<b>第1章 アップグレードの概要</b> .....	<b>3</b>
1.1. 前提条件	3
1.2. アップグレードパス	4
1.3. アップグレードの進捗の追跡	5
1.4. SATELLITE とは別の CAPSULE のアップグレード	5
<b>第2章 SATELLITE SERVER のクローン</b> .....	<b>7</b>
2.1. クローン作成プロセスの概要	7
2.2. 前提条件	7
2.3. PULP データの考慮事項	8
2.4. SATELLITE SERVER のクローン	8
<b>第3章 RED HAT SATELLITE のアップグレード</b> .....	<b>12</b>
3.1. SATELLITE SERVER のアップグレード	12
3.2. 新しいリポジトリーの同期	25
3.3. CAPSULE SERVER のアップグレード	26
3.4. SATELLITE WEB UI での CAPSULE SERVER の同期	30
3.5. SATELLITE クライアントのアップグレード	30
<b>第4章 アップグレード後のタスク</b> .....	<b>33</b>
4.1. DISCOVERY のアップグレード	33
4.2. VIRT-WHO のアップグレード	35
4.3. 以前のバージョンの SATELLITE TOOLS リポジトリーの削除	35
4.4. POSTGRESQL 領域の確保	35
4.5. テンプレート、パラメーター、ルックアップキーおよび値の更新	36
4.6. 事前定義済みプロファイルを使用した SATELLITE SERVER の調整	37
4.7. 外部 CAPSULE SERVER での PUPPET 環境の検証	38
<b>第5章 SATELLITE SERVER、CAPSULE SERVER、およびコンテンツホストの更新</b> .....	<b>40</b>
5.1. SATELLITE SERVER の更新	40
5.2. 接続されていない SATELLITE SERVER の更新	41
5.3. CAPSULE SERVER の更新	45



# 第1章 アップグレードの概要

本章では、Red Hat Satellite 6.10 の前提条件、および利用可能なアップグレードパスを説明します。現在の Red Hat Satellite 6 インストールをアップグレードする前にお読みください。

本ガイドでは、更新、アップグレード、マイグレーション (移行) を以下の意味で使用します。

## アップグレード

y-stream を基準にして、Satellite Server および Capsule Server のインストールを次のリリースに上げるプロセスです (たとえば Satellite 6.9 から Satellite 6.10)。

## 更新

z-stream を基準にして、Satellite Server および Capsule Server のインストールを次のリリースに上げるプロセスです (たとえば Satellite 6.10.0 から Satellite 6.10.1)。

## マイグレーション (移行)

既存の Satellite インストールを、別の Red Hat Enterprise Linux サーバーに移行するプロセスです。

Red Hat カスタマーポータル [の Red Hat Satellite Upgrade Helper](#) では、対話式のアップグレード手順がご利用になれます。このアプリケーションは、現在のバージョン番号に適した手順を提供します。アップグレードパスに固有の手順や、既知の問題を回避する手順を確認できます。詳細は、カスタマーポータル [の Satellite Upgrade Helper](#) を参照してください。

Capsule は、Satellite とは別にアップグレードできます。詳しくは、[「Satellite とは別の Capsule のアップグレード」](#) をご覧ください。

## 1.1. 前提条件

Satellite 6.10 へのアップグレードは、Satellite インフラストラクチャー全体に影響します。アップグレード前に以下を完了してください。

- [Release Notes](#) を確認する。
- このガイドで、アップグレードプロセスとその影響について確認する。
- アップグレードパスの計画を立てます。詳細は、[「アップグレードパス」](#) を参照してください。
- 必要とされるダウンタイムを計画します。Satellite サービスはアップグレード時は停止します。アップグレードプロセスの期間は、ハードウェアの設定、ネットワークの速度、サーバーに保存されているデータ量により異なる可能性があります。Satellite のアップグレードには約 1 ~ 2 時間かかる。

Capsule のアップグレードには約 10 ~ 30 分かかります。

ただし、6.9 から 6.10 にアップグレードすると、Pulp コンテンツも移行します。この手順にはかなり時間がかかる場合があります。Pulp 移行およびアップグレードプロセスの準備に関する詳細は、[「Satellite Server のアップグレード」](#) を参照してください。

- サーバーに十分なストレージ容量があることを確認します。詳細は、[オンラインネットワークからの Satellite Server のインストール](#) の [インストールのための環境準備](#) と [Capsule Server のインストール](#) の [インストールのための環境準備](#) を参照してください。

- Satellite Server およびすべての Capsule Server をバックアップします。詳細は、[Red Hat Satellite 6.9 の管理の Satellite Server および Capsule Server のバックアップ](#) を参照してください。
- Satellite のバージョンごとに API コマンドが異なる場合がありますので、使用しているスクリプトに Satellite API コマンドが含まれる場合は、更新の計画を立てます。



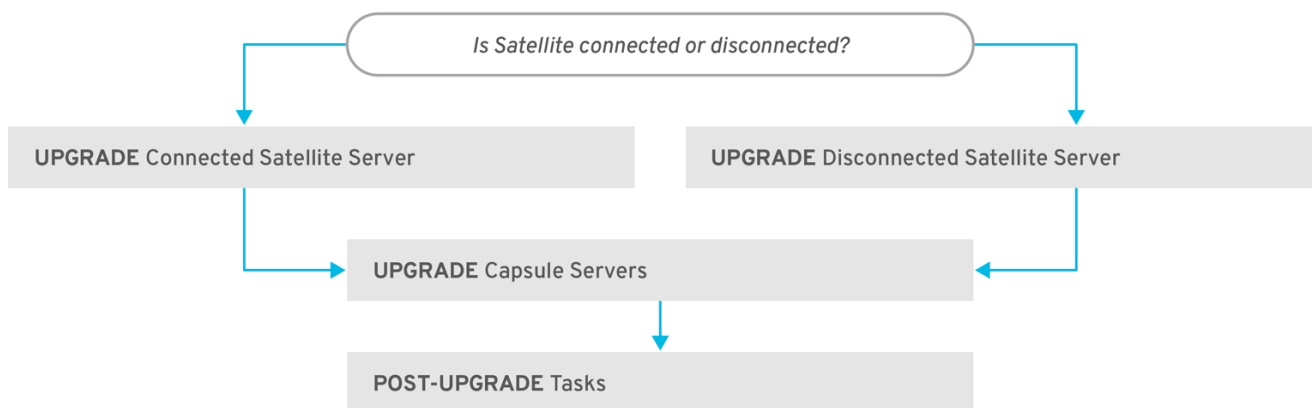
### 警告

設定ファイルを手動で、または Hiera などのツールを使用して、カスタマイズした場合には、このカスタマイズした内容は、アップグレード時または更新時にインストールスクリプトを実行すると上書きされます。satellite-installer スクリプトで **--noop** オプションを使用すると、変更をテストできます。詳細は、Red Hat ナレッジベースソリューションの [How to use the noop option to check for changes in Satellite config files during an upgrade](#) を参照してください。

## 1.2. アップグレードパス

Red Hat Satellite 6.9 から Red Hat Satellite 6.10 にアップグレードできます。以前のバージョンの Satellite Server と Capsule Server は、先に Satellite 6.9 にアップグレードする必要があります。詳細は、Satellite 6.9 の [Upgrading and Updating Red Hat Satellite](#) を参照してください。

図1.1 Satellite 6.10 アップグレードパスの概要



SATELLITE\_461763\_0718



### 警告

ベータから GA バージョンへのアップグレードはサポートされていません。

以下は、Satellite 6.10 にアップグレードする手順の概要です。



1. 既存の Satellite Server をクローンします。詳細は、[2章Satellite Server のクローン](#) を参照してください。
2. Satellite Server と全 Capsule Server を Satellite 6.10 にアップグレードします。詳細は、「[Satellite Server のアップグレード](#)」を参照してください。
3. すべての Satellite クライアントで、Satellite Tools 6.10 にアップグレードします。詳細は、「[Satellite クライアントのアップグレード](#)」を参照してください。

## Satellite を将来のバージョンにアップグレードする際の留意事項

Satellite Server での Pulp 3 へのアップグレードが含まれる、Satellite 6.10 へのアップグレードを開始する前に、Pulp コンテンツの事前移行を完了することを強く推奨します。6.9 で移行を実行しないと、すべて 6.10 へのアップグレード中実行されます。コンテンツの設定が小さい場合や、長いダウンタイムが許容できる場合は、事前移行なしに次に進むことができます。

Capsule の場合は、アップグレードするのではなく、新しい 6.10 Capsule のデプロイを選択できます。

Satellite 6.11 以降のアップグレードでは、Satellite Server および Capsule で、オペレーティングシステムを RHEL 7 から RHEL 8 にアップグレードする必要があります。オペレーティングシステムはインプレースまたはクローン作成プロセスを通じてアップグレードできます。後者には、すべてのデータ、設定、および同期コンテンツの移行が含まれます。



### 注記

Pulp 2 から Pulp 3 へのアップグレードは回避し、Pulp 3 の変更が原因で新しい Satellite 6.10 インフラストラクチャーをデプロイする場合は、Satellite 6.11 を RHEL 8 で直接デプロイするまで待つことができます。

## 1.3. アップグレードの進捗の追跡

アップグレード時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。また、**screen** の man ページでも、詳細を確認できます。

アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、`/var/log/foreman-installer/satellite.log` のログで、プロセスが完全に終了したかどうかを確認できます。

## 1.4. SATELLITE とは別の CAPSULE のアップグレード

Satellite はバージョン 6.10 にアップグレードし、Capsule は、アップグレードする容量が確保されるまでバージョン 6.9 に保つことができます。

これまで動作していた機能はすべて 6.9 Capsules で動作します。しかし、6.10 リリースで追加された機能は Capsule を 6.10 にアップグレードするまで動作しません。

Satellite のアップグレード後に Capsule をアップグレードすると、以下のようなシナリオ例で役に立ちます。

1. 長期にわたる停止期間の発生を避け、停止期間を複数回に分けて短くする場合。
2. 組織内の Capsule が複数のチームで管理されており、別のロケーションに配置されている場合。

3. 負荷分散設定を使用する場合に、負荷分散されている Capsule をアップグレードして、残りの負荷分散型 Capsule を1つ前のバージョンにしておくことができます。こうすることで、サービスを停止せずに全 Capsule を順番にアップグレードできます。

## 第2章 SATELLITE SERVER のクローン

Satellite Server のアップグレード時に、アップグレード中にデータが損失されないように、オプションで Satellite のクローンを作成してアップグレードすることができます。アップグレードが完了したら、Satellite Server の以前のバージョンの使用を停止できます。

以下の手順を使用して、Satellite インスタンスのクローンを作成し、環境を保存してアップグレードの準備を整えます。

Satellite クローンツールは、Capsule Server から Red Hat Enterprise Linux 7 への移行をサポートしません。代わりに、既存の Capsule Server のバックアップを作成し、Red Hat Enterprise Linux 7 に Capsule Server を復元してから再設定します。

### 用語

次の用語を理解するようにしてください。

**ソースサーバー:** クローンするサーバー

**ターゲットサーバー:** ファイルをコピーし、ソースサーバーのクローンを作成する先の新規サーバー

### 2.1. クローン作成プロセスの概要

1. ソースサーバーをバックアップします。
2. ソースサーバーからターゲットサーバーにクローンを作成します。
3. ソースサーバーの電源を切断します。
4. 新規ホスト名とターゲットサーバーの IP アドレスが一致するように、ターゲットサーバー上のネットワーク設定を更新します。
5. コンテンツホストと Capsule で **goferd** を再起動して、接続をリフレッシュします。
6. 新規ターゲットサーバーをテストします。

### 2.2. 前提条件

Satellite Server のクローンを作成するには、以下のリソースが用意されていることを確認します。

- ターゲットサーバーとして使用する Red Hat Enterprise Linux 7 サーバーの最小インストール。Red Hat Enterprise Linux 7 ソフトウェアグループやサードパーティーのアプリはインストールしないでください。また、お使いのサーバーが **Satellite Server のインストールのインストールのための環境準備** に記載されている仕様すべてに準拠していることを確認してください。
- **satellite-maintain backup** スクリプトを使用して作成した Satellite 6.9 のバックアップ。Pulp データありでも、なしでもバックアップを使用できます。
- ターゲットサーバーの Satellite のサブスクリプション。

クローンを開始する前に、以下の条件を満たしていることを確認してください。

- ターゲットサーバーが分離ネットワーク上にあること。これにより、Capsule Server とホスト間で必要のない通信を回避できます。

- ソースサーバーからのバックアップファイルすべてを格納できるだけの容量があるターゲットサーバー

## カスタマイズした設定ファイル

**satellite-installer** ツールまたは Satellite バックアッププロセスのマネージドでないソースサーバーで、設定をカスタマイズしている場合には、これらのファイルを手動でバックアップする必要があります。

## 2.3. PULP データの考慮事項

Pulp データを含めずに、Satellite Server のクローンを作成できます。ただし、クローンした環境を機能させるためには、Pulp データが必要です。ターゲットサーバーに、Pulp データがない場合には、Satellite は完全に機能しません。

Pulp データをターゲットサーバーに転送するには、2つのオプションがあります。

- Pulp データを含むバックアップを使用したクローン作成
- Pulp データなしのバックアップを使用してクローンを作成し、ソースサーバーから手動で `/var/lib/pulp` をコピーします。

**pulp\_data.tar** ファイルが 500 GB 以上の場合や、**pulp\_data.tar** ファイルが 100 GB 以上で、NFS など、速度の遅いストレージシステムを使用している場合には、デプロイメント時にメモリーエラーが発生する可能性があるため、バックアップに **pulp\_data.tar** を含めないようにしてください。ソースサーバーからターゲットサーバーに **pulp\_data.tar** ファイルをコピーするようにしてください。

### Pulp データなしでバックアップする方法

「[Satellite Server のクローン](#)」の手順の内容に従います。ただし、Pulp データが含まれるクローンを作成する手順を、以下の手順に置き換えてください。

1. PostgreSQL データベースをアクティブにし、Pulp データは除外してバックアップを実行します。

```
# satellite-maintain backup offline --skip-pulp-content \
--assumeyes /var/backup
```

2. **satellite-maintain** サービスを停止して、無効にします。

```
# satellite-maintain service stop
# satellite-maintain service disable
```

3. Pulp データをターゲットサーバーにコピーします。

```
# rsync --archive --partial --progress --compress \
/var/lib/pulp target_server.example.com:/var/lib/pulp
```

「[ターゲットサーバーのクローン作成](#)」に進みます。

## 2.4. SATELLITE SERVER のクローン

以下の手順を使用して、Satellite Server のクローンを作成します。この手順の一部として大量のデータをコピーして転送する必要があるため、完了までにかかり時間がかかる可能性がある点に注意してください。

### 2.4.1. クローン作成のためのソースサーバーの準備

ソースサーバーで、以下の手順を実行します。

1. Satellite サブスクリプションのプール ID を確認します。

```
# subscription-manager list --consumed \
--matches 'Red Hat Satellite'|grep "Pool ID:"|awk '{print $3}'
```

後で使用できるように、**プール ID** をメモしてください。

2. Red Hat Satellite のサブスクリプションを削除します。

```
# subscription-manager remove --serial=$(subscription-manager list \
--consumed \
--matches 'Red Hat Satellite'|grep "Serial:"|awk '{print $2}')
```

3. Pulp データのサイズを判断します。

```
# du -sh /var/lib/pulp/
```

4. Pulp データが 500 GB 未満の場合は、PostgreSQL のデータベースをアクティブにし、Pulp データを含めてバックアップを実行します。Pulp データが 500 GB 以上の場合には、以下の手順は省略して、続行する前に「[Pulp データの考慮事項](#)」の手順を実行してください。

```
# satellite-maintain backup offline --assumeyes /var/backup
```

5. **satellite-maintain** サービスを停止して無効にします。

```
# satellite-maintain service stop
# satellite-maintain service disable
```

「[ターゲットサーバーのクローン作成](#)」に進みます。

### 2.4.2. ターゲットサーバーのクローン作成

サーバーのクローンを作成するには、ターゲットサーバーで以下の手順を実行してください。

1. **satellite-clone** ツールはデフォルトで、**/backup/** をバックアップフォルダーとして使用します。別のフォルダーにコピーする場合には、**/etc/satellite-clone/satellite-clone-vars.yml** ファイルの **backup\_dir** 変数を更新してください。
2. 更新前の Satellite のバックアップファイルを、移行先のサーバーの **/backup/** フォルダーに配置します。共有ストレージをマウントするか、移行先のサーバーの **/backup/** フォルダーにバックアップファイルをコピーします。
3. ソースサーバーの電源を切断します。
4. 以下のコマンドを入力してカスタマーポータルに登録して、サブスクリプションのアタッチし、必要なサブスクリプションだけを有効化します。

```
# subscription-manager register your_customer_portal_credentials
# subscription-manager attach --pool=pool_ID
# subscription-manager repos --disable=*
```

```
# subscription-manager repos \
--enable=rhel-7-server-rpms \
--enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-satellite-maintenance-6-rpms \
--enable=rhel-7-server-satellite-6.9-rpms
```

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

```
# yum install satellite-clone
```

**satellite-clone** ツールをインストールした後に、独自のデプロイメントに合わせて `/etc/satellite-clone/satellite-clone-vars.yml` ファイルで設定を調節します。

6. **satellite-clone** ツールを実行します。

```
# satellite-clone
```

7. DHCP、DNS、TFTP、およびリモート実行サービスを再設定します。ソースの Satellite Server と競合しないように、クローンプロセスにより、ターゲットの Satellite Server でこのサービスが無効になります。
8. Satellite Web UI で DHCP、DNS、TFTP を再設定し、有効にします。詳細については、**Satellite Server のインストール**の [Satellite Server における外部サービスの設定](#) を参照してください。
9. リモート実行を有効にします。

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

10. ユーザー名 **admin** とパスワード **changeme** で、Satellite Web UI にログインします。すぐに、管理者パスワードを変更して認証情報のセキュリティを確保します。
11. 正しい組織が選択されていることを確認します。
12. **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。
13. **リフレッシュ** ボタンをクリックして **終了** をクリックし、サブスクリプションのリストに戻ります。
14. 利用可能なサブスクリプションが正しいことを確認します。
15. `/usr/share/satellite-clone/logs/reassociate_capsules.txt` ファイルの説明に従い、Capsules とライフサイクル環境間の関係を復元します。
16. ターゲットサーバーの IP アドレスと新規ホスト名が一致するように、DNS など、ネットワーク設定を更新します。**satellite-clone** ツールにより、ホスト名をソースサーバーのホスト名に変更します。ホスト名を別のものに変更する場合には、**satellite-change-hostname** ツールを使用してください。詳細については、**Red Hat Satellite の管理**の [Satellite または Capsule サーバーの名前の変更](#) を参照してください。
17. ソースサーバーが **virt-who** デーモンを使用する場合は、ターゲットサーバーにインストールして設定します。ソースサーバーのディレクトリー `/etc/virt-who.d/` にある **virt-who** 設定ファイルをすべて、ターゲットサーバーにある同じディレクトリーに移動します。詳細は、[Red Hat](#)

[Satellite での仮想マシンサブスクリプションの設定](#) を参照してください。以下の章を使用してアップグレードを行った後に、安全にソースサーバーの使用を停止することができます。

## 第3章 RED HAT SATELLITE のアップグレード



### 警告

高可用性設定で Satellite 6 をインストールしている場合は、Satellite 6.10 へアップグレードする前に Red Hat サポートにご連絡ください。

以下の手順を使用して既存の Red Hat Satellite を Red Hat Satellite 6.10 にアップグレードします。

1. [「Satellite Server のアップグレード」](#)
2. [「新しいリポジトリの同期」](#)
3. [「Capsule Server のアップグレード」](#)
4. [「Satellite クライアントのアップグレード」](#)
5. [4章 アップグレード後のタスク](#)

アップグレードを行う前に「[前提条件](#)」を参照してください。

### 3.1. SATELLITE SERVER のアップグレード

このセクションでは、Satellite Server を 6.9 から 6.10 にアップグレードする方法を説明します。

次に進む前に、Satellite Server 6.9 を最新の 6.9.z リリースに更新します。以下のコマンドを使用すると、最新の z-stream に更新できます。

```
# satellite-maintain upgrade run --target-version 6.9.z
```

6.10 にアップグレードする前に、最新の z-stream に更新したことを確認してください。Satellite Server の更新の詳細については、「[Satellite Server の更新](#)」を参照してください。

6.9 から 6.10 にアップグレードするプロセスにより、Pulp コンテンツも移行しますが、この手順にはかなり時間がかかる場合があります。「[コンテンツを Pulp 3 に移行するための準備](#)」の手順に従って、この時間を短くすることができます。

Pulp 2 から Pulp 3 にコンテンツを事前に移行していない場合は、Pulp コンテンツの移行を開始する前に Satellite Server の最新バージョンにアップグレードする必要があります。

#### 作業開始前の準備

- Capsule は、Satellite とは別にアップグレードできます。詳しくは、「[Satellite とは別の Capsule のアップグレード](#)」をご覧ください。
- Satellite Server をアップグレードする前に、ファイアウォールの設定を確認して更新してください。詳細については、[Satellite Server のインストールのインストールのための環境準備](#)を参照してください。



- カスタマーポータルまたは Satellite Web UI からマニフェストを削除しないでください。削除すると、コンテンツホストからエンタイトルメントがすべて削除されます。
- アップグレードする前に、全 Foreman フックのバックアップを作成して、その後フックを削除します。アップグレードが完了し、Satellite が動作しているのを確認したら、フックを復元します。Foreman Hooks 機能は非推奨となり、次の Satellite バージョンで削除される予定です。
- デフォルトのテンプレートを編集した場合は、ファイルのクローンを作成するか、ファイルをエクスポートしてバックアップしてください。推奨される方法はクローン作成です。今後の更新やアップグレードでファイルが上書きされることがなくなるためです。テンプレートの変更の有無を確認するには、アップグレード前に **履歴** を確認するか、アップグレード後に監査ログで変更を表示できます。Satellite Web UI で **Monitor > Audits** に移動し、テンプレートを検索すると、変更履歴を確認できます。エクスポートを使用する場合は、エクスポートしたテンプレートと、デフォルトテンプレートを比較し、手動で変更を適用して変更を復元します。
- Pulp コンテンツの移行は、6.10 へのアップグレード前に 6.9 で実行されます。
- 移行プロセスを開始する前に、`/var/lib/pulp/published` の 2 倍のコンテンツを保存するのに十分なストレージ容量があることを確認します。以下を使用して `/var/lib/pulp/published` のサイズを確認します。

```
# du -sh /var/lib/pulp/published/
```

- コンテンツの移行はオンラインで実行できますが、プロセッサ、ディスク、およびメモリリソースを使用します。同期およびコンテンツビューの公開操作は、結果として時間がかかる可能性があります。
- Pulp コンテンツを事前に移行していない場合は、**PULP\_CONTENT\_PREMIGRATION\_BATCH\_SIZE** 設定は、同時に処理されるコンテンツユニットの数を定義します。使用される RAM の容量と I/O 負荷に影響します。値が小さいほど **satellite-maintain content prepare** の完了に時間がかかります。その時点で移行するコンテンツが残っている場合は、アップグレードのダウンタイムも長くなります。
  - デフォルト値は 1000 です。
  - システムにハードディスクドライブがある場合や、I/O 負荷について懸念がある場合は、推奨の値は 50 になります。
  - 小さい値は推奨されていません。
- 以下の方法を使用して、**PULP\_CONTENT\_PREMIGRATION\_BATCH\_SIZE** を設定します。
  1. ディレクトリーおよびファイルを作成します。

```
$ mkdir /etc/systemd/system/pulpcore-worker@.service.d/
$ vi /etc/systemd/system/pulpcore-worker@.service.d/settings.conf
```

コンテンツの場合:

```
[Service]
User=pulp
Environment=PULP_CONTENT_PREMIGRATION_BATCH_SIZE=1000
```

2. 以下のコマンドを使用して、**satellite-maintain** サービスを再起動します。

```
# systemctl daemon-reload
# satellite-maintain service restart
```

詳細は、[コンテンツを Pulp 3 に移行するための準備](#) を参照してください。

## Capsule に関する留意事項

- Capsule Server のベースオペレーティングシステム、または Capsule Server リポジトリへの更新をコンテンツビューで管理する場合は、更新したコンテンツビューを公開する必要があります。
- 6.9 から 6.10 にアップグレードされた Satellite Server は、引き続き 6.9 の Capsule Server を使用できることに注意してください。



### 警告

カスタムの証明書を実装している場合は、`/root/ssl-build` ディレクトリーと、カスタム証明書に関連するソースファイルを作成したディレクトリーのコンテンツを保持する必要があります。

アップグレード時にこのファイルを保持できないと、アップグレードは失敗します。ファイルを削除してしまった場合は、アップグレードを進めるためにバックアップから復元する必要があります。

## アップグレードシナリオ

- Red Hat コンテンツ配信ネットワークに接続されている Satellite Server をアップグレードする場合は「[接続されている Satellite Server のアップグレード](#)」に進みます。
- Red Hat コンテンツ配信ネットワークに接続されていない Satellite Server をアップグレードする場合は「[切断されている Satellite Server のアップグレード](#)」に進みます。

自己登録の Satellite をアップグレードすることはできません。自己登録の Satellite は、Red Hat コンテンツ配信ネットワーク (CDN) に移行すればアップグレードを実行できます。自己登録の Satellite を CDN に移行する方法は、[Satellite 6.10 Red Hat Satellite のアップグレードと更新ガイドの自己登録された Satellite の移行](#) を参照してください。

## FIPS モード

FIPS モードを使用していない RHEL ベースのシステムから、FIPS モードを使用する RHEL ベースのシステムに Satellite Server をアップグレードすることはできません。

FIPS モードの Red Hat Enterprise Linux ベースシステムで Satellite Server を実行するには、FIPS モードで稼働する RHEL ベースのオペレーティングシステムを新規にプロビジョニングして、Satellite をインストールする必要があります。詳細については、[Satellite Server のインストールのインストールのための環境準備](#) を参照してください。

### 3.1.1. コンテンツを Pulp 3 に移行するための準備

Pulp 3 用にコンテンツを準備する時間は、コンテンツの量とコンテンツビューの数によって異なります。大規模なシステムでは、数日間のダウンタイムを意味します。これを防ぐには、最新バージョンの

Satellite Server 6.9 の実行中に Pulp コンテンツを事前に移行します。これにより、全体的なアップグレードのダウンタイムが短縮されます。

Pulp3 への移行中に、`/var/lib/pulp/published/` ディレクトリーに保存されるデータの量が2倍になる可能性があります。`/var/lib/pulp` ボリュームに十分なスペースがあることを確認してください。ユーザーが **satellite-maintain prep-6.10-upgrade** を実行する場合、`/var/lib/pulp/content` のコンテンツを複製する必要はありません。Pulp 2 を削除したら、余分な領域を戻すことができます。ただし、XFS ボリュームの縮小はできません。ボリュームのサイズを以前の値に下げるには、別のパーティションにデータをコピーする必要がある場合があります。

Pulp 3 への移行中に、PostgreSQL データディレクトリーに保存されるデータの量が大幅に増加する可能性があります。移行のための十分なスペースがあることを確認してください。必要なスペースの概算は、MongoDB Pulp 2 データベースの2~3倍のサイズです。

## 移行前のチェック

複合コンテンツビューがある場合は常にシーケンスを実行し、最初にそれらに対してアクションを実行してから、コンテンツビューで作業する必要があります。

- 有効になっているすべてのリポジトリーが Synchronization プロセスを完了していることを確認してください。
- リポジトリーが **警告** または **エラー** 状態で同期されている場合は、根本的な問題を修正して、リポジトリーを再同期します。
- Pulp 2 から Pulp 3 への移行がエラー **NoMethodError: undefined method 'link?' for nil:NilClass** で失敗する場合、Red Hat ナレッジベースソリューション [The Pulp 2 to Pulp 3 migration fails with error "NoMethodError: undefined method 'link?' for nil:NilClass" in Red Hat Satellite 6.9](#)
- システム内の一時停止されたタスクがリポジトリーまたはコンテンツビューに関連していないことを確認してください。
- 一部のコンテンツビューまたは複合コンテンツビューのバージョンが不要になった場合は、それらのクリーンアップの実行を検討してください。
- **hammer\_cli** を使用して古い Content View または Composite Content View バージョンのクリーンアップを実行する方法の詳細については、[How to remove old content view versions in Red Hat Satellite 6 using CLI/hammer?](#) を参照してください。
- 特定のリポジトリーが不要になった、または使用しなくなった場合は、それらの特定のリポジトリーを無効にします。
- 移行が完了するまで、すべての同期プランが無効になっていることを確認します。
- クリーンアッププロセスが終了したら、[How to delete orphaned content in /var/lib/pulp on Capsule?](#) の手順に従って孤立したデータをクリーンアップするようにしてください。
- Satellite でエラーなしで次のコマンドを入力できることを確認してください。

```
# foreman-rake katello:correct_repositories COMMIT=true --trace
```

以下の手順を使用して、Pulp 2 から Pulp 3 へのコンテンツの移行を開始します。

## 手順

1. 以下のコマンドを使用して、Satellite Server をアップグレードする前にファイルパーミッションを更新します。

```
# satellite-maintain prep-6.10-upgrade
```

レイテンシーが高いシステムでは時間がかかる場合があります。

2. 以下のコマンドを使用して、事前に移行するコンテンツの詳細を表示します。

```
# satellite-maintain content migration-stats
```

移行プロセス中に必要に応じてこのコマンドを使用して、プロセスにかかる時間を判断します。また、移行失敗の原因となる可能性のある破損または不足しているコンテンツも特定します。出力は以下のようになります。

```
Running Retrieve Pulp 2 to Pulp 3 migration statistics
=====
Retrieve Pulp 2 to Pulp 3 migration statistics:
=====Migration Summary=====
Migrated/Total RPMs: 0/367633
Migrated/Total errata: 0/20780140
Migrated/Total repositories: 0/33924
Estimated migration time based on yum content: 47 hours, 23 minutes
Note: ensure there is sufficient storage space for /var/lib/pulp/published to
double in size before starting the migration process.
Check the size of /var/lib/pulp/published with 'du -sh /var/lib/pulp/published/'
Note: ensure there is sufficient storage space for postgresql.
You will need additional space for your postgresql database.
The partition holding '/var/opt/rh/rh-postgresql12/lib/pgsql/data/' will need
additional free space equivalent to the size of your Mongo db database (/var/lib/mongodb/).
[OK]
```

3. 以下のコマンドを使用して、Pulp 移行用にコンテンツを準備します。

```
# satellite-maintain content prepare
```

最終ステップの一環として、**satellite-maintain content prepare** は、コンテンツユニットの移行がまだかどうか確認します。破損したコンテンツや欠落しているコンテンツを特定すると、以下のような内容が表示される場合があります。

```
=====Missing/Corrupted Content Summary=====
WARNING: MISSING OR CORRUPTED CONTENT DETECTED
Corrupted or Missing Rpm: 1000/104583
Corrupted or missing content has been detected, you can examine the list of content in
/tmp/unmigratable_content-20211025-74422-16cxfae and take action by either:
  1. Performing a 'Verify Checksum' sync under Advanced Sync Options, let it complete, and
re-running the migration
  2. Deleting/disabling the affected repositories and running orphan cleanup (foreman-rake
katello:delete_orphaned_content) and re-running the migration.
  3. Manually correcting files on the filesystem in /var/lib/pulp/content/ and re-running the
migration
  4. Mark currently corrupted or missing content as skipped (foreman-rake
katello:approve_corrupted_migration_content). This will skip migration of missing or
corrupted content.
```

コンテンツが破損または欠落している原因はいくつかあります。

- 移行中にリポジトリの同期または Content View の公開またはプロモーションが発生しました。
- `/var/lib/pulp/content/` 内のディスク上のファイルは、ディスクの腐敗により破損しました。
- `/var/lib/pulp/content/` 内のファイルは、部分的な復元によるディスクの損失などの過去のイベントのために欠落しています。
- Katello や Pulp などの Satellite のサブシステム間にはいくつかの不一致があります。これは、一時停止した Red Hat Satellite タスクの手順をスキップしてリポジトリまたはコンテンツビューが不適切に削除された場合に発生する可能性があります。**foreman-rake katello:correct\_repositories COMMIT=true** を実行すると、これを修正できます。上記の `/tmp/unmigratable_content-20211025-74422-16cxfae20211120-2149-1j4pmfae` サンプルでは、**satellite-maintain content migration-stats** コマンドの一部として、ディスクに書き込まれた破損または欠落している RPM のリストを確認できます。

修復不能コンテンツが所属するリポジトリを判断する方法は、[How to determine the repository to run Verify Checksum on for reported corrupted RPMs](#) を参照してください。

移行できないコンテンツが含まれるリポジトリが **On-demand** ダウンロードポリシーを使用して設定されており、欠落や破損のある RPM の数が少ない場合にはほぼ、以下のコマンドを使用してスキップするようにマークできます。

```
# foreman-rake katello:approve_corrupted_migration_content
```

破損または欠落しているコンテンツを承認してアップグレードを続行すると、アップグレード後にそのコンテンツはリポジトリまたはコンテンツビューのバージョンに表示されなくなります。

これらのパッケージがアップストリームリポジトリに存在する場合は、アップグレード後にリポジトリを再同期すると、これらのパッケージが再度追加されます。これらのコンテンツビューを再公開しない限り、Content View に復元されません。これらのパッケージは使用できない可能性が高いため、アップグレードプロセスはその事実のみを検出します。

移行中にリポジトリの同期または Content View の公開またはプロモーションが発生したと思われる場合は、**satellite-maintain content prepare** を再実行すると、残りのコンテンツが移行されます。初めて **satellite-maintain content prepare** を実行する場合、Red Hat は、破損または欠落している RPM の数を減らすために、必要な頻度で実行することが推奨されます。残りのコンテンツの移行にかかる時間は短くなります。



### 注記

**Ctrl + C** を使用して、**satellite-maintain content prepare** プロセスを終了することはできません。**Ctrl + C** を使用して、あるいは SSH セッションを切断してプロセスを停止しようとする、プロセスは終了せずにバックグラウンドで続行されます。必要に応じて、いつでも以下のコマンドを使用してプロセスを正常に終了し、後で続行できます。

```
# satellite-maintain content prepare-abort
```

**satellite-maintain content prepare-abort** は、プロセスを終了するまでに数分の時間がかかる場合があります。可能な状況になったら、いつでも **satellite-maintain content prepare** を使用して移行プロセスを続行できます。

- このプロセスでは、移行が完了していることを確認しません。次のコマンドを使用して、プロセスがどの程度完了に近いかを判断できます。

```
# satellite-maintain content migration-stats
```

指定した移行時間がゼロまたはほぼゼロになるまでの間隔。

- Pulp コンテンツの移行の最終手順は、Satellite Server を 6.9 から 6.10 にアップグレードする際に完了します。



### 注記

問題が発生した場合には、以下のコマンドを使用して、事前移行のプロセスを最初から再起動することができます。

```
# satellite-maintain content migration-reset
```

- Pulp コンテンツの移行の最終手順は、Satellite Server を 6.9 から 6.10 にアップグレードする際に完了します。

## 3.1.2. 接続されている Satellite Server のアップグレード

パブリックインターネットにアクセスできる Satellite Server には、この手順を使用します。



### 警告

設定ファイルを手動で、または Hiera などのツールを使用してカスタマイズした場合、その変更内容は、アップグレード時または更新時にインストールスクリプトを実行すると上書きされます。satellite-installer スクリプトで **--noop** オプションを使用すると、変更をテストできます。詳細は、Red Hat ナレッジベースソリューションの [How to use the noop option to check for changes in Satellite config files during an upgrade](#) を参照してください。

## Satellite Server のアップグレード

1. バックアップを作成します。
  - 仮想マシンで、スナップショットを作成します。
  - 物理マシンで、バックアップを作成します。  
バックアップに関する詳細は、[Red Hat Satellite 6.9 の管理ガイドの Satellite Server と Capsule Server のバックアップ](#) を参照してください。
2. オプション: `/etc/zones.conf` または `/etc/dhcp/dhcpd.conf` ファイルで DNS または DHCP の設定を手動で編集した場合には、設定ファイルをバックアップしてください。インストーラーはドメインまたはサブネットを1つしかサポートしないので、これらのバックアップから変更を復元しなければならない場合があります。
3. オプション: DNS または DHCP の設定ファイルを手動で編集した場合に、変更の上書きを避けるには、以下のコマンドを実行します。

```
# satellite-installer --foreman-proxy-dns-managed=false \  
--foreman-proxy-dhcp-managed=false
```

4. Satellite Web UI で、**Hosts > Discovered hosts** に移動します。検出されたホストページで、検出されたホストの電源を切って削除します。**Select an Organization** メニューで、組織を順番に選択し、検出されたホストの電源を切って削除するプロセスを繰り返します。アップグレードが完了したら、これらのホストを再起動することをメモしておきます。
5. 6.10 にアップグレードする Satellite 6.9/Capsule 6.9 Server の `/etc/foreman-installer/custom-hiera.yaml` ファイルで `apache::purge_configs: false` エントリがないか、コメントアウトされているようにしてください。
6. Satellite Maintenance リポジトリが有効になっていることを確認します。

```
# subscription-manager repos --enable \  
rhel-7-server-satellite-maintenance-6-rpms
```

7. 利用可能なバージョンを確認して、希望のバージョンが表示されていることを確認します。

```
# satellite-maintain upgrade list-versions
```

8. ヘルスチェックオプションを使用して、システムをアップグレードする準備が完了しているかどうかを確認します。プロンプトが表示されたら、hammer の管理者ユーザー認証情報を入力して `satellite-maintain` を設定します。この変更は、`/etc/foreman-maintain/foreman-maintain-hammer.yml` ファイルに適用されます。

```
# satellite-maintain upgrade check --target-version 6.10
```

結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。

9. アップグレード時間は長くなるため、通信セッションの中断と再接続を可能にする `screen` などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。

アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、`/var/log/foreman-installer/satellite.log` ファイルのログメッセージで、プロセスが完全に終了したかどうかを確認できます。

10. アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.10
```

11. カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

12. オプション: カーネルを更新してから再起動していない場合には、システムを再起動します。

```
# reboot
```

13. BASH シェルを使用している場合は、アップグレードに成功または失敗した後に、以下を入力します。

```
# hash -d satellite-maintain service 2> /dev/null
```

14. Pulp 2 から Pulp 3 にコンテンツを移行した場合は、Pulp 2 コンテンツをすべて削除します。

```
# satellite-maintain content remove-pulp2
```

これにより、Pulp 2 RPM、`/var/lib/pulp/content/` のコンテンツ、mongo データベース、および Pulp 3 データベースの移行コンテンツが削除されます。

### 3.1.3. 切断されている Satellite Server のアップグレード

Satellite Server が Red Hat コンテンツ配信ネットワークに接続されていない場合には、この手順を使用します。





### 警告

- 設定ファイルを手動または Hiera などのツールを使用してカスタマイズしている場合、これらの変更内容はアップグレードまたは更新時に **satellite-maintain** コマンドを入力すると上書きされます。**satellite-installer** コマンドを **--noop** オプションを指定して使用し、アップグレードまたは更新時に適用された変更を確認します。詳細は、Red Hat ナレッジベースソリューションの [How to use the noop option to check for changes in Satellite config files during an upgrade](#) を参照してください。
- **hammer import** および **export** コマンドが **hammer content-import** および **hammer content-export** ツールに置き換えられました。**hammer content-view version export**、**hammer content-view version export-legacy**、**hammer repository export**、またはそれぞれの **import** コマンドを使用するスクリプトがある場合は、代わりに **hammer content-export** コマンドおよびそれぞれの **import** コマンドを使用するように調整する必要があります。
- カスタムの証明書を実装している場合は、**/root/ssl-build** ディレクトリーと、カスタム証明書に関連するソースファイルを作成したディレクトリーのコンテンツを保持する必要があります。アップグレード時にこのファイルを保持できないと、アップグレードは失敗します。ファイルを削除してしまった場合は、アップグレードを進めるためにバックアップから復元する必要があります。

### 作業開始前の準備

- Satellite Server をアップグレードする前に、ファイアウォールの設定を確認して更新してください。詳細は、[Installing Satellite Server from a Disconnected Network](#) の [Ports and Firewalls Requirements](#) を参照してください。
- カスタマーポータルまたは Satellite Web UI からマニフェストを削除しないでください。削除すると、コンテンツホストからエンタイトルメントがすべて削除されます。
- アップグレードする前に、全 Foreman フックのバックアップを作成して、その後フックを削除します。アップグレードが完了し、Satellite が動作しているのを確認できるまで、フックを元に戻さないでください。



### 警告

カスタムの証明書を実装している場合は、**/root/ssl-build** ディレクトリーと、カスタム証明書に関連するソースファイルを作成したディレクトリーのコンテンツを保持する必要があります。

アップグレード時にこのファイルを保持できないと、アップグレードは失敗します。ファイルを削除してしまった場合は、アップグレードを進めるためにバックアップから復元する必要があります。

## 切斷されている Satellite Server のアップグレード

1. バックアップを作成します。
  - 仮想マシンで、スナップショットを作成します。
  - 物理マシンで、バックアップを作成します。
2. アップグレード前スクリプトを使用すると、競合を検出して、Satellite Server に重複したエントリーがあり、アップグレード後に登録解除および削除できるホストをリストできます。また、このスクリプトは組織に割り当てられていないホストを検出します。組織が関連付けられていないホストが **Hosts > All hosts** にリストされており、同じ名前のコンテンツホストに組織がすでに関連付けられている場合、そのコンテンツホストは自動的に登録解除されます。これは、アップグレード前にこのようなホストを組織に関連付けることによって回避できます。アップグレード前チェックスクリプトを実行して、アップグレード後に削除できるホストのリストを取得します。関連付けられていないホストが検出された場合は、アップグレードする前にそれらを組織に関連付けることを推奨します。

```
# foreman-rake katello:upgrade_check
```

3. オプション: `/etc/zones.conf` または `/etc/dhcp/dhcpd.conf` ファイルで DNS または DHCP の設定を手動で編集した場合には、設定ファイルをバックアップしてください。インストーラーはドメインまたはサブネットを1つしかサポートしないので、これらのバックアップから変更を復元しなければならない場合があります。
4. オプション: DNS または DHCP の設定ファイルを手動で編集した場合に、変更の上書きを避けるには、以下のコマンドを実行します。

```
# satellite-installer --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

5. オプション: PostgreSQL を外部データベースとして使用する場合には、PostgreSQL サーバーで `rh-postgresql12-postgresql-evr` パッケージをインストールします。このパッケージは `rhel-7-server-satellite-6.10-rpms` リポジトリから取得できます。

```
# yum install rh-postgresql12-postgresql-evr
```

6. Satellite Web UI で、**Hosts > Discovered hosts** に移動します。使用可能な検出されたホストがある場合は、それらをオフにして、**Discovered hosts** ページにあるすべてのエントリーを削除します。必要に応じて、組織設定メニューから、その他の組織を順番に選択し、すべてのエントリーを削除します。これらのホストは、アップグレード完了後に再起動します。
7. 6.10 にアップグレードする Satellite 6.9/Capsule 6.9 Server の `/etc/foreman-installer/custom-hiera.yaml` ファイルで `apache::purge_configs: false` エントリーがないか、コメントアウトされているようにしてください。
8. すべての外部 Capsule Server が組織に割り当てられていることを確認します。割り当てられていないと、ホスト統合の変更により登録が解除される可能性があります。
9. 以前のリポジトリを削除します。

```
# rm /etc/yum.repos.d/*
```

10. `satellite-maintain` サービスを停止します。

```
# satellite-maintain service stop
```

追加の手順の詳細は、[セクション 5.2 オフラインの Satellite Server の更新](#) を参照してください。

- オプション: カスタムの Apache サーバー設定を適用している場合に、アップグレードを実行すると、カスタム設定がインストール時のデフォルト設定に戻る点に注意してください。アップグレード時に適用された変更をプレビューするには、**--noop** (操作なし) オプションを指定して **satellite-installer** コマンドを実行します。これらの変更は、以下の手順で **satellite-maintain upgrade** コマンドを入力すると適用します。

- 次の行を **/etc/httpd/conf/httpd.conf** 設定ファイルに追加します。

```
Include /etc/httpd/conf.modules.d/*.conf
```

- httpd** サービスを再起動します。

```
# systemctl restart httpd
```

- postgresql** データベースサービスおよび **rh-mongodb34-mongod** データベースサービスを起動します。

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

- noop** オプションを指定して、**satellite-installer** コマンドを入力します。

```
# satellite-installer --scenario satellite --verbose --noop
```

アップグレード時に適用された変更をプレビューするには、**/var/log/foreman-installer/satellite.log** を確認します。設定ファイルへの変更を示す **+++** と **---** シンボルの場所を特定します。**--noop** オプションを指定して **satellite-installer** コマンドを入力しても、Satellite に変更は適用されませんが、モジュール内の Puppet リソースによっては変更が適用されることを想定している場合があり、エラーのメッセージが表示される可能性があります。

- satellite-maintain** サービスを停止します。

```
# satellite-maintain service stop
```

- アップグレード時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。

アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、**/var/log/foreman-installer/satellite.log** のログで、プロセスが完全に終了したかどうかを確認できます。

- 利用可能なバージョンを確認して、希望のバージョンが表示されていることを確認します。

```
# satellite-maintain upgrade list-versions
```

- ヘルスチェックオプションを使用して、システムをアップグレードする準備が完了しているかどうかを確認します。プロンプトが表示されたら、hammer の管理者ユーザー認証情報を入力

して **satellite-maintain** を設定します。この変更は、`/etc/foreman-maintain/foreman-maintain-hammer.yml` ファイルに適用されます。

```
# satellite-maintain upgrade check --target-version 6.10 \
--whitelist="repositories-validate,repositories-setup"
```

結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。

- アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.10 \
--whitelist="repositories-validate,repositories-setup"
```



### 警告

**config** サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/satellite.yaml) was not found, can not
continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再実行します。

パッケージが古い、足りないためにスクリプトに失敗した場合には、これらのパッケージを個別にダウンロードしてインストールする必要があります。詳細は、[Installing Satellite Server from a Disconnected Network](#) ガイドの [Resolving Package Dependency Errors](#) のセクションを参照してください。

- BASH シェルを使用している場合は、アップグレードに成功または失敗した後に、以下を入力します。

```
# hash -d satellite-maintain service 2> /dev/null
```

- Pulp 2 から Pulp 3 にコンテンツを移行した場合は、Pulp 2 コンテンツをすべて削除します。

```
# satellite-maintain content remove-pulp2
```

これにより、アップロードおよび同期された Pulp 2 パッケージ、`/var/lib/pulp/content/` のコンテンツ、MongoDB、および Pulp3 データベースの移行コンテンツが削除されます。

- カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

- オプション: 最後の再起動以降にカーネルが更新された場合には、**satellite-maintain** サービスを停止して、システムを再起動します。

```
# satellite-maintain service stop
# reboot
```

- オプション: DNS または DHCP 設定ファイルを手動で編集した場合には、作成したバックアップを使用して、DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。
- 以前の手順で変更を加えた場合には、**satellite-maintain** サービスを再起動します。

```
# satellite-maintain service restart
```

- OpenSCAP プラグインがインストールされているにもかかわらず、デフォルトの OpenSCAP コンテンツが利用できない場合は、以下のコマンドを実行します。

```
# foreman-rake foreman_openscap:bulk_upload:default
```

- Satellite Web UI で **Configure** > **Discovery Rules** に移動し、選択した組織および場所を検出ルールに関連付けます。

## 3.2. 新しいリポジトリの同期

Capsule Server と Satellite クライアントをアップグレードする前に、新しい 6.10 リポジトリを有効にして同期する必要があります。

### 手順

- Satellite Web UI で、**Content** > **Red Hat Repositories** に移動します。
- Recommended Repositories** を、**オン** の位置に切り替えます。
- 結果の一覧から、以下のリポジトリを展開して、**Enable** アイコンをクリックして、リポジトリを有効にします。
  - Satellite クライアントをアップグレードするには、クライアントが使用するすべての Red Hat Enterprise Linux バージョンで **Red Hat Satellite Tools 6.10** リポジトリを有効にします。
  - Capsule Server を使用している場合に、Capsule Server をアップグレードするには、以下のリポジトリも有効にします。

**Red Hat Satellite Capsule 6.10(RHEL 7 Server 用)(RPM)**

**Red Hat Satellite Maintenance 6 (RHEL 7 Server 用) (RPM)**

**Red Hat Ansible Engine 2.9 RPMs for Red Hat Enterprise Linux 7 Server**

**Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server**

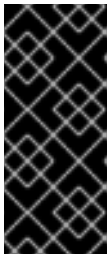


### 注記

バージョン 6.10 のリポジトリが利用できない場合には、サブスクリプション マニフェストをリフレッシュします。**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、**リフレッシュ** をクリックしてください。

- コンテンツ** > **同期ステータス** に移動します。

5. 製品の横にある矢印をクリックして、利用可能なリポジトリを表示します。
6. 6.10 のリポジトリを選択します。
7. **Synchronize Now** をクリックします。



### 重要

リポジトリを同期しようとしたときにエラーが発生した場合は、マニフェストをリフレッシュしてください。問題が解決しない場合は、サポートリクエストを作成してください。カスタマーポータルまたは Satellite Web UI からマニフェストを削除しないでください。削除すると、コンテンツホストのエンタイトルメントがすべて削除されます。

8. コンテンツビューを使用して Capsule Server のベースオペレーティングシステムへの更新を制御する場合、それらのコンテンツビューを新しいリポジトリで更新し、更新済みのバージョンを公開して、プロモートします。詳細は、[コンテンツ管理ガイドのコンテンツビューの管理](#)を参照してください。

## 3.3. CAPSULE SERVER のアップグレード

このセクションは、Capsule Server を 6.9 から 6.10 にアップグレードする方法を説明します。

### 作業開始前の準備

- Capsule Server をアップグレードする前に、Satellite Server をアップグレードする必要があります。Capsule は、Satellite とは別にアップグレードできます。詳細は、[「Satellite とは別の Capsule のアップグレード」](#)を参照してください。
- Red Hat Satellite Capsule 6.10 リポジトリが Satellite Server で有効になっており、同期されていることを確認します。
- Satellite Server 上の必要なリポジトリを必ず同期してください。詳しくは、[「新しいリポジトリの同期」](#)をご覧ください。
- コンテンツビューを使用して Capsule Server のベースオペレーティングシステムへの更新を制御する場合、それらのコンテンツビューを新しいリポジトリで更新し、更新済みのバージョンを公開します。詳細は、[コンテンツ管理ガイドのコンテンツビューの管理](#)を参照してください。
- 新たにアップグレードした Satellite Server に、Capsule のベースシステムが登録されていることを確認します。
- 新たにアップグレードした Satellite Server で、Capsule の組織と場所が正しく設定されていることを確認します。
- Capsule Server をアップグレードする前に、ファイアウォールの設定を確認して更新してください。詳細は、[Capsule Server のインストールの Capsule のインストールに必要な環境の準備](#)を参照してください。
- `/etc/puppetlabs/code/environments` ファイルのバックアップを作成してください。



### 警告

カスタムの証明書を実装している場合は、`/root/ssl-build` ディレクトリーと、カスタム証明書に関連するソースファイルを作成したディレクトリーのコンテンツを保持する必要があります。

アップグレード時にこのファイルを保持できないと、アップグレードは失敗します。ファイルを削除してしまった場合は、アップグレードを進めるためにバックアップから復元する必要があります。

## Capsule Server のアップグレード

1. バックアップを作成します。
  - 仮想マシンで、スナップショットを作成します。
  - 物理マシンで、バックアップを作成します。  
バックアップに関する詳細は、[Red Hat Satellite 6.9 の管理ガイドの Satellite Server および Capsule Server のバックアップ](#) を参照してください。

2. yum のキャッシュを消去します。

```
# yum clean metadata
```

3. **satellite-maintain** が含まれる **rubygem-foreman\_maintain** パッケージがインストールされており、最新の状態になっていることを確認します。

```
# yum install rubygem-foreman_maintain
```

4. Capsule Server で **foreman\_url** 設定が Satellite FQDN を参照していることを確認します。

```
# grep foreman_url /etc/foreman-proxy/settings.yml
```

5. 6.10 にアップグレードする Satellite 6.9/Capsule 6.9 Server の `/etc/foreman-installer/custom-hiera.yaml` ファイルで **apache::purge\_configs: false** エントリーがないか、コメントアウトされているようにしてください。

6. 利用可能なバージョンを確認して、希望のバージョンが表示されていることを確認します。

```
# satellite-maintain upgrade list-versions
```

7. アップグレード時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。

アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、`/var/log/foreman-installer/satellite.log` ファイルのログメッセージで、プロセスが完全に終了したかどうかを確認できます。

- ヘルスチェックオプションを使用して、システムがアップグレードの準備ができているかどうかを確認します。

```
# satellite-maintain upgrade check --target-version 6.10
```

結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。

- アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.10
```



### 警告

**config** サブディレクトリーを含むディレクトリーからコマンドを実行すると、以下のエラーが発生します。

```
ERROR: Scenario (config/capsule.yaml) was not found, can not continue.
```

このような場合は、**root** ユーザーのホームディレクトリーに移動し、コマンドを再実行します。

- Pulp 2 から Pulp 3 にコンテンツを移行した場合は、Pulp 2 コンテンツをすべて削除します。

```
# satellite-maintain content remove-pulp2
```

これにより、Pulp 2 RPM、**/var/lib/pulp/content/** のコンテンツ、mongo データベース、および Pulp 3 データベースの移行コンテンツが削除されます。

この Capsule を介してコンテンツへのアクセスは、アップグレードされた Capsule の完全な同期が実行されるまで失敗します。

- カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

- オプション: カーネルを更新してから再起動していない場合には、システムを再起動します。

```
# reboot
```

- オプション: DNS または DHCP 設定ファイルを手動で編集した場合には、以前に作成したバックアップを使用して、DNS と DHCP の設定ファイルに必要なすべての変更を確認し、復元します。
- オプション: カスタムリポジトリーを使用する場合は、アップグレードの完了後にそのカスタムリポジトリーを必ず有効にしてください。



15. MongoDB および RPM リポジトリは Satellite に自動的に移行されないため、Satellite Server で、アップグレードされた Capsule の完全同期を実行します。

```
# hammer capsule content synchronize --name ${Capsule} --skip-metadata-check true --
async
```



### 注記

Satellite Server をアップグレードする前にリポジトリを同期しなかった場合には、このコマンドを実行すると、Capsule Server とのコンテンツの同期に失敗します。この場合は、「[Satellite Web UI での Capsule Server の同期](#)」に従って Capsule Server を同期します。

## Satellite Web UI のリモート実行を使用した Capsule Server のアップグレード

- バックアップを作成します。
  - 仮想マシンで、スナップショットを作成します。
  - 物理マシンで、バックアップを作成します。  
バックアップに関する詳細は、[Red Hat Satellite 6.9 の管理ガイドの Satellite Server および Capsule Server のバックアップ](#) を参照してください。
- オプション: `/etc/zones.conf` または `/etc/dhcp/dhcpd.conf` ファイルで DNS または DHCP の設定を手動で編集した場合には、設定ファイルをバックアップしてください。インストーラーはドメインまたはサブネットを1つしかサポートしないので、これらのバックアップから変更を復元しなければならない場合があります。
- オプション: DNS または DHCP の設定ファイルを手動で編集した場合に、変更の上書きを避けるには、以下のコマンドを実行します。

```
# {foreman-installer} --foreman-proxy-dns-managed=false \
--foreman-proxy-dhcp-managed=false
```

- Satellite Web UI で、**Monitor > Jobs** に移動します。
- Run Job** をクリックします。
- Job category** リストから **Maintenance Operations** を選択します。
- Job template** リストから **Capsule Upgrade Playbook** を選択します。
- Search Query** フィールドに Capsule のホスト名を入力します。
- Resolves to** で 1 host が表示されていることを確認します。
- target\_version** フィールドに、Capsule のターゲットバージョンを入力します。
- whitelist\_options** フィールドに、許可リストオプションを入力します。
- Type of query** には、クエリーの種類に合わせて **Static Query** または **Dynamic Query** をクリックします。
- Schedule** でジョブ実行のスケジュールを選択します。

### 3.4. SATELLITE WEB UI での CAPSULE SERVER の同期

Satellite Server をアップグレードする前にリポジトリを同期していない場合は、この手順を使用します。

#### 手順

1. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
2. 同期する Capsule Server の名前をクリックします。
3. **同期** をクリックします。
4. **同期の完了** を選択します。

### 3.5. SATELLITE クライアントのアップグレード

Satellite Tools 6.10 リポジトリには、エラータを管理するための通信サービスを提供する **katello-agent** および **katello-host-tools** が含まれます。



#### 注記

Katello エージェントは非推奨で、今後の Satellite のバージョンで削除されます。ワークロードを移行して、リモート実行機能を使用してクライアントをリモートで更新します。詳細は、**Managing Hosts Guide** の [Migrating from Katello Agent to Remote Execution](#) を参照してください。

**katello-agent** および **goferd** を使用したデプロイメントの場合は、すべてのクライアントを **katello-agent** の新しいバージョンに更新します。**katello-agent** および **goferd** を使用しないデプロイメントの場合は、すべてのクライアントを **katello-host-tools** の新しいバージョンに更新します。お使いのクライアントが Satellite Server との互換性を完全に確保できるように、できるだけ早くこの作業を完了してください。

#### 前提条件

- Satellite Server がアップグレードされている。
- Satellite で、新しい Satellite Tools 6.10 リポジトリを有効にしておく。
- Satellite で、新しいリポジトリを同期しておく。
- 以前にクライアントに **katello-agent** をインストールしたことがなく、インストールする場合は、手動の方法を使用する。詳しくは、[Satellite クライアントの手動アップグレード](#) を参照する。



### 警告

カスタムの証明書を実装している場合は、`/root/ssl-build` ディレクトリーと、カスタム証明書に関連するソースファイルを作成したディレクトリーのコンテンツを保持する必要があります。

アップグレード時にこのファイルを保持できないと、アップグレードは失敗します。ファイルを削除してしまった場合は、アップグレードを進めるためにバックアップから復元する必要があります。

## 一括リポジトリ設定 UI を使用した Satellite クライアントのアップグレード

1. Satellite Web UI で、**ホスト > コンテンツホスト** に移動し、アップグレードするコンテンツホストを選択します。
2. **アクションの選択** リストから **リポジトリセットの管理** を選択します。
3. **リポジトリセットの管理** リストから、Red Hat Satellite Tools 6.9のチェックボックスを選択します。
4. **アクションの選択** リストから **Override to Disabled (無効に上書き)** を選択し、**Done (完了)** をクリックします。
5. プロセスが完了したら、以前の手順で使用した同じホストセットの **アクションの選択** リストから、**リポジトリセットの管理** を選択します。
6. **リポジトリセットの管理** リストから、Red Hat Satellite Tools 6.10のチェックボックスを選択します。
7. **アクションの選択** リストから **Override to Enabled (有効に上書き)** を選択し、**Done (完了)** をクリックします。
8. プロセスが完了したら、以前の手順で使用した同じホストセットの **アクションの選択** リストから、**パッケージの管理** を選択します。
9. **パッケージ** 検索フィールドに、設定に応じて以下のいずれかのオプションを入力します。
  - お使いのデプロイメントで **katello-agent** および **goferd** を使用する場合は、**katello-agent** と入力します。
  - お使いのデプロイメントで **katello-agent** および **goferd** を使用しない場合は、**katello-host-tools** と入力します。
10. **Update** リストから、**via remote execution** オプションを選択する必要があります。Katello エージェントを使用してパッケージを更新すると、これによりクライアントと Satellite または Capsule Server 間の通信が中断され、更新が失敗することから、リモート実行経由の選択が必要です。詳細は、**ホストの管理** ガイドの **リモートジョブの設定とセットアップ** を参照してください。

## Satellite クライアントの手動アップグレード

1. クライアントシステムにログインします。

2. 以前のバージョンの Satellite のリポジトリを無効にします。

```
# subscription-manager repos \  
--disable=rhel-7-server-satellite-tools-6.9-rpms
```

3. このバージョンの Satellite 向け Satellite Tools 6.10 リポジトリを有効にします。

```
# subscription-manager repos \  
--enable=rhel-7-server-satellite-tools-6.10-rpms
```

4. お使いの設定に応じて、以下のいずれかの手順を実行します。

- お使いのデプロイメントで **katello-agent** および **goferd** を使用する場合は、以下のコマンドを入力して **katello-agent** をインストールまたはアップグレードします。

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

- お使いのデプロイメントで **katello-agent** および **goferd** を使用しない場合は、以下のコマンドを入力して **katello-host-tools** をインストールまたはアップグレードします。

```
# yum install katello-host-tools
```

## 第4章 アップグレード後のタスク

本セクションで紹介する手順の一部はオプションです。お使いのインストールに関連する手順のみを選択できます。

PXE ベースの検出プロセスを使用する場合には、Satellite の **ホスト > 検出されたホスト** ページに表示させるホストで、Satellite または Capsule Server 上で Discovery アップグレードの手順を実行する必要があります。

- Satellite 6.10 の新規インストールでは、**katello-agent** は無効となっており、**qpidd** サービスおよび **qdroutered** サービスの両方が利用できません。
- Satellite 6.9 が Satellite 6.10 にアップグレードすると、**katello-agent** および **qpidd** サービスおよび **qdroutered** サービスは有効のままとなります。
- **katello-agent** を使用しておらず、リモート実行に移行している場合は、Satellite 6.10 と Capsule 6.10 の両方のアップグレード後タスクとして **katello-agent** を無効にすることもできます。

```
# satellite-installer --foreman-proxy-content-enable-katello-agent false
```

### 4.1. DISCOVERY のアップグレード

このセクションでは、PXE ブートを使用して Satellite Server に登録するホストに渡した PXELinux テンプレートとブートイメージを更新する方法を説明します。

Satellite 6.10 以降、プロビジョニングテンプレートには別途サブネットが関連付けられるので、対象のサブネットに対して TFTP Capsule を使用するように初期設定しないようにしてください。アップグレード後にサブネットを作成する場合には、特に Satellite または Capsule が Discovery テンプレートにプロキシサービスを提供できるようにしてから、**テンプレート Capsule** を使用するように、検出されたホストで全ホストを設定する必要があります。

アップグレード中は、TFTP プロキシが設定された各サブネットが有効化されている場合には、テンプレート Capsule を TFTP Capsule と同じに設定してください。アップグレード後には、すべてのサブネットでこれが正しく設定されていることを確認してください。

ホストで PXE ブートを使用しない場合には、Satellite に新規ホストを検出させるために、これらの手順は、必要ありません。

#### 4.1.1. Satellite Server での Discovery のアップグレード

1. Satellite Web UI で Discovery テンプレートを更新します。
  - a. **ホスト > プロビジョニングテンプレート** に移動します。
  - b. **PXELinux global default** 行で **クローン** をクリックします。
  - c. **名前** フィールドに、テンプレートの新しい名前を入力します (例: **ACME PXE global default**)。
  - d. テンプレートエディターフィールドで、**ONTIMEOUT local** 行を **ONTIMEOUT discovery** に変更し、**送信** をクリックします。
  - e. **管理 > 設定** に移動します。

- f. **Global default PXELinux template** の **値** をクリックします。
  - g. 新しく作成したテンプレートの名前を選択し、チェックボタンをクリックします。
  - h. **ホスト > プロビジョニングテンプレート** に移動します。
  - i. **PXE デフォルトのビルド** をクリックして、**OK** をクリックします。
2. Satellite Web UI で **Configure > Discovery Rules** に移動し、選択した組織および場所を検出ルールに関連付けます。

#### 4.1.2. Capsule Server での Discovery のアップグレード

1. Satellite Server で、Foreman Discovery パッケージが最新であることを確認します。

```
# satellite-maintain packages install tfm-rubygem-foreman_discovery
```

2. 以前の手順で更新が行われた場合には、**satellite-maintain** サービスを再起動します。

```
# satellite-maintain service restart
```

3. 検出されたホストでプロビジョニングネットワークに接続した Satellite Capsule の Discovery イメージ、または検出されたホストに TFTP サービスを提供する Satellite Capsule の Discovery イメージをアップグレードします。

```
# satellite-maintain packages install foreman-discovery-image
```

4. 同じインスタンスに、Proxy サービスを提供するパッケージをインストールして、**foreman-proxy** サービスを再起動します。

```
# satellite-maintain packages install tfm-rubygem-smart_proxy_discovery  
# service foreman-proxy restart
```

5. Satellite Web UI で、**インフラストラクチャー > Capsule** に移動して、関連する Capsule の機能コラムに **Discovery** が表示されていることを確認します。必要に応じて、**アクション** ドロップダウンメニューから **リフレッシュ** を選択します。
6. **インフラストラクチャー > サブネット** に移動し、検出を使用する各サブネットで以下を行います。
  - a. サブネット名をクリックします。
  - b. **Capsule** タブで、上で設定した Capsule に **Discovery Capsule** が設定されているのを確認します。

#### 4.1.3. サブネットにテンプレート Capsule があることの確認

検出されたホストが含まれる全サブネットに、テンプレート Capsule が設定されていることを確認します。

1. Satellite Web UI で、**インフラストラクチャー > Capsule** に移動します。
2. 確認するサブネットを選択します。

3. **Capsule** タブで、**テンプレート Capsule** が、このサブネットに設定されていることを確認します。

テンプレート Capsules によるサブネットの設定の詳細は、[プロビジョニングガイドの Discovery サービスの設定](#) を参照してください。

## 4.2. VIRT-WHO のアップグレード

Satellite Server または Capsule Server に virt-who がインストールされている場合は、Satellite Server または Capsule Server のアップグレード時に一緒にアップグレードされるため、追加の作業は必要ありません。これ以外の作業は必要ありません。virt-who を他の場所にインストールしている場合は、手動でアップグレードする必要があります。

### 作業開始前の準備

Satellite Server または Capsule Server に登録しているホストに virt-who がインストールされている場合は、最初にホストを Satellite Tools 6.10 リポジトリで利用可能な最新パッケージにアップグレードします。ホストのアップグレードに関する詳細は、「[Satellite クライアントのアップグレード](#)」を参照してください。

### virt-who の手動アップグレード

1. virt-who をアップグレードします。

```
# yum upgrade virt-who
```

2. virt-who サービスを再起動して、新しいバージョンを有効にします。

```
# systemctl restart virt-who.service
```

## 4.3. 以前のバージョンの SATELLITE TOOLS リポジトリの削除

Satellite 6.10 へのアップグレードが完了したら、コンテンツビューから Red Hat Satellite Tools 6.9 リポジトリを削除して、無効にすることができます。

バージョン 6.9 の Satellite Tools リポジトリを無効にします。

1. Satellite Web UI で、**コンテンツ > Red Hat リポジトリ** に移動します。
2. **有効化されたリポジトリ** エリアで、**Red Hat Satellite Tools 6.9 for RHEL 7 Server RPMs x86\_64** を探し出します。
3. 右側の **無効化** アイコンをクリックします。

リポジトリがまだコンテンツビューに含まれている場合には、無効にできません。スケジュールされているタスクにより、無効にされたリポジトリからパッケージが自動的に削除されます。

## 4.4. POSTGRESQL 領域の確保

PostgreSQL データベースは、特に負荷の高いデプロイメントにおいて、大容量のディスク領域を使用できます。この手順を使用して、Satellite でこのディスク領域の一部を確保します。

### 手順

1. **postgresql** サービス以外の全サービスを停止します。

```
# satellite-maintain service stop --exclude postgresql
```

2. **postgres** ユーザーに切り替えてデータベースの領域を確保します。

```
# su - postgres -c 'vacuumdb --full --dbname=foreman'
```

3. Vacuum が完了したら、他のサービスを開始します。

```
# satellite-maintain service start
```

4. ファイルが **/var/lib/pgsql/** ディレクトリーに存在することを確認します。

```
# ls -l /var/lib/pgsql/
# du -sh /var/lib/pgsql/
```

5. **/var/lib/pgsql/** ディレクトリーからデータを削除します。

```
# rm -rf /var/lib/pgsql/*
```

## 4.5. テンプレート、パラメーター、ルックアップキーおよび値の更新

アップグレードプロセスで、Satellite は Satellite 6.10 で非推奨となったマクロの場所を特定し、デフォルトの Satellite テンプレート、パラメーター、ルックアップキーおよび値の以前の構文を新しい構文に変換します。ただし、Satellite は、以前に作成したカスタムテンプレートや、クローンしたテンプレートに含まれる以前の構文は変換しません。

このプロセスは、以下のような単純なテキスト置換を使用します。

```
@host.params['parameter1'] -> host_param('parameter1')
@host.param_true?('parameter1') -> host_param_true?('parameter1')
@host.param_false?('parameter1') -> host_param_false?('parameter1')
@host.info['parameters'] -> host_enc['parameters']
```



### 警告

Satellite で複製されたテンプレートを使用する場合、複製されたテンプレートが Satellite の元のテンプレートの最新バージョンと異なるかどうかを確認します。同じテンプレートの構文は、Satellite のバージョンによって異なる場合があります。複製されたテンプレートに古い構文が含まれている場合は、テンプレートの最新バージョンに一致するように構文を更新します。

アップグレード中に、このテキスト置換でファイルの変数を破損したり、省略したりしないように、以前の構文のテンプレート、パラメーター、ルックアップキーおよび値のすべてを確認して、手動で置き換えます。



以下のエラーは、アップグレード後にファイル内に以前の構文が残っていることが原因で発生します。

```
undefined method '#params' for Host::Managed::Jail
```

### Fixing the outdated subscription\_manager\_registration snippet

Satellite 6.4 以降では、**subscription\_manager\_registration** スニペットの代わりに **redhat\_register** スニペットを使用します。

Satellite 6.3 以前からアップグレードする場合は、以下のようにカスタムテンプレートの **subscription\_manager\_registration** スニペットを置き換えます。

```
<%= snippet "subscription_manager_registration" %>
↓
<%= snippet 'redhat_register' %>
```

## 4.6. 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 tuning プロファイルを使用して Satellite のパフォーマンスを向上できます。

Capsule では tuning プロファイルを使用できない点に注意してください。

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` ディレクトリにあります。

**--tuning** オプションを指定して **satellite-installer** コマンドを実行した場合には、デプロイメント設定が以下の順番で Satellite Server に適用されます。

1. `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` ファイルで定義したデフォルトの tuning プロファイル
2. `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/` ディレクトリで定義され、デプロイメントに適用する tuning プロファイル
3. オプション: `/etc/foreman-installer/custom-hiera.yaml` ファイルを設定した場合、Satellite はこれらの設定を適用します。

`/etc/foreman-installer/custom-hiera.yaml` ファイルで定義した設定は、tuning プロファイルで定義した設定を上書きすることに注意してください。

したがって、tuning プロファイルを適用する前に、`/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` のデフォルトの tuning プロファイルに定義されている設定、適用する tuning プロファイル、および `/etc/foreman-installer/custom-hiera.yaml` ファイルを比較して、重複する設定内容を `/etc/foreman-installer/custom-hiera.yaml` ファイルから削除する必要があります。

#### default

マネージドホスト数: 0-5000

RAM: 20G

CPU コア数: 4

### medium

マネージドホスト数: 5001-10000

RAM: 32G

CPU コア数: 8

### large

マネージドホスト数: 10001-20000

RAM: 64G

CPU コア数: 16

### extra-large

マネージドホスト数: 20001-60000

RAM: 128G

CPU コア数: 32

### extra-extra-large

マネージドホスト数: 60000+

RAM: 256G

CPU コア数: 48+

## 手順

1. オプション: Satellite Server で、**custom-hiera.yaml** ファイルを設定した場合、**/etc/foreman-installer/custom-hiera.yaml** ファイルを **custom-hiera.original** にバックアップします。**/etc/foreman-installer/custom-hiera.yaml** ファイルが破損した場合には、バックアップファイルを使用して、ファイルを元の状態に戻します。

```
# cp /etc/foreman-installer/custom-hiera.yaml \  
/etc/foreman-installer/custom-hiera.original
```

2. オプション: Satellite Server で **custom-hiera.yaml** ファイルを設定した場合、**/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** のデフォルト tuning プロファイルの定義と、**/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** に適用する tuning プロファイルを確認します。**/etc/foreman-installer/custom-hiera.yaml** ファイルの設定内容と比較して、**/etc/foreman-installer/custom-hiera.yaml** ファイルで重複設定を削除します。
3. 適用するプロファイルに対して、**--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
```

## 4.7. 外部 CAPSULE SERVER での PUPPET 環境の検証

Capsule Server を 6.10 にアップグレードした後、Puppet 環境が Capsule Server で使用可能であることを確認します。

## 手順

1. Puppet 環境を表示するには、**/etc/puppetlabs/code/environments** ファイルを開きます。

```
# vim /etc/puppetlabs/code/environments
```

2. いずれかの Puppet 環境が見つからない場合は、Satellite Server またはアップグレード前に取得したバックアップからコピーし直してください。
3. 復元した後に、**/etc/puppetlabs/code/environments/** ファイル内の不足しているコンテンツに適切なパーミッションおよび所有権があることを確認します。
4. ファイルを保存してから閉じます。

## 第5章 SATELLITE SERVER、CAPSULE SERVER、およびコンテンツホストの更新

本章を使用して、既存の Satellite Server、Capsule Server おびコンテンツホストを、6.10.0 から 6.10.1 などのように、新しいマイナーバージョンに更新します。

更新すると、コードの発表後に検出されたセキュリティー脆弱性およびマイナーな問題が修正され、多くの場合、更新はすぐに終わり、稼働環境への影響はありません。

更新前に Satellite Server および全 Capsule Server をバックアップします。詳細は、[Red Hat Satellite 管理ガイドの Satellite Server および Capsule Server のバックアップ](#) を参照してください。

### 5.1. SATELLITE SERVER の更新

#### 前提条件

- Satellite、Capsule、および Satellite Tools 6.10 の Satellite Server リポジトリが同期されていることを確認する。
- 更新したリポジトリを関連するすべてのコンテンツビューにプロモートして、外部 Capsule とコンテンツホストがそれぞれ更新できるようにしておく。



#### 警告

設定ファイルを手動で、または Hiera などのツールを使用して、カスタマイズした場合には、このカスタマイズした内容は、アップグレード時または更新時にインストールスクリプトを実行すると上書きされます。satellite-installer スクリプトで **--noop** オプションを使用すると、変更をテストできます。詳細は、Red Hat ナレッジベースソリューションの [How to use the noop option to check for changes in Satellite config files during an upgrade](#) を参照してください。

#### Satellite Server を次のマイナーバージョンに更新

##### Satellite Server の更新手順:

1. Satellite Maintenance リポジトリが有効になっていることを確認します。

```
# subscription-manager repos --enable \
  rhel-7-server-satellite-maintenance-6-rpms
```

2. 利用可能なバージョンを確認して、次のマイナーバージョンがリストされていることを確認します。

```
# satellite-maintain upgrade list-versions
```

3. ヘルスチェックオプションを使用して、システムをアップグレードする準備が完了しているかどうかを確認します。このコマンドを最初に使用したときに、**satellite-maintain** により hammer 管理者ユーザー認証情報を入力して、**/etc/foreman-maintain/foreman-maintain-**

**hammer.yml** ファイルに保存します。

```
# satellite-maintain upgrade check --target-version 6.10.z
```

結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。

- 更新時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、Red Hat ナレッジベースの [How do I use the screen command?](#) を参照してください。  
アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、**/var/log/foreman-installer/satellite.log** ファイルのログメッセージで、プロセスが完全に終了したかどうかを確認できます。

- アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.10.z
```

- カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

- オプション: 最後の再起動以降にカーネルが更新された場合には、**satellite-maintain** サービスを停止して、システムを再起動します。

```
# satellite-maintain service stop
# reboot
```

## 5.2. 接続されていない SATELLITE SERVER の更新

このセクションでは、接続されている Satellite Server (CDN からのコンテンツを同期するサーバー) が、接続されていない Satellite Server からエアギャップで隔離されているエアギャップ非接続環境での更新に必要な手順について説明します。

接続されている Satellite Server で次の手順を実行します。

- 接続されている Satellite Server で次のリポジトリが同期されていることを確認します。

```
rhel-7-server-ansible-2.9-rpms
rhel-7-server-rpms
rhel-7-server-satellite-6.10-rpms
rhel-7-server-satellite-maintenance-6-rpms
rhel-server-rhsc7-7-rpms
```

- 組織のデバッグ証明書をダウンロードして、たとえば **/etc/pki/katello/certs/org-debug-cert.pem** または選択した場所にローカルに保存します。
- 次のリポジトリ情報を使用して、**/etc/yum.repos.d** の下に Yum 設定ファイルを作成します。

```
[rhel-7-server-ansible-2.9-rpms]
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/ser
```

```

ver/7/$releasever/$basearch/ansible/2.9/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

[rhel-7-server-rpms]
name=Red Hat Enterprise Linux 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/ser
ver/7/7Server/x86_64/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

[rhel-7-server-satellite-6.10-rpms]
name=Red Hat Satellite 6 for RHEL 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/ser
ver/7/7Server/x86_64/satellite/6.10/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt

[rhel-7-server-satellite-maintenance-6-rpms]
name=Red Hat Satellite Maintenance 6 for RHEL 7 Server RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/ser
ver/7/7Server/x86_64/sat-maintenance/6/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

[rhel-server-rhsc1-7-rpms]
name=Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel/ser
ver/7/7Server/x86_64/rhsc1/1/os/
enabled=1
sslclientcert = /etc/pki/katello/certs/org-debug-cert.pem
sslclientkey = /etc/pki/katello/certs/org-debug-cert.pem
sslcacert = /etc/pki/katello/certs/katello-server-ca.crt
sslverify = 1

```

4. 設定ファイルの **sslclientcert** および **sslclientkey** の **/etc/pki/katello/certs/org-debug-cert.pem** は、ダウンロードした組織のデバッグ証明書の場所に置き換えます。
5. **satellite.example.com** は、デプロイメントに合わせて、正しい FQDN で更新します。
6. **baseurl** 内の **My\_Organization** は、正しい組織ラベルに置き換えます。組織ラベルを取得するには、次のコマンドを入力します。

```
# hammer organization list
```

7. **reposync** コマンドを入力します。

```
# reposync --delete --download-metadata -p ~/Satellite-repos -n \
-r rhel-7-server-ansible-2.9-rpms \
-r rhel-7-server-rpms \
-r rhel-7-server-satellite-6.10-rpms \
-r rhel-7-server-satellite-maintenance-6-rpms \
-r rhel-server-rhsc7-7-rpms
```

これにより、接続された Satellite Server からリポジトリのコンテンツがダウンロードされ、ディレクトリー **~/Satellite-repos** に保存されます。Red Hat Enterprise Linux 7 の **reposync** コマンドは、RPM をダウンロードしますが、Yum メタデータはダウンロードしません。

このため、**Satellite-repos** の各サブディレクトリーで **createrepo** を手動で実行する必要があります。**createrepo** rpm がインストールされていることを確認してください。そうでない場合は、次のコマンドを使用してインストールします。

```
# satellite-maintain packages install createrepo
```

次のコマンドを実行して、**~/Satellite-repos** の各サブディレクトリーにリポジトリデータを作成します。：

```
# cd ~/Satellite-repos
# for directory in */
do
  echo "Processing $directory"
  cd $directory
  createrepo .
  cd ..
done
```

8. RPM がダウンロードされ、リポジトリデータディレクトリーが **~/Satellite-repos** の各サブディレクトリーに生成されていることを確認します。
9. ディレクトリーの内容をアーカイブします。

```
# cd ~
# tar czf Satellite-repos.tgz Satellite-repos
```

10. 生成された **Satellite-repos.tgz** ファイルを使用して、切断された Satellite Server でアップグレードします。

切断された Satellite Server で次の手順を実行します

1. 生成された **Satellite-repos.tgz** ファイルを切断された Satellite Server にコピーします
2. **root** ユーザーがアクセスできる場所にアーカイブを展開します。次の例では、**/root** が展開場所です。

```
# cd /root
# tar zxf Satellite-repos.tgz
```

3. 次のリポジトリ情報を使用して、**/etc/yum.repos.d** の下に Yum 設定ファイルを作成します。

-

```
[rhel-7-server-ansible-2.9-rpms]
```

```
name=Ansible 2.9 RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-ansible-2.9-rpms
enabled=1
```

```
[rhel-7-server-rpms]
```

```
name=Red Hat Enterprise Linux 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-rpms
enabled=1
```

```
[rhel-7-server-satellite-6.10-rpms]
```

```
name=Red Hat Satellite 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-6.10-rpms
enabled=1
```

```
[rhel-7-server-satellite-maintenance-6-rpms]
```

```
name=Red Hat Satellite Maintenance 6 for RHEL 7 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/rhel-7-server-satellite-maintenance-6-rpms
enabled=1
```

```
[rhel-server-rhsc-7-rpms]
```

```
name=Red Hat Software Collections RPMs for Red Hat Enterprise Linux 7 Server x86_64
baseurl=file:///root/Satellite-repos/rhel-server-rhsc-7-rpms
enabled=1
```

4. 設定ファイルの **/root/Satellite-repos** は、展開先の場所に置き換えます。
5. 利用可能なバージョンを確認して、次のマイナーバージョンがリストされていることを確認します。

```
# satellite-maintain upgrade list-versions
```

6. ヘルスチェックオプションを使用して、システムをアップグレードする準備が完了しているかどうかを確認します。このコマンドを最初に使用したときに、**satellite-maintain** により hammer 管理者ユーザー認証情報を入力して、**/etc/foreman-maintain/foreman-maintain-hammer.yml** ファイルに保存します。

```
# satellite-maintain upgrade check --whitelist="check-upstream-repository,repositories-validate" --target-version 6.10.z
```

7. 結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。
8. 更新時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティーを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。  
アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、**/var/log/foreman-installer/satellite.log** ファイルのログメッセージで、プロセスが完全に終了したかどうかを確認できます。
9. アップグレードを実行します。

```
# satellite-maintain upgrade run --whitelist="check-upstream-repository,repositories-validate" --target-version 6.10.z
```



- 
- カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

- オプション: 最後の再起動以降にカーネルが更新された場合には、**satellite-maintain** サービスを停止して、システムを再起動します。

```
# satellite-maintain service stop
# reboot
```

### 5.3. CAPSULE SERVER の更新

この手順を使用して、Capsule Server を次のマイナーバージョンに更新します。

#### 手順

- Satellite Maintenance リポジトリが有効になっていることを確認します。

```
# subscription-manager repos --enable \
rhel-7-server-satellite-maintenance-6-rpms
```

- 利用可能なバージョンを確認して、次のマイナーバージョンがリストされていることを確認します。

```
# satellite-maintain upgrade list-versions
```

- ヘルスチェックオプションを使用して、システムがアップグレードの準備ができているかどうかを確認します。

```
# satellite-maintain upgrade check --target-version 6.10.z
```

結果を確認し、アップグレードを実行する前に、強調表示されているエラー状態に対応します。

- 更新時間は長くなるため、通信セッションの中断と再接続を可能にする **screen** などのユーティリティを使用します。これにより、コマンドシェルに接続し続けなくてもアップグレードの進捗が確認できるようになります。screen コマンドの使用方法は、[Red Hat ナレッジベースの How do I use the screen command?](#) を参照してください。  
アップグレードコマンドを実行しているコマンドシェルへの接続がなくなった場合は、`/var/log/foreman-installer/satellite.log` ファイルのログメッセージで、プロセスが完全に終了したかどうかを確認できます。

- アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.10.z
```

- カーネルパッケージが最後に更新された日時を確認します。

```
# rpm -qa --last | grep kernel
```

- オプション: 最後の再起動以降にカーネルが更新された場合には、**satellite-maintain** サービスを停止して、システムを再起動します。

```
# satellite-maintain service stop  
# reboot
```