



Red Hat Satellite 6.12

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 6.12 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

法律上の通知

Copyright © 2024 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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

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

目次

はじめに	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
第1章 アップグレードの概要	5
1.1. 前提条件	5
1.2. アップグレードパス	6
1.3. アップグレードの進捗の追跡	6
1.4. SATELLITE とは別の CAPSULE のアップグレード	6
第2章 SATELLITE SERVER のクローン	8
2.1. クローン作成プロセスの概要	8
2.2. 前提条件	8
2.3. PULP データの考慮事項	9
2.4. SATELLITE SERVER のクローン	10
第3章 RED HAT SATELLITE のアップグレード	13
3.1. SATELLITE SERVER のアップグレード	13
3.2. 新しいリポジトリの同期	20
3.3. CAPSULE SERVER のアップグレード	21
3.4. 外部データベースのアップグレード	24
3.5. アップグレード後のタスクの実行	24
第4章 SATELLITE SERVER と CAPSULE SERVER のアップグレード	31
4.1. SATELLITE SERVER の更新	31
4.2. 接続されていない SATELLITE SERVER の更新	32
4.3. CAPSULE SERVER の更新	35

はじめに

このガイドでは、アップグレード、更新、移行という用語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 (例: Satellite 6.11 から Satellite 6.12 へ)。詳細は、[1章アップグレードの概要](#)を参照してください。

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 (例: Satellite 6.12.0 から Satellite 6.12.1)。詳細は、[4章Satellite Server と Capsule Server のアップグレード](#)を参照してください。

移行

既存の Satellite インストールを新しいインスタンスに移動するプロセスです。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. [Bugzilla](#) のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

第1章 アップグレードの概要

現在の RedHatSatellite のインストールを Red Hat Satellite 6.12 にアップグレードする前に、以下の前提条件と利用可能なアップグレードパスを確認してください。

Red Hat カスタマーポータルでの 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、RedHat 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。

1.1. 前提条件

Satellite 6.12 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。

- Red Hat Satellite 6.12 の [リリースノート](#) を参照してください。
- 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 時間かかります。

Capsule のアップグレードには約 10 - 30 分かかります。

- 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、[オンライン接続ネットワーク環境での 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 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るので、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。



注記

Satellite 6.12 は、Red Hat Enterprise Linux 8 でのみリリースされます。Red Hat Enterprise Linux 7 で Satellite 6.11 を実行している場合は、Satellite 6.12 にアップグレードする前に Red Hat Enterprise Linux 8 にアップグレードする必要があります。Red Hat Enterprise Linux 7 から Red Hat Enterprise Linux 8 へのアップグレードの詳細は、[Leapp を使用した Satellite または Capsule 6.11 の Red Hat Enterprise Linux 8 へのインプレースアップグレード](#) を参照してください。

すべての Satellite Server が同じバージョンであることを確認してください。



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。

1.2. アップグレードパス

Red Hat Satellite 6.11 から Red Hat Satellite 6.12 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 6.11 にアップグレードする必要があります。詳細は、[Upgrading and Updating Red Hat Satellite to 6.11](#)を参照してください。

アップグレード手順の概要

Satellite を 6.12 にアップグレードするための俯瞰的な手順は次のとおりです。

1. オプション: 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 を 6.12 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 Capsule Server を 6.12 にアップグレードします。詳細は、「[Capsule Server のアップグレード](#)」を参照してください。
4. 「[アップグレード後のタスクの実行](#)」に進みます。

1.3. アップグレードの進捗の追跡

アップグレードには時間がかかるため、**tmux** などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。詳細は、**tmux** の man ページ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。

1.4. SATELLITE とは別の CAPSULE のアップグレード

Satellite はバージョン 6.12 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 6.11 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 6.11 Capsule で動作します。しかし、6.12 リリースで追加された機能は Capsule を 6.12 にアップグレードするまで動作しません。

Satellite アップグレード後の Capsule のアップグレードは、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、別の場所に配置されている場合。
3. 負荷分散設定を使用している場合は、1つの負荷分散 Capsule だけをアップグレードして、残りの負荷分散 Capsule を1つ前のバージョンに保つことができます。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。

Satellite Server のクローンを作成してインスタンスを作成し、インスタンスの別のマシンまたはオペレーティングシステムへのアップグレードや移行をテストできます。これは、アップグレードまたは移行中の柔軟性を高めるための任意のステップです。

Capsule Server で Satellite クローンツールは使用できません。代わりに、既存の Capsule Server をバックアップし、それをターゲットサーバーに復元してから、Capsule Server を再設定する必要があります。



注記

Satellite Server の新規インスタンスを作成する場合は、バックアップを復元した後に以前のインスタンスを廃止します。クローン作成されたインスタンスは、実稼働環境で並行して実行することは想定されていません。

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

クローンの起点。

ターゲットサーバー

ファイルをコピーしてソースサーバーのクローンを作成する新しいサーバー。

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なる Red Hat Enterprise Linux 8 の最小インストール。Red Hat Enterprise Linux 8 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。サーバーがすべての必要な仕様に準拠していることを確認してください。詳細は、[オンラインネットワーク環境での Satellite Server のインストールのインストールのための環境準備](#) を参照してください。

- **satellite-maintain backup** スクリプトを使用して作成した Satellite 6.11 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。
- ターゲットサーバーのストレージ容量が、ソースサーバーと少なくとも同じである。

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 \
--assumeeyes /var/backup
```

2. Satellite 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 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ップロードし、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-8-for-x86_64-appstream-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=satellite-6.11-for-rhel-8-x86_64-rpms \
--enable=satellite-maintenance-6.11-for-rhel-8-x86_64-rpms
```

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

```
# dnf 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。詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の [Satellite Server](#) での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. Satellite Web UI で、**Content > Subscriptions** に移動し、**Manage Manifest** 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の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

以下の手順を使用して既存の Red Hat Satellite を Red Hat Satellite 6.12 にアップグレードします。

1. 「[前提条件](#)」を確認してください。
2. 「[Satellite Server のアップグレード](#)」
3. 「[新しいリポジトリの同期](#)」
4. 「[Capsule Server のアップグレード](#)」
5. 「[アップグレード後のタスクの実行](#)」

3.1. SATELLITE SERVER のアップグレード

このセクションでは、Satellite Server を 6.11 から 6.12 にアップグレードする方法を説明します。Satellite Server 6.11 のマイナーバージョンからアップグレードすることができます。

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。
- 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、「[オンラインネットワーク環境での Satellite Server のインストールのインストールのための環境準備](#)」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。
- デフォルトのジョブテンプレートまたはプロビジョニングテンプレートのいずれかを編集した場合は、ファイルを複製またはエクスポートしてバックアップします。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。

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。
- 6.11 から 6.12 にアップグレードされた Satellite Server は、引き続き 6.11 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「[接続されていない 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。

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。詳細は、[オンラインネットワーク環境での Satellite Server 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. 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. すべての Satellite サービスを停止します。

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

2. スナップショットを作成するか、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。
3. すべての Satellite サービスを起動します。

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

4. 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。
5. 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
```

6. 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。
7. 現在のアップグレードプロセスでは、Satellite 6.12 (または Capsule) リポジトリを有効にして、Satellite Maintenance リポジトリ以外の更新を制限する必要があります。先に進む前に、回避策としてリリースノートの [既知の問題](#) をお読みください。Satellite Maintenance リポジトリと Satellite 6.12 リポジトリを有効にします。

```
# subscription-manager repos \
--enable satellite-maintenance-6.12-for-rhel-8-x86_64-rpms \
--enable satellite-6.12-for-rhel-8-x86_64-rpms
```

Leapp を使用して Satellite 6.11 システムを Red Hat Enterprise Linux 7 から Red Hat Enterprise Linux 8 にアップグレードした場合は、Satellite モジュールを手動で有効にします。

```
# dnf module enable satellite:el8
```

8. 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
```

9. 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 6.12
```

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。

- アップグレードには時間がかかるため、**tmux**などのユーティリティを使用して、通信セッションを一時停止し、再接続してください。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。

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 6.12
```

- システムが再起動する必要があるかどうかを判断します。

- 最新のインストールされているカーネルのバージョンを確認します。

```
# rpm --query --last kernel | head -n 1
```

- これを、現在実行中のカーネルのバージョンと比較します。

```
# uname --kernel-release
```

- オプション: 最新のカーネルが現在実行中のカーネルと異なる場合は、システムを再起動します。

```
# reboot
```

- 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
```

3.1.2. 接続されていない 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、[オフラインネットワークからの Satellite Server のインストールのポートとファイアウォールの要件](#) 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。
- すべての Satellite Server が同じバージョンである必要があります。

接続されていない Satellite Server のアップグレード

1. すべての Satellite サービスを停止します。

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

2. スナップショットを作成するか、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。

- すべての Satellite サービスを起動します。

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

- 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。

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

- 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。
- 以前のリポジトリを削除します。

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

- [オフラインネットワークからの Satellite Server のインストールガイドの バイナリー DVD イメージのダウンロード](#) の手順に従って、最新の ISO ファイルを取得します。
- マウントポイントとして機能するディレクトリを作成し、ISO イメージをマウントし、**切断されたネットワーク環境での Satellite Server のインストール** の [RHEL 8 でオフラインリポジトリを使用してベースオペレーティングシステムを設定](#) の手順に従って、**rhel8** リポジトリを設定します。
この段階では、パッケージのインストールや更新はしないでください。

- ISO ファイルから Satellite 6.12 リポジトリを設定します。

- Red Hat Satellite パッケージ用に ISO ファイルのリポジトリデータファイルをコピーします。

```
# cp /media/sat6/Satellite/media.repo /etc/yum.repos.d/satellite.repo
```

- `/etc/yum.repos.d/satellite.repo` ファイルを編集します。

```
# vi /etc/yum.repos.d/satellite.repo
```

- デフォルトの **InstallMedia** リポジトリ名を **Satellite-6.12** に変更します。

```
[Satellite-6.12]
```

- baseurl** ディレクティブを追加します。

```
baseurl=file:///media/sat6/Satellite
```

- ISO ファイルから Red Hat Satellite Maintenance リポジトリを設定します。

- a. Red Hat Satellite Maintenance パッケージ用に ISO ファイルのリポジトリデータファイルをコピーします。

```
# cp /media/sat6/Maintenance/media.repo /etc/yum.repos.d/satellite-maintenance.repo
```

- b. `/etc/yum.repos.d/satellite-maintenance.repo` ファイルを編集します。

```
# vi /etc/yum.repos.d/satellite-maintenance.repo
```

- i. デフォルトの **InstallMedia** リポジトリ名を **Satellite-Maintenance** に変更します。

```
[Satellite-Maintenance]
```

- ii. **baseurl** ディレクティブを追加します。

```
baseurl=file:///media/sat6/Maintenance/
```

12. アップグレードには時間がかかるため、**tmux** などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。
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。

13. 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
```

14. 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 6.12 \
--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。

15. アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.12 \
--whitelist="repositories-validate,repositories-setup"
```

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、**オフラインネットワーク環境での Satellite Server のインストール** ガイドの **パッケージの依存関係エラーの解決** のセクションを参照してください。

16. 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
```

- 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 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。

- 前の手順で変更を加えた場合には、Satellite 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. 新しいリポジトリーの同期

Capsule Server と Satellite Client をアップグレードする前に、新しい 6.12 リポジトリー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、以下のリポジトリーを展開して、**Enable** アイコンをクリックして、リポジトリーを有効にします。

- 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 Client 6** リポジトリーを有効にします。

- 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、以下のリポジトリーも有効にします。

Red Hat Satellite Capsule 6.12 (for RHEL 8 x86_64) (RPMs)

Red Hat Satellite Maintenance 6.12 (for RHEL 8 x86_64) (RPMs)

Red Hat Enterprise Linux 8 (for x86_64 – BaseOS) (RPMs)

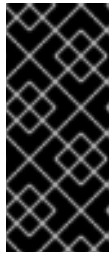
Red Hat Enterprise Linux 8 (for x86_64 – AppStream) (RPMs)



注記

6.12 リポジトリが利用できない場合は、Red Hat Subscription Manifest を更新してください。Satellite Web UI で、**Content** > **Subscriptions** に移動し、**Manage Manifest** をクリックして、**Refresh** をクリックします。

4. Satellite Web UI で、**Content** > **Sync Status** 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. 6.12 のリポジトリを選択します。**Red Hat Satellite Client 6**には 6.12 バージョンがないことに注意してください。代わりに **Red Hat Satellite Client 6** を選択してください。
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トリー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、**コンテンツの管理** 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 6.11 から 6.12 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 6.12 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 上の必要なりポジトリを必ず同期してください。詳細は、[「新しいリポジトリの同期」](#) を参照してください。
- 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、**コンテンツの管理** 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。詳細については、[Capsule Server のインストールの Capsule インストールのための環境準備](#) を参照してください。



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 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。

2. yum のキャッシュを消去します。

```
# yum clean metadata
```

3. Satellite Server の `satellite-capsule-6.15-for-rhel-8-x86_64-rpms` リポジトリーを同期します。
4. Capsule が登録されているコンテンツビューの新しいバージョンを公開およびプロモートします。
5. **rubygem-foreman_maintain** は、Satellite Maintenance リポジトリーからインストールされるか、現在インストールされている場合は Satellite Maintenance リポジトリーからアップグレードされます。

現在のアップグレードプロセスでは、Satellite 6.12 (または Capsule) リポジトリーを有効にして、Satellite Maintenance リポジトリー以外の更新を制限する必要があります。先に進む前に、回避策としてリリースノートの [既知の問題](#) をお読みください。

Capsule が `satellite-maintenance-6.12-for-rhel-8-x86_64-rpms` にアクセスして、以下を実行します。

```
# subscription-manager repos --enable \
satellite-capsule-6.12-for-rhel-8-x86_64-rpms
```

```
# satellite-maintain self-upgrade
```

Leapp を使用して Capsule 6.11 システムを Red Hat Enterprise Linux 7 から Red Hat Enterprise Linux 8 にアップグレードした場合は、Capsule モジュールを手動で有効にします。

```
# dnf module enable satellite-capsule:el8
```

6. 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
```

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. アップグレードには時間がかかるため、**tmux** などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。
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/capsule.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. 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 6.12
```

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。

10. アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.12
```

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. 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。
14. オプション: カスタムリポジトリを使用する場合は、アップグレードの完了後にそのカスタムリポジトリを必ず有効にしてください。

リモート実行を使用した Capsule Server のアップグレード

1. バックアップを作成するか、スナップショットを作成します。
バックアップに関する詳細は、[Red Hat Satellite 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。
2. Satellite Web UI で、**Monitor > Jobs** に移動します。
3. **Run Job** をクリックします。

4. **Job category** リストから **Maintenance Operations** を選択します。
5. **Job template** リストから **Capsule Upgrade Playbook** を選択します。
6. **Search Query** フィールドに Capsule のホスト名を入力します。
7. **Resolves to** に **Apply to 1 host** が表示されていることを確認します。
8. **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。
9. **whitelist_options** フィールドにオプションを入力します。
10. **Schedule** でジョブ実行のスケジュールを選択します。
11. **Type of query** セクションで、**Static Query** をクリックします。

3.4. 外部データベースのアップグレード

Satellite を 6.11 から 6.12 にアップグレードする際に、外部データベースを Red Hat Enterprise Linux 7 から Red Hat Enterprise Linux 8 にアップグレードできます。

前提条件

- Red Hat Enterprise Linux 8 の外部データベースに関するドキュメントに従って、PostgreSQL サーバー用の新しい Red Hat Enterprise Linux 8 ベースのホストを作成する。詳細は、[Satellite で外部データベースの使用](#) を参照してください。

手順

1. バックアップを作成します。
2. 新しいサーバーでバックアップを復元します。
3. Satellite が古い名前でも新しいデータベースサーバーに到達する場合、それ以上の変更は必要ありません。それ以外の場合は、Satellite を再設定して新しい名前を使用します。

```
# satellite-installer \  
--foreman-db-host newpostgres.example.com \  
--katello-candlepin-db-host newpostgres.example.com \  
--foreman-proxy-content-pulpcore-postgresql-host newpostgres.example.com
```

3.5. アップグレード後のタスクの実行

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。



警告

Satellite のアップグレード後に、Capsule の同期がデッドロックで失敗する可能性があります。アップグレード後は、Capsule の同期タスクが進行中ではないことを確認してから、Capsule 上のすべてのワーカーを停止します。

```
# systemctl stop pulpcore-worker@{2..32}
# systemctl status pulpcore-worker@{1..32} | grep Active
```

詳細は、[Red Hat Capsule 6 の同期がデッドロックで検出できない](#) を参照してください。

3.5.1. Discovery のアップグレード

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。

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 6.12 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 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が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。

関連情報

Discovery サービスの設定に関する詳細は、ホストの **プロビジョニングの Discovery サービスの設定** を参照してください。

3.5.1.1. 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. Satellite Web UI で、**Hosts > Provisioning templates** 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. Satellite Web UI で、**Administer > Settings** に移動します。
- f. **Provisioning** タブで、**Default PXE global template entry** を環境のカスタム値に設定します。
- g. **Global default PXELinux template** の **値** をクリックします。
- h. メニューから新しく作成したテンプレートの名前を選択し、**Submit** をクリックします。
- i. Satellite Web UI で、**Hosts > Provisioning templates** に移動します。
- j. **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。



注記

テンプレートが変更された場合、Satellite のアップグレードによってテンプレートがデフォルトバージョンに上書きされます。PXE デフォルト設定が構築されると、**Settings** で設定されたテンプレートが TFTP にデプロイされます。これにより、新しいテンプレートが **Settings** で正しく設定されている場合、デフォルトのテンプレートがデプロイされる可能性があります。

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。

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

テンプレート機能が環境で有効にされている場合には、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 があることを確認します。

1. Satellite Web UI で、**Infrastructure > Subnets** 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、[ホストのプロビジョニングの Discovery サービスの設定](#) を参照してください。

3.5.2. 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 Client 6 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。

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
```

3.5.3. 以前のバージョンの Satellite Tools リポジトリの削除

Satellite 6.12 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 6.11 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 6.11 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. **Enabled Repositories** エリアで、**Red Hat Satellite Tools 6.11 for RHEL 7 Server RPMs x86_64** を探します。
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。

3.5.4. Ansible コンテンツの移行

Red Hat Enterprise Linux 7 から Red Hat Enterprise Linux 8 へのアップグレードには、Ansible Engine 2.9 から Ansible Core 2.12 へのアップグレードが含まれます。

Playbook、REX 内のジョブテンプレート、ディスク上のロールおよびコレクションなどのカスタム Ansible コンテンツがあり、Satellite 上の Ansible RPM によって配信されるモジュールに依存している場合は、Ansible インストールを適応させるか、Ansible コンテンツを移行するために追加の手順を実行する必要があります。

Ansible Core には必須モジュールのみが含まれています。FQCN 表記 **namespace.collection.module** に関しては、**ansible.builtin.*** を引き続き使用できますが、それ以外のもは Ansible Core にはありません。つまり、以前のように非組み込みの Ansible モジュールを使用できなくなるため、最終的には別のソースから入手する必要があります。

アップグレード後に Ansible コンテンツを処理するには、次のオプションがあります。

- Ansible Galaxy から必須ではない機能を提供する、コミュニティが管理する追加のコレクションを取得できます。詳細は、[Galaxy User Guide の Installing collections](#) を参照してください。
Red Hat はこのコンテンツへのサポートを提供していない点に注意してください。
- [Red Hat Automation Hub](#) のサブスクリプションをお持ちの場合は、**ansible-galaxy** を設定して Automation Hub サーバーと通信し、そこからコンテンツをダウンロードできます。そのコンテンツは Red Hat によってサポートされています。**ansible-galaxy** の Automation Hub 接続の設定に関する詳細は、[コンテンツの主なソースとして Red Hat Automation Hub を設定](#) を参照してください。

- Ansible ロール、テンプレート、およびその他の影響を受けるコンテンツを書き換えることができます。Red Hat は、お客様が独自に維持するコンテンツへのサポートを提供していない点に注意してください。



注記

外部の Ansible Galaxy サーバーに接続していない Capsule に Ansible コンテンツをダウンロードしてインストールする場合は、Capsule の設定で Ansible Galaxy サーバーの URL を直接使用するのではなく、Satellite Server を介してコンテンツを渡す必要があります。

1. Ansible Galaxy サーバーからのコンテンツを、Satellite Server のカスタムリポジトリに同期します。
2. Capsule で Ansible を設定して、Satellite Server からコンテンツをダウンロードします。

関連情報

- [Updates to using Ansible in RHEL 8.6 and 9.0](#)
- [Using Ansible in RHEL 8.6 and later](#)
- [Red Hat Enterprise Linux 8.6 リリースノート](#)

3.5.5. 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。Satellite でこのディスク領域の一部を回収するには、この手順を使用します。

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回収します。

```
# su - postgres -c 'vacuumdb --full --all'
```

3. Vacuum が完了したら、他のサービスを開始します。

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

3.5.6. 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` ディレクトリーにあります。

`--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/` ディレクトリーで定義され、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

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

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
```

第4章 SATELLITE SERVER と CAPSULE SERVER のアップグレード

本章を使用して、既存の Satellite Server および Capsule Server を、新しいパッチバージョンに更新します（例：6.12.0 から 6.12.1）。

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ます。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 管理の Satellite Server および Capsule Server のバックアップ](#) を参照してください。

4.1. SATELLITE SERVER の更新

前提条件

- Satellite、Capsule、および Satellite Client 6 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 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、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 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 \
satellite-maintenance-6.12-for-rhel-8-x86_64-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 6.12.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。

- アップグレードに時間がかかるため、**tmux** などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。

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 6.12.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 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
```

4.2. 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-8-for-x86_64-baseos-rpms
rhel-8-for-x86_64-appstream-rpms
satellite-6.12-for-rhel-8-x86_64-rpms
satellite-maintenance-6.12-for-rhel-8-x86_64-rpms
```

- 組織のデバッグ証明書をダウンロードして、**/etc/pki/katello/certs/org-debug-cert.pem** など、選択した場所にローカルに保存します。詳細は、Red Hat Satellite の管理の [組織のデバッグ証明書の作成](#) を参照してください。
- /etc/yum.repos.d** の下に、次の内容の Yum 設定ファイル (**satellite-disconnected.repo** など) を作成します。

```
[rhel-8-for-x86_64-baseos-rpms]
name=Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel8/8/x86_64/baseos/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-8-for-x86_64-appstream-rpms]
name=Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/rhel8/8/
x86_64/appstream/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

[satellite-6.12-for-rhel-8-x86_64-rpms]
name=Red Hat Satellite 6.12 for RHEL 8 RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/layered
rhel8/x86_64/satellite/6.12/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

[satellite-maintenance-6.12-for-rhel-8-x86_64-rpms]
name=Red Hat Satellite Maintenance 6.12 for RHEL 8 RPMs x86_64
baseurl=https://satellite.example.com/pulp/content/My_Organization/Library/content/dist/layered
rhel8/x86_64/sat-maintenance/6.12/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 \
--disableplugin=foreman-protector \
--repoid rhel-8-for-x86_64-baseos-rpms \
--repoid rhel-8-for-x86_64-appstream-rpms \
--repoid satellite-6.12-for-rhel-8-x86_64-rpms \
--repoid satellite-maintenance-6.12-for-rhel-8-x86_64-rpms

```

これにより、接続されている 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。

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、接続されていない Satellite Server でアップグレードします。

接続されていない 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、接続されていない 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. `/etc/yum.repos.d` の下に、次の内容の Yum 設定ファイル (**satellite-disconnected.repo** など) を作成します。

```
[rhel-8-for-x86_64-baseos-rpms]
name=Red Hat Enterprise Linux 8 for x86_64 - BaseOS (RPMs)
baseurl=file:///root/Satellite-repos/rhel-8-for-x86_64-baseos-rpms
enabled=1

[rhel-8-for-x86_64-appstream-rpms]
name=Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
baseurl=file:///root/Satellite-repos/rhel-8-for-x86_64-appstream-rpms
enabled=1

[satellite-6.12-for-rhel-8-x86_64-rpms]
name=Red Hat Satellite 6 for RHEL 8 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/satellite-6.12-for-rhel-8-x86_64-rpms
enabled=1

[satellite-maintenance-6.12-for-rhel-8-x86_64-rpms]
name=Red Hat Satellite Maintenance 6 for RHEL 8 Server RPMs x86_64
baseurl=file:///root/Satellite-repos/satellite-maintenance-6.12-for-rhel-8-x86_64-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
```

- 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 6.12.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。
- アップグレードに時間がかかるため、**tmux** などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。
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 --whitelist="check-upstream-repository,repositories-setup,repositories-validate" --target-version 6.12.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 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
```

4.3. 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 \
satellite-maintenance-6.12-for-rhel-8-x86_64-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 6.12.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。
- 4. アップグレードに時間がかかるため、**tmux**などのユーティリティーを使用して、通信セッションを一時停止し、再接続してください。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。
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/capsule.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。
- 5. アップグレードを実行します。

```
# satellite-maintain upgrade run --target-version 6.12.z
```

- 6. 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
```

- 7. 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 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
```