



# Red Hat Satellite

## 6.2

## アーキテクチャーガイド

---

Satellite 6 デプロイメント計画

Red Hat Satellite Documentation Team



## Satellite 6 デプロイメント計画

Red Hat Satellite Documentation Team  
satellite-doc-list@redhat.com

## 法律上の通知

Copyright © 2017 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

本ガイドでは、Red Hat Satellite 6 のアーキテクチャーの概念について説明し、デプロイメント計画における推奨案を提供します。

# 目次

パート I. SATELLITE 6 アーキテクチャー .....	3
第1章 RED HAT SATELLITE 6 について .....	4
1.1. システムアーキテクチャー .....	4
1.2. システムコンポーネント .....	8
1.3. サポートされる使用方法 .....	8
第2章 CAPSULE SERVER の概要 .....	11
2.1. CAPSULE の機能 .....	11
2.2. CAPSULE のタイプ .....	12
2.3. CAPSULE のネットワーク .....	12
第3章 ホスト分類の概念 .....	15
3.1. ホストグループの構造 .....	15
パート II. SATELLITE 6 デプロイメントの計画 .....	17
第4章 デプロイメントに関する考慮事項 .....	18
4.1. SATELLITE SERVER の設定 .....	18
4.2. ロケーションおよびトポロジー .....	19
4.3. コンテンツソース .....	20
4.4. コンテンツのライフサイクル .....	20
4.5. プロビジョニング .....	22
4.6. ロールベースの認証 .....	22
4.7. 追加のタスク .....	23
第5章 共通のデプロイメントシナリオ .....	24
5.1. 単一ロケーション .....	24
5.2. サブネットが分類されている単一ロケーション .....	24
5.3. 複数ロケーション .....	24
5.4. 接続されていない SATELLITE .....	24
5.5. CAPSULE と外部サービス .....	25
付録A SATELLITE で提供される必要になるテクニカルユーザー .....	26
付録B 用語集 .....	28



## パート I. SATELLITE 6 アーキテクチャー

## 第1章 RED HAT SATELLITE 6 について

**Red Hat Satellite** は、物理環境、仮想環境、およびクラウド環境でシステムをデプロイ、設定、および保守することを可能にするシステム管理ソリューションです。Satellite は、一元化された単一のツールによる、複数の Red Hat Enterprise Linux デプロイメントのプロビジョニング、リモート管理、および監視を可能にします。**Red Hat Satellite Server** は、Red Hat カスタマーポータルおよびその他のソースからのコンテンツを同期し、詳細なライフサイクル管理、ユーザーおよびグループのロールベースのアクセス制御、サブスクリプションの統合管理、高度な GUI、CLI、または API アクセスを含む機能を提供します。

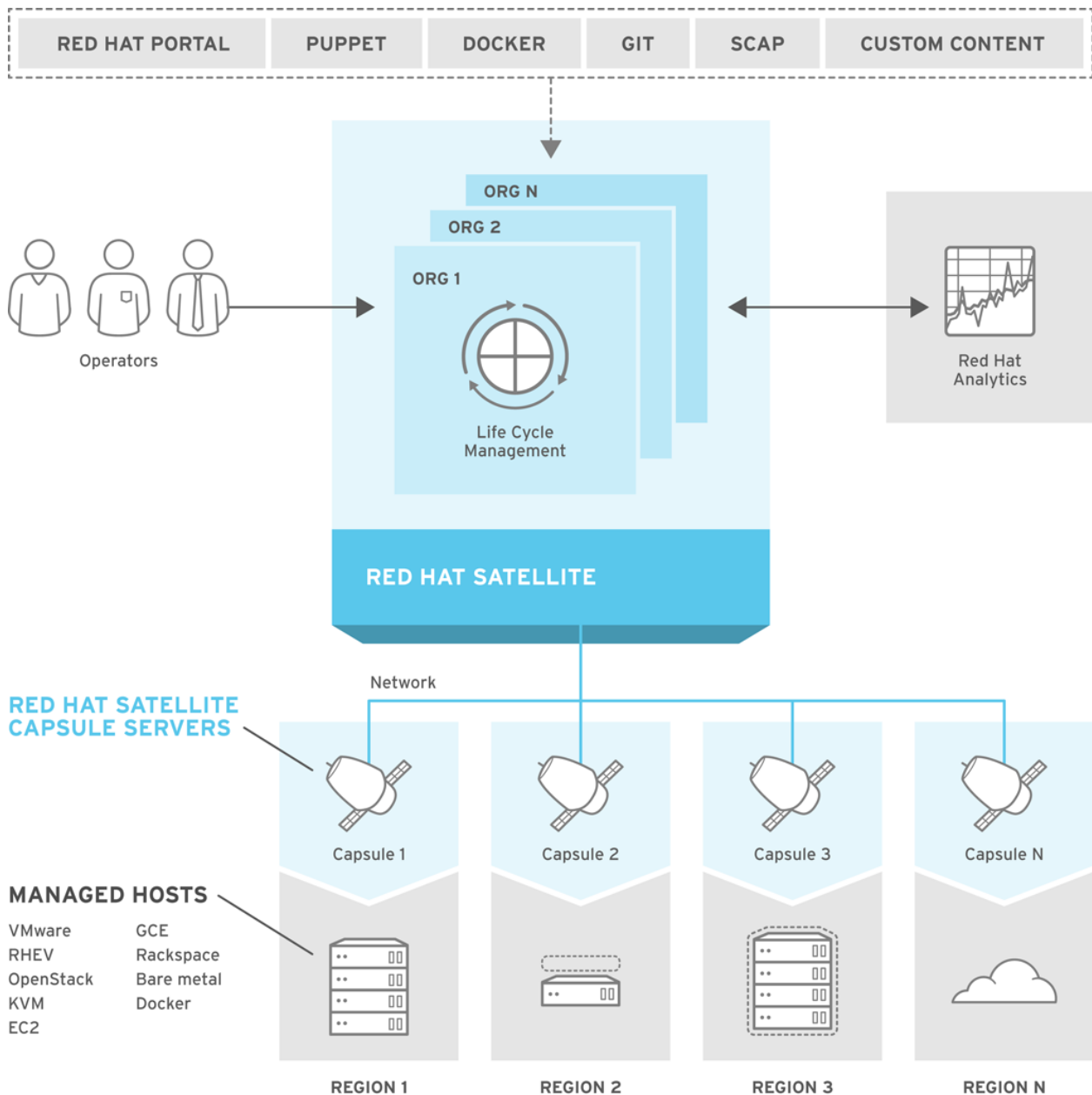
**Red Hat Satellite Capsule Server** は、さまざまな地理的な場所でのコンテンツフェデレーションを実現するために Red Hat Satellite Server からのコンテンツをミラーリングします。ホストシステムは中央 Satellite Server からではなくローカルの Capsule Server からコンテンツおよび設定をプルできます。また、Capsule Server は Puppet マスター、DHCP、DNS、TFTP などのローカライズされたサービスも提供します。Capsule Server を使用すると、環境内で管理対象システムの数が増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

### 1.1. システムアーキテクチャー

以下の図は、Red Hat Satellite 6 のアーキテクチャー全体の俯瞰図を示しています。

図1.1 Red Hat Satellite 6 システムアーキテクチャー





SATELLITE6\_352601\_0715

このアーキテクチャーでは、コンテンツが4つのステージを通過します。

## 外部コンテンツソース

**Red Hat Satellite Server** は、各種ソースからのさまざまなタイプのコンテンツを消費します。これには、Red Hat カスタマーポータルへの接続が必要です。Red Hat カスタマーポータルは、ソフトウェアパッケージ、エラータ、Puppet モジュールおよびコンテナイメージの主なソースになります。これらに加えて、サポートされている他のコンテンツソース (Git リポジトリ、Docker ハブ、Puppet Forge、SCAP リポジトリ) やお客様の組織で使用されている内部データストアも使用することができます。

## Red Hat Satellite Server

Red Hat Satellite Server により、コンテンツライフサイクルの計画と管理、および GUI、CLI または API を使用した Capsule Server とホストの設定の計画と管理を行うことができます。

Satellite Server では、組織を主な区分単位としてライフサイクル管理を行います。組織により、特定の要件および管理タスクが設定されたホストのグループのコンテンツが分類されます。たとえば、OS ビルドチームは Web 開発チームとは異なる組織を使用できます。

Satellite Server には詳細に設定された認証システムも含まれ、これにより Satellite オペレーターには、各自の責任範囲内にあるインフラストラクチャーの特定部分にアクセスするパーミッションが付与されます。

## Capsule Server

Capsule Server は、さまざまな地理的な場所にコンテンツソースを設定するために Satellite Server からのコンテンツをミラーリングします。これにより、ホストシステムは中央 Satellite Server からではなくローカルの Capsule Server からコンテンツおよび設定をプルできます。そのため Capsule Server の最小数として、Satellite を使用する組織が機能する地理的な場所の数を指定することが推奨されます。

コンテンツビューを使用すると、Capsule Server がホストに対して利用可能にするコンテンツのサブセットを指定できます。コンテンツビューを使用したライフサイクル管理の詳細については、[図1.2「Red Hat Satellite 6 におけるコンテンツのライフサイクル」](#)を参照してください。

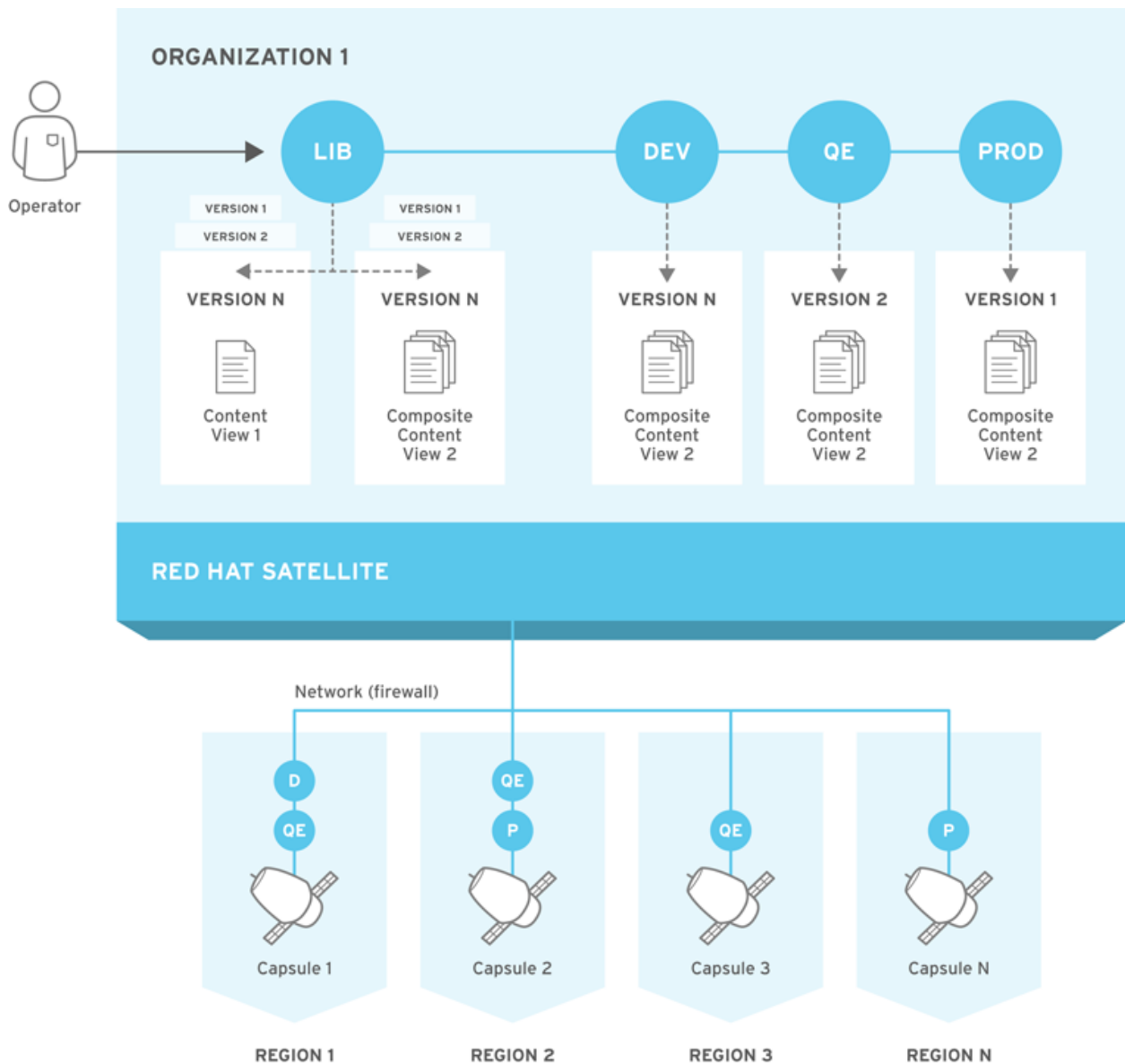
管理対象のホストと Satellite Server 間の通信については、その経路はホストの代わりに複数のサービスを管理することもできる Capsule Server 経由で指定されます。これらのサービスの多くは専用ネットワークポートを使用しますが、Capsule Server はホストから Satellite Server へのすべての通信に単一ソース IP アドレスが使用されるようにします。これによりファイアウォールの管理が簡単になります。Capsule Server についての詳細は、[2章Capsule Server の概要](#)を参照してください。

## 管理対象ホスト

ホストは Capsule Server からコンテンツを受信します。ホストは物理ホストの場合も、(KVM、VMware vSphere、OpenStack、Amazon EC2、Rackspace Cloud Services、Google Compute Engine、または Docker コンテナにデプロイされる) 仮想ホストの場合もあります。Satellite Server には直接管理されるホストを持たせることができます。Capsule Server を実行しているベースシステムも Satellite Server の管理対象ホストです。

以下の図は、Satellite Server から Capsule へのコンテンツ配信の詳細を示しています。

**図1.2 Red Hat Satellite 6 におけるコンテンツのライフサイクル**



SATELLITE6\_352601\_0615

デフォルトでは、各組織は外部ソースのコンテンツのライブラリーを持ちます。コンテンツビューはインテリジェントなフィルタリングによって作成されるライブラリーのコンテンツのサブセットです。コンテンツビューはライフサイクル環境 (通常は「Dev (開発)」、「QA」および「Production (本番)」に公開し、プロモートできます。Capsule Server の作成時にその Capsule にコピーし、管理対象ホストで利用できるようにするライフサイクル環境を選択できます。

コンテンツビューを組み合わせることで複合コンテンツビューを作成することができます。オペレーティングシステムで必要とするパッケージのリポジトリやアプリケーションで必要とするパッケージのリポジトリに別々のコンテンツビューを使用することには利点があります。1つの利点として、1つのリポジトリにあるパッケージへのすべての更新には、関連するコンテンツビューの再公開のみが必要になります。その後は複合コンテンツビューを使用して公開済みのコンテンツビューを組み合わせることができるので、管理が容易になります。

どのコンテンツビューがどの Capsule Server にプロモートされるかは、Capsule の意図されている機能によって異なります。いずれの Capsule Server も、コンテンツまたは設定サービスなどで補完されるインフラストラクチャーサービスとして DNS、DHCP、および TFTP を実行できます。

ライブラリーの同期されたコンテンツを使用して新規バージョンのコンテンツビューを作成することで Capsule Server を更新できます。コンテンツビューの新規バージョンはライフサイクル環境でプロモートされます。さらにコンテンツビューのインプレース更新を作成することもできます。

これは、ライブラリーからプロモートせずに現在のライフサイクル環境でコンテンツビューのマイナーバージョンを作成するという意味です。たとえば、「Production (本番)」のコンテンツビューにセキュリティーエラーを適用する必要がある場合、他のライフサイクルのプロモートせずにコンテンツビューを直接更新することができます。コンテンツ管理についての詳細は、[Red Hat Satellite Content Management Guide](#) を参照してください。

## 1.2. システムコンポーネント

Red Hat Satellite 6 は、Satellite 6 として統合され、検証され、提供されるいくつかのオープンソースプロジェクトで構成されています。この情報については、Red Hat カスタマーポータル上で参照でき、定期的に更新されています。[Satellite 6 Component Versions](#) を参照してください。

Red Hat Satellite 6 は、以下のオープンソースプロジェクトで構成されています。

### Foreman

Foreman は、物理システムと仮想システムのプロビジョニングとライフサイクル管理に使用されるオープンソースのアプリケーションです。Foreman は、キックスタートや Puppet モジュールなどの各種の方法を使ってこれらのシステムを自動的に設定します。さらに Foreman はレポートや監査、およびトラブルシューティングに使用される履歴データを提供します。

### Katello

Katello は、サブスクリプションおよびリポジトリ管理のための Foreman プラグインです。Katello は Red Hat リポジトリにサブスクライブし、コンテンツをダウンロードする手段となります。コンテンツについては、複数の異なるバージョンを作成し、管理することが可能であり、コンテンツのバージョンは、ユーザーが定義するアプリケーションライフサイクルの各ステージ内にある特定のシステムに適用できます。

### Candlepin

Candlepin は、サブスクリプションの管理を行う Katello 内のサービスです。

### Pulp

Pulp は、リポジトリおよびコンテンツの管理を行う Katello 内のサービスです。Pulp は、異なる組織の複数のコンテンツビューによってリクエストされる場合にも RPM パッケージの重複を防ぐことにより、保存スペースを効率的に確保します。

### Hammer

Hammer は、大半の Web UI と同等の機能を持つコマンドラインおよびシェルを提供する CLI ツールです。

### REST API

Red Hat Satellite 6 には RESTful API サービスが含まれます。このサービスにより、システム管理者や開発者はカスタムスクリプトや Red Hat Satellite とのインターフェースを取るサードパーティーアプリケーションを作成することができます。

Red Hat Satellite およびそのアップストリームコンポーネントで使用される用語は広範囲に及びます。よく使われる用語の説明については、[付録B 用語集](#) を参照してください。

## 1.3. サポートされる使用方法

各 Red Hat Satellite サブスクリプションには、Red Hat Enterprise Linux Server のサポートされたインスタンスが 1 つ含まれます。このインスタンスは Red Hat Satellite を実行する目的にのみ取り分けておく必要があります。Satellite に含まれるオペレーティングシステムを使用して環境内で他のデーモン、アプリケーション、またはサービスを実行することはサポート対象外となります。



### 注記

すべての Red Hat Satellite コンポーネントの使用は Red Hat Satellite の環境内でのみサポートされます。これらのコンポーネントのサードパーティーでの使用は、サポート対象外となります。

Red Hat Satellite コンポーネントのサポートについては、以下で説明されています。

## Puppet

Red Hat Satellite 6 には、サポートされる Puppet パッケージが含まれます。インストールプログラムを使うユーザーは Puppet マスターを Red Hat Satellite Capsule Server の一部としてインストールし、設定することができます。Red Hat Satellite Server または Satellite Capsule Server で実行される Puppet マスターで実行される Puppet モジュールも Red Hat によってサポートされています。サポートされている Puppet のバージョンについての情報は、Red Hat ナレッジベースアーティクル [Satellite 6 Component Versions](#) を参照してください。

Red Hat は、Puppet モジュールを含む数多くの異なるスクリプトおよび他のフレームワークをサポートしています。これらのフレームワークのサポートは、アーティクルの [How does Red Hat support scripting frameworks?](#) に基づいて行われています。

## Pulp

Pulpの使用は、Satellite Server web UI、CLI、および API 経由でのみサポートされます。Pulp のローカル API またはデータベースの直接的な変更やこれらとの対話については、Red Hat Satellite 6 データベースに修復不能な破損が生じる可能性があるのでサポートされていません。

## Foreman

Foreman はプラグインを使用することで拡張できますが、Red Hat Satellite でパッケージ化されたプラグインのみがサポートされます。Red Hat Satellite Optional リポジトリのプラグインについてはサポートされません。

Red Hat Satellite には、Red Hat Enterprise Linux 以外のオペレーティングシステムのプロビジョニングや設定を行うための各種コンポーネント、設定および機能も含まれます。これらの機能はすでに組み込まれており、使用することができますが、Red Hat は、Red Hat Enterprise Linux 用の使用についてサポートします。

## Candlepin

Candlepin の使用は、Red Hat Satellite 6 web UI、CLI、および API 経由でのみサポートされます。Candlepin、そのローカル API またはデータベースとの直接的な対話については、Red Hat Satellite 6 データベースに修復不能な破損が生じる可能性があるのでサポートされていません。

## 組み込み Tomcat アプリケーションサーバー

組み込み Tomcat アプリケーションサーバーについては、Red Hat Satellite 6 WebUI、API およびデータベースでの使用のみサポートされます。組み込み Tomcat アプリケーションサーバーのローカル API またはデータベースとの直接の対話についてはサポートされていません。



## 第2章 CAPSULE SERVER の概要

Capsule Server は、コンテンツフェデレーションを提供し、ローカライズされたサービスを使用してホストを検出し、ホストのプロビジョニングを実行し、ホストの制御および設定を実行します。Capsule を使用して Satellite デプロイメントを各種の地理的な場所に拡張することができます。このセクションでは、Capsule で有効にできる機能の概要とそれらの単純な分類について説明します。

Capsule の要件、インストールプロセス、スケーラビリティなどの考慮事項についての詳細は、[Red Hat Satellite インストールガイド](#)を参照してください。

### 2.1. CAPSULE の機能

Capsule Server は 2 つのセットの機能を提供します。まず、Satellite Server からのコンテンツをミラーリングできるように Capsule を設定できます。さらにホスト管理に必要なサービスを実行するために Capsule を使用することもできます。

以下はコンテンツに関連する機能になります。

- ※ **リポジトリの同期**: Satellite Server (より具体的には選択されたライフサイクル環境) からのコンテンツをコンテンツ配信のために Capsule Server にプルします (Pulp で有効にされる)。
- ※ **コンテンツ配信**: Capsule Server を使用するように設定されたホストは、中央 Satellite Server からではなく Capsule からコンテンツをダウンロードします (Pulp で有効にされる)。
- ※ **ホスト動作のデリバリー**: Capsule Server は、パッケージ更新などのホストでのスケジュールされた動作を実行します (ホストの Katello エージェントおよび Capsule の Qpid Dispatch Router で提供される)。
- ※ **Red Hat Subscription Management (RHSM) プロキシ**: ホストは中央 Satellite Server または Red Hat カスタマーポータルではなく、関連付けられた Capsule Server に登録されます (Candlepin で提供される)。

以下はインフラストラクチャーおよびホスト管理サービスになります。

- ※ **DHCP**: Capsule は DHCP サーバーとして機能するか、または ISC DHCP サーバー、Active Directory および Libvirt インスタンスを含む既存のソリューションと統合できます。
- ※ **DNS**: Capsule は DNS サーバーとして機能するか、または ISC DNS、Active Directory、または BIND を含む既存のソリューションと統合できます。
- ※ **TFTP**: Capsule は TFTP サーバーとして機能するか、または UNIX ベースの TFTP サーバーと統合できます。
- ※ **レルム**: Capsule は Kerberos レルムまたはドメインを管理し、ホストがプロビジョニング時にそれらを自動的に参加できるようにします。Capsule は IdM、FreeIPA、および Active Directory を含む既存のインフラストラクチャーと統合できます。
- ※ **Puppet マスター**: Capsule は Puppet マスターを実行することにより設定管理サーバーとして管理できます。
- ※ **Puppet 認証局**: Capsule はホストに証明書を提供するために Puppet CA として機能します。
- ※ **電源管理用のベースボード管理コントローラー (BMC)**: Capsule はホストの電源管理を行います。
- ※ **プロビジョニングテンプレートプロキシ**: Capsule はホストにプロビジョニングテンプレートを提供できます。

- ※ **OpenSCAP**: Capsule はホストにセキュリティーコンプライアンススキャンを実行できます。

## 2.2. CAPSULE のタイプ

すべての Capsule 機能を1度に有効にする必要はありません。Capsule Server は特定の限定された目的のために設定できます。一部の共通の設定には以下が含まれます。

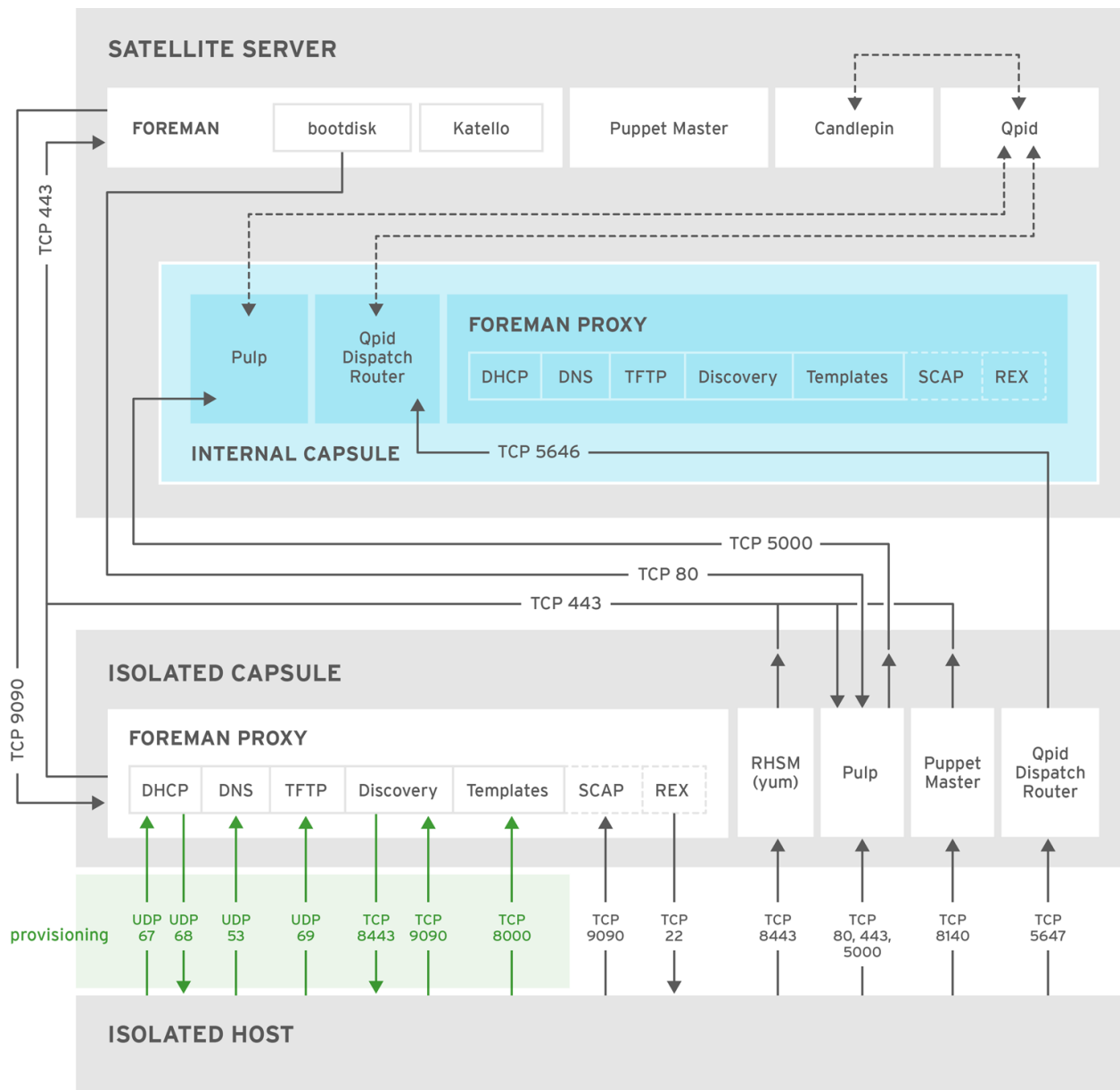
- ※ **インフラストラクチャー Capsule** [DNS + DHCP + TFTP]: ホストのインフラストラクチャーサービスを提供します。プロビジョニングテンプレートプロキシを有効にすると、インフラストラクチャー Capsule に新規ホストのプロビジョニングに必要なすべてのサービスを持たせることができます。
- ※ **コンテンツ Capsule** [Pulp]: 同期したコンテンツを Satellite Server からホストに提供します。
- ※ **設定 Capsule** [Pulp + Puppet + PuppetCA]: コンテンツを提供し、ホストの設定サービスを実行します。
- ※ **オールインワン Capsule** [DNS + DHCP + TFTP + Pulp + Puppet + PuppetCA]: Capsule の完全な機能セットを提供します。オールインワン Capsule は、管理対象ホストの単一接続点を提供することでホストの分離を有効にします。

## 2.3. CAPSULE のネットワーク

Capsule を分離する目的は、リモートネットワークセグメントでファイアウォールポートを Capsule 自体に開くことだけで済むようにホストのネットワーク通信すべての単一エンドポイントを提供することにあります。以下の図は、ホストが分離した Capsule に接続されているシナリオで Satellite コンポーネントが対話する方法について示しています。

図2.1 分離した Capsule を含む Satellite トポロジー

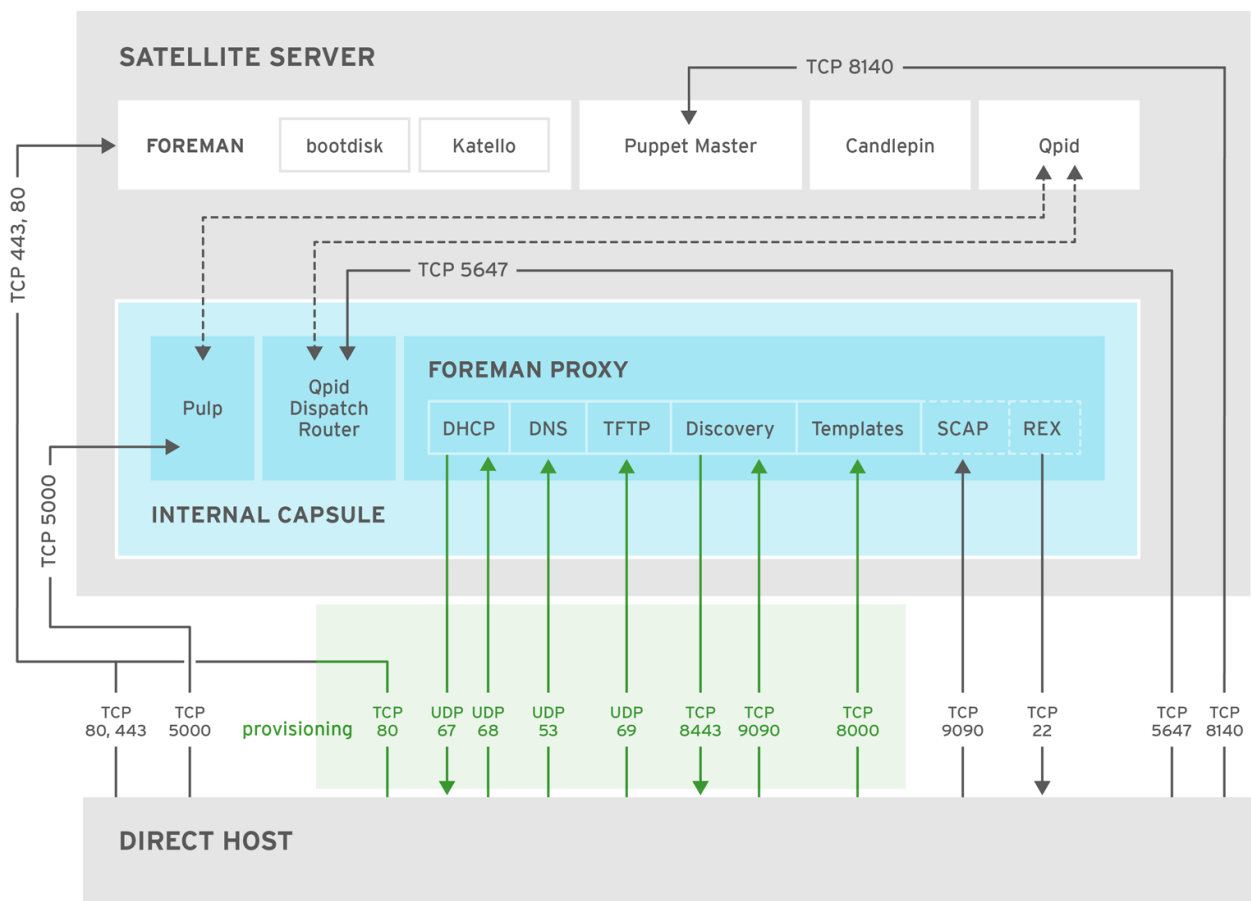




SATELLITE\_439131\_0317

以下の図は、ホストが Satellite Server に直接接続される場合に Satellite コンポーネントが対話する方法を示しています。外部 Capsule のベースシステムは Satellite のクライアントであるため、この図はホストを直接接続する場合でなくても関連性があることに留意してください。

図2.2 内部 Capsule を含む Satellite トポロジー



SATELLITE\_439131\_0317

Red Hat Satellite インストールガイドの [ポートとファイアウォールの要件](#) のセクションには、必要なポートを開くためにホストベースのファイアウォールを設定するための方法が詳述されています。ポートのマトリックス表も、ナレッジベースソリューション [Red Hat Satellite 6.2 List of Network Ports](#) で参照できます。

## 第3章 ホスト分類の概念

Capsule Server の物理的なトポロジーのほかに、Red Hat Satellite はホストを分類するためのいくつかの論理ユニットを提供します。それらのグループのメンバーであるホストはグループ設定を継承します。たとえば、プロビジョニング環境を定義する単純なパラメーターは以下のレベルで適用できます (パラメーターの使用についての詳細は、[Red Hat Satellite Host Configuration Guide](#) を参照してください)。

Global > Organization > Location > Domain > Host group > Host

以下は Red Hat Satellite の主な論理グループです。

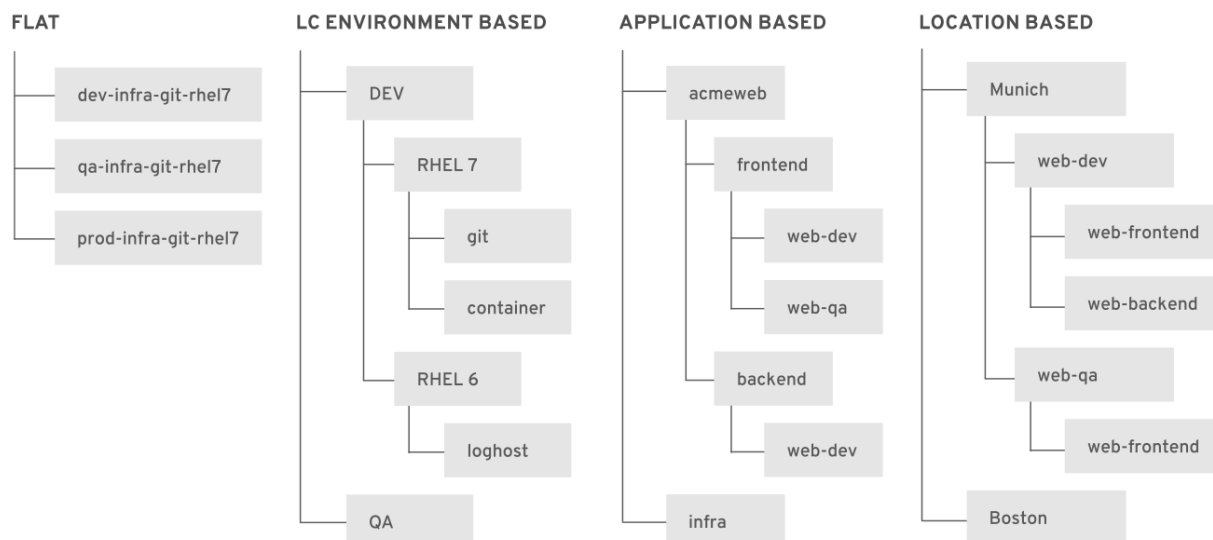
- ※ **組織**: ホストの高位の論理グループです。組織により、コンテンツと設定が明確に区別されます。各組織には別個のサブスクリプションマニフェストが必要であり、それぞれを Satellite Server の別個の仮想インスタンスと見なすことができます。低位のホストグループが適用可能な場合には組織を使用しないようにしてください。
- ※ **ロケーション**: 物理的な場所に対応するホストのグループ分けです。ロケーションは、間違ったホストの配置または設定を防ぐためにネットワークインフラストラクチャーのマッピングに使用できます。たとえば、サブネット、ドメイン、またはコンピュータリソースは Capsule Server に直接割り当てることができず、ロケーションのみに割り当てることができます。
- ※ **ホストグループ**: 割り当て済みの Puppet クラス、コンテンツビューまたはオペレーティングシステムを含むホストの定義を主に提供する媒体です。ホストグループのパラメーターの詳細の一覧については、[Red Hat Satellite Host Configuration Guide](#) を参照してください。設定の大半は、ホストを直接定義するのではなく、ホストグループレベルで設定することが推奨されます。そのため、新規ホストの設定は、それを適切なホストグループに追加することとほぼ同じことになります。ホストグループはネスト化できるため、要件に最も適した構造を作成できます (「[ホストグループの構造](#)」を参照)。
- ※ **ホストコレクション**: サブスクリプションおよびコンテンツ管理の目的で Satellite Server に登録されたホストは **コンテンツホスト** と呼ばれます。コンテンツホストはホストコレクションに区別できるため、パッケージ管理やエラータインストールなどの一括処理が可能になります。

ロケーションおよびホストグループはネスト化できます。組織およびホストコレクションはフラット (非階層的) です。

### 3.1. ホストグループの構造

ホストグループをネスト化して相互のパラメーターを継承できるということは、特定のワークフローに適するホストグループの階層を定義できるということになります。入念に計画されたホストグループの構造は、ホスト設定の保守を単純化するのに役立ちます。このセクションでは、ホストグループを区別するための 4 つのアプローチについて説明します。

#### 図3.1 ホストグループ構造の例



## フラット構造

フラット構造の利点は、継承が行われないためにそれほど複雑ではない点にあります。ホストのタイプが限られているデプロイメントでは、このシナリオが最善のオプションになります。ただし、継承が行われないためにホストグループ間の設定に重複が生じるリスクがあります。

## ライフサイクル環境ベースの構造

この階層では、最初のホストグループレベルはライフサイクル環境に固有のパラメーターに取り分けられます。2番目のレベルにはオペレーティングシステム関連の定義が含まれ、3番目のレベルにはアプリケーション固有の設定が含まれます。これらの構造は、各種の責任がライフサイクル環境ごとに分けられているシナリオで役に立ちます (たとえば、「**Development (開発)**」、「**QA**」および「**Production (本番)**」のライフサイクルの各ステージごとに所有者がそれぞれ設定されている場合)。

## アプリケーションベースの構造

この階層は、特定のアプリケーション内のホストのロールに基づいています。たとえば、これはバックエンドおよびフロントエンドサーバーグループについてのネットワーク設定の定義を可能にします。ホストの一部の特徴を、複雑な設定の Puppet ベースの管理に対応した形で分類できます。ただし、コンテンツビューはこの階層の最下位レベルのホストグループにのみ割り当てることができます。

## ロケーションベースの構造

この階層では、ロケーションの区分がホストグループ構造に連動します。ロケーション (Capsule Server) トポロジーが他の多くの属性を決定するシナリオでは、このアプローチが最善のオプションになります。しかし、この構造は複数のロケーションでのパラメーターの共有を複雑にします。そのため、多数のアプリケーションを含む複雑な環境では、設定の変更のたびに必要になるホストグループの変更数が大幅に増加します。

## パート II. SATELLITE 6 デプロイメントの計画

## 第4章 デプロイメントに関する考慮事項

このセクションでは、Red Hat Satellite 6 デプロイメントの計画時に検討される一般的なトピックの概要を、推奨事項やさらに具体的なドキュメントの参照と共に提供します。たとえば、お客様のシナリオ例 (Satellite 6.1 に特化) に基づくインプリメンテーションについては、[アーティクルの 10 Steps to Build an SOE: How Red Hat Satellite 6 Supports Setting up a Standard Operating Environment](#) を参照してください。

### 4.1. SATELLITE SERVER の設定

作業用の Satellite インフラストラクチャーに対する最初のステップとして、[Red Hat Satellite インストールガイド](#) に説明されているように Red Hat Satellite Server を専用 Red Hat Enterprise 7 Server にインストールします。本ガイドで説明している [インストールの前提条件](#) および [大規模なデプロイメントについての考慮事項](#) について検討してください。

#### Satellite サブスクリプションマニフェストの Satellite Server への追加

サブスクリプションマニフェストは、サブスクリプション情報が含まれる暗号化されたファイルのセットです。Satellite Server はこの情報を使って CDN にアクセスし、関連付けられたサブスクリプションで利用可能なリポジトリを検索できます。サブスクリプションマニフェストの作成およびインポート方法の詳細は、[Red Hat Satellite Content Management Guide](#) の [Managing Subscriptions](#) を参照してください。

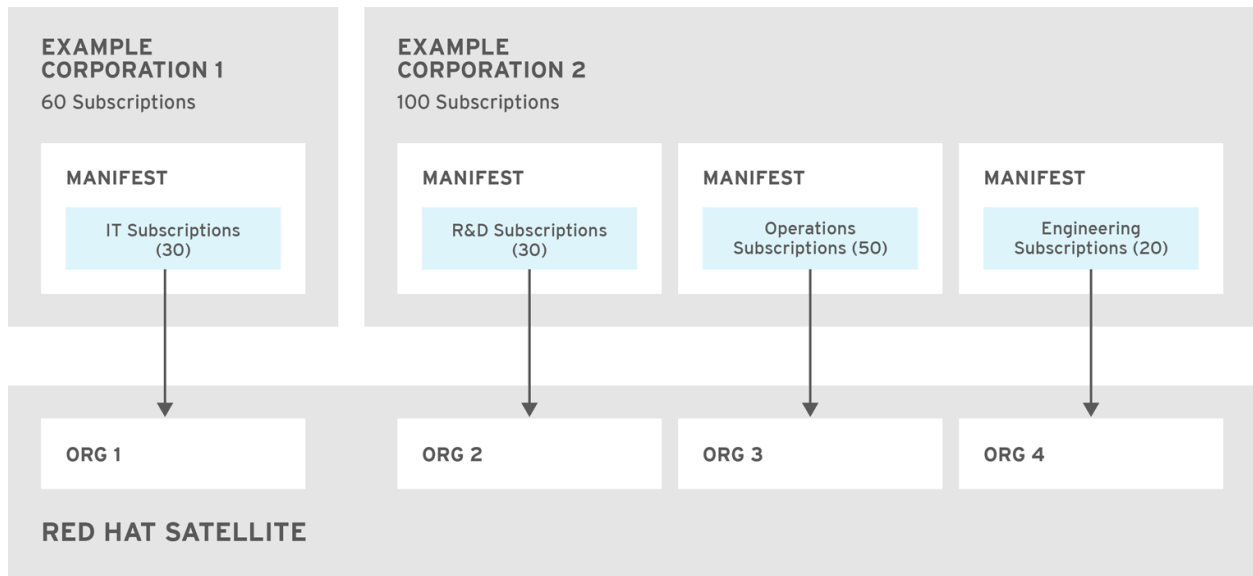
Red Hat Satellite 6 では、Satellite で設定される各組織の単一マニフェストが必要になります。Satellite 6 の組織機能を使用して、Red Hat Network アカウントでインフラストラクチャーの別々のユニットを管理する予定がある場合、必要に応じて 1 つのアカウントのサブスクリプションを組織ごとのマニフェストに割り当てます。

複数の Red Hat Network アカウントを持つ予定の場合や、Red Hat Network アカウントの所有者でもある別のエンティティに属するシステムを管理する必要がある場合には、ユーザーと別のアカウント所有者は必要に応じてマニフェストにサブスクリプションを割り当てることができます。次に 1 つの Satellite Server で複数のマニフェストを使用して複数の組織を管理できます。

システムを管理する必要があるもののサブスクリプションへのアクセスがない場合、Smart Management サブスクリプションを使用します。これにより、ソフトウェアリポジトリにサブスクライブするエンタイトルメントなしに複数のシステムを管理することができます。

以下の図は、複数のシステムを同じ Satellite 6 インストールで管理する必要のある、2 つの Red Hat Network アカウント保持者について示しています。このシナリオでは、Example Corporation 1 は 60 サブスクリプションの任意のサブセットを割り当てることができます。この例では、30 をマニフェストに割り当てています。これは別個の組織として Satellite にインポートできます。これにより、システム管理者は Satellite 6 を使用し、Example Corporation 2 の組織 (R&D、Operations、および Engineering) と完全に切り離して Example Corporation 1 のシステムを管理する機能できます。

#### 複数のマニフェストを含む Satellite Server



SATELLITE\_441823\_0317

サブスクリプションマニフェストの作成時:

- ※ 切断されて Satellite Server または自己登録 Satellite Server について計画する場合は Satellite Server のサブスクリプションをマニフェストに追加します。これは、ベースシステム上で Red Hat Subscription Manager ユーティリティーを使用しているサブスクライブされている接続済みの Satellite Server には不要です。
- ※ 作成する必要があるすべての Capsule Server のサブスクリプションを追加します。
- ※ Satellite で管理する必要があるすべての Red Hat 製品のサブスクリプションを追加します。
- ※ 1つの組織につき1つのマニフェストを作成します。複数のマニフェストを使用することができ、かつ複数の異なる Red Hat サブスクリプションのマニフェストを使用することができます。

サブスクリプションマニフェストは、インフラストラクチャーに変更があった場合やサブスクリプションを追加する場合に変更でき、Satellite Server にリロードできることに注意してください。マニフェストを削除することはできません。マニフェストを Red Hat カスタマーポータルや Satellite Web UI で削除する場合には、すべてのコンテンツホストの登録解除が行われます。

## 4.2. ロケーションおよびトポロジー

このセクションでは、Satellite 6 デプロイメントシナリオを指定するのに役立つ一般的な考慮事項について説明します。デプロイメントの最も一般的なシナリオについては、[5章 共通のデプロイメントシナリオ](#) にその一覧が記載されています。以下はよくある質問です。

- ※ **必要な Capsule Server の数は?** 組織が機能する地理的な場所の数が Capsule Server の数であると考えられます。Capsule を各ロケーションに割り当てることで、Satellite Server の負荷が軽減されると同時に、冗長性が強化され、帯域幅の使用量が減少します。Satellite Server 自体も Capsule として機能できます (これにはデフォルトで統合 Capsule が含まれます)。これは単一ロケーションのデプロイメントで使用でき、Capsule Server のベースシステムをプロビジョニングするために使用できます。統合 Capsule を使用してリモートロケーションのホストと通信することは、ネットワークの使用が最適化されない可能性があるために推奨されません。
- ※ **Capsule Server が提供するサービスは?** Capsule の数を設定したら、各 Capsule で有効にするサービスを決定します。コンテンツおよび設定管理機能のスタック全体が利用可能な場合でも、一部のインフラストラクチャーサービス (DNS、DHCP、TFTP) は Satellite 管理者の管理対象外である場合があります。その場合は、Capsule はそれらの外部サービスと統合する必要



があります (「[Capsule と外部サービス](#)」を参照)。

- ※ **Satellite Server をインターネットから切断している必要があるか?** 切断された Satellite は一般的なデプロイメントシナリオで使用されます (「[接続されていない Satellite](#)」を参照)。切断された Satellite で Red Hat コンテンツの更新が頻繁に必要な場合には、Satellite 間の同期を図るための追加の Satellite インスタンスについて計画してください。
- ※ **ホストに必要なコンピュータリソースは?** ベアメタルホストのプロビジョニングのほかにも、Satellite 6 でサポートされる各種のコンピュータリソースを使用できます。複数の異なるコンピュータリソースでのプロビジョニングについての詳細は、[Red Hat Satellite Provisioning Guide](#) を参照してください。

### 4.3. コンテンツソース

サブスクリプションマニフェストは Satellite Server からアクセスできる Red Hat リポジトリを決定します。Red Hat リポジトリを有効にすると、関連付けられた Satellite 製品が自動的に作成されます。カスタムソースからコンテンツを配信するには、製品およびリポジトリを手動で作成する必要があります。Red Hat リポジトリは、デフォルトでは GPG キーで署名されるため、カスタムリポジトリ用にも GPG キーを作成することが推奨されます。カスタムリポジトリの設定は、それらが保持するコンテンツのタイプ (RPM パッケージ、Puppet モジュール、Docker イメージ、または OSTree スナップショット) によって異なります。

RPM パッケージのみを含む **yum** リポジトリとして設定されるリポジトリは、同期時間と保存スペースを節約するための新規ダウンロードポリシー設定を利用できます。この設定により、**即時**、**オンデマンド**、および**バックグラウンド**から選択することができます。**オンデマンド**設定は、クライアントの要求時にのみパッケージをダウンロードすることでスペースと時間を節約します。**バックグラウンド**設定は、初期の同期後にダウンロードを完了することで時間を節約します。コンテンツソースのセットアップについての詳細な説明については、[Red Hat Satellite Content Management Guide](#) の [Importing Red Hat Content](#) セクションを参照してください。

Satellite Server 内のカスタムリポジトリには、ほとんどの場合、外部ステージングサーバーからのコンテンツが設定されます。これらのサーバーは Satellite インフラストラクチャー外に置かれませんが、カスタムコンテンツをより効果的に制御するために (Git などの) リビジョンコントロールシステムを使用することが推奨されます。

### 4.4. コンテンツのライフサイクル

Satellite 6 は、コンテンツライフサイクルを詳細に管理するための各種機能を提供します。**ライフサイクル環境** は、コンテンツライフサイクルのステージを表し、**コンテンツビュー** はコンテンツのフィルターされたコンテンツのセットであり、コンテンツの定義済みのサブセットとみなすことができます。コンテンツビューをライフサイクル環境に関連付けると、コンテンツを定義された方法でホストに対して利用可能にできます (プロセスの仮想化については、[図1.2 「Red Hat Satellite 6 におけるコンテンツのライフサイクル」](#)を参照)。コンテンツ管理プロセスの詳細の説明については、[Red Hat 6 Content Management Guide](#) を参照してください。以下のセクションでは、ライフサイクル環境と共にコンテンツビューをデプロイするための一般的なシナリオについて説明します。

**ライブラリー**と呼ばれるデフォルトのライフサイクル環境はすべての接続されたソースからコンテンツを収集します。ホストをライブラリーに直接関連付けることは、ホストに利用可能にする前のコンテンツのテストを実行できなくなるために推奨されていません。その代わりに、コンテンツのワークフローに適したライフサイクル環境パスを作成します。よくあるシナリオは以下のようになります。



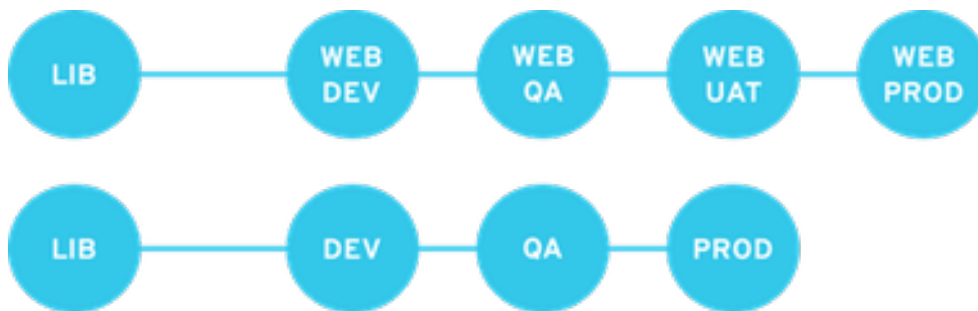
- ※ **単一ライフサイクル環境** ライブラリーからのコンテンツは本番ステージに直接プロモートされます。このアプローチでは、複雑さの面で制限がありますが、ホストに利用可能にする前のライブラリー内でのコンテンツのテストが可能です。



- ※ **単一ライフサイクル環境パス** オペレーティングシステムとアプリケーションコンテンツは共に同じパスでプロモートされます。そのパスは複数のステージ (たとえば、「**Development (開発)**」、「**QA**」、「**Production (本番)**」) で構成されており、詳細なテストを可能にしますが、これには追加の作業が必要です。



- ※ **アプリケーション固有のライフサイクル環境パス** 各アプリケーションには別個のパスがあり、これにより個別のアプリケーションリリースサイクルが可能になります。特定のコンピュータリソースをアプリケーションライフサイクルのステージに関連付けて、テストを容易にすることができます。しかしながら、このシナリオではメンテナンスが複雑になります。



以下はよくあるコンテンツビューのシナリオです。

- ※ **オールインワンコンテンツビュー** 大半のホストに必要なすべてのコンテンツが含まれるコンテンツビューです。コンテンツビューの数を減らすことは、リソース (時間、保存スペース) に制限のあるデプロイメントやホストタイプが同一のデプロイメントの場合に利点があります。ただし、このシナリオは時間ベースのスナップショットやインテリジェントなフィルタリングなどのコンテンツビューの各種機能を制限します。コンテンツソースのいずれの変更も一部のホストに影響を与えます。
- ※ **ホスト固有のコンテンツビュー** 各ホストタイプの専用コンテンツビューです。このアプローチは、少数のホストタイプ (最高 30) を含むデプロイメントで役立ちます。ただし、この場合はホストタイプ間のコンテンツの共有や、ホストタイプ以外の基準に基づく分離 (オペレーティングシステムとアプリケーション間など) を防ぎます。重要な更新があった場合は、すべてのコンテンツビューを更新する必要があり、これによりメンテナンスの作業が増加します。
- ※ **ホスト固有の複合コンテンツビュー** 各ホストタイプ専用のコンテンツビューの組み合わせです。このアプローチにより、ホスト固有のコンテンツと共有コンテンツの分離が可能になります。たとえば、Puppet 設定の専用のコンテンツビューを設定することができます。このコンテンツビューを複数のホストタイプの複合コンテンツビューに組み込むと、他のホストコンテンツよりも高い頻度で Puppet 設定を更新することができます。
- ※ **コンポーネントベースのコンテンツビュー** 特定のアプリケーションの専用コンテンツビューです。たとえば、データベースコンテンツビューはいくつかの複合コンテンツビューに組み込むことができます。このアプローチにより標準化のレベルが上がりますが、コンテンツビューの数が増えることにもなります。

最適なソリューションはホスト環境の性質によって異なります。多数のコンテンツビューを作成することは避けませんが、コンテンツビューのサイズは関連する操作 (公開、プロモート) の速度に影響を与えることに注意してください。また、コンテンツビューのパッケージのサブセットを作成する際には、すべての依存関係も含まれていることを確認してください。キックスタートリポジトリは、ホストのプロビジョニングにのみ使用されるのでコンテンツビューに追加できないことに注意してください。

## 4.5. プロビジョニング

Satellite 6 は、プロビジョニングテンプレート、Puppet を使用した設定管理、ホストロールの標準化されたプロビジョニングのためのホストグループを含む、ホストのプロビジョニングの自動化に役立つ複数の機能を提供します。プロビジョニングのワークフローについては、**the Red Hat Satellite 6 Provisioning Guide** の [Understanding the Provisioning Workflow](#) を参照してください。同じガイドには、各種のコンピュートリソースでのプロビジョニングの方法について記載されています。

## 4.6. ロールベースの認証

ロールをユーザーに割り当てることにより、一連のパーミッションに基づく Satellite 6 コンポーネントへのアクセスを制御できます。ロールベースの認証は、ユーザーにとって不要なオブジェクトをユーザーから非表示にする方法として理解することができます。

組織内の複数の異なるロールを区別する基準は多岐に及びます。管理者ロールのほかにも、一般的に以下のタイプが見られます。

- ※ **アプリケーションまたはインフラストラクチャーの一部に関連するロール** たとえば、オペレーティングシステムとしての Red Hat Enterprise Linux の所有者とアプリケーションサーバーおよびデータベースサーバーの所有者を比較できます。
- ※ **ソフトウェアライフサイクルの特定のステージに関連するロール** たとえば、開発、テストおよび本番フェーズで区分されたロールで、各フェーズに 1 人以上の所有者が設定されるケース。
- ※ **特定のタスクに関連するロール** セキュリティーマネージャー、ライセンスマネージャー、または Access Insights 管理者など。

カスタムロールを定義する際に、以下の推奨事項を考慮してください。

- ※ **予想されるタスクおよび責任の定義** 所定のロールからアクセスできる Satellite インフラストラクチャーのサブセットおよびこのサブセットで許可されるアクションを定義します。ロールの責任について、また他のロールとの違いについて検討します。
- ※ **事前に定義されたロールの使用 (可能な場合)** Satellite 6 は、単独またはロールの組み合わせの一部として使用できる数多くのサンプルロールを提供します。既存のロールのコピーおよび編集からカスタムロールの作成を開始すると良いでしょう。
- ※ **影響を受けるすべてのエンティティーを考慮に入れる** たとえば、コンテンツビューのプロモートにより、特定のライフサイクル環境およびコンテンツビューの組み合わせに対する新規の Puppet 環境が自動的に作成されます。そのため、あるロールがコンテンツビューをプロモートすることが予想される場合に、Puppet 環境を作成し、編集するパーミッションも必要になります。
- ※ **関連分野を考慮に入れる** ロールの責任範囲が限定されている場合でも、関連する分野は広い範囲に及ぶ場合があります。そのため、ロールの責任範囲に影響を与える Satellite インフラストラクチャーの部分に対する読み取り専用アクセスをロールに付与することができます。これにより、ユーザーは今後の変更の可能性についての情報に早期に取得することができます。

- ※ **パーミッションを段階的に追加する:** カスタムロールが予定通りに機能することを確認するためにテストします。問題が生じた場合には、限定されたパーミッションセットでこれを開始し、パーミッションを段階的に追加して継続的にテストするのが効果的な方法です。

ロールを定義し、これらをユーザーに割り当てる方法については、[Red Hat Satellite サーバー管理ガイド](#)の説明を参照してください。同ガイドには、外部の認証ソースについての情報が記載されています。

## 4.7. 追加のタスク

このセクションでは、特定のタスクを自動化したり、Satellite 6 のコアの使用を拡張したりするために使用できる選択された Satellite 機能の短い概要を示します。

- ※ **既存ホストのインポート:** 過去に Satellite 6 で管理されてこなかった既存のホストがある場合、それらのホストを Satellite Server にインポートできます。通常、この手順には Red Hat Satellite 5 からの移行における 1 つのステップが必要になります。詳細のドキュメントについては、[Red Hat Satellite Transition Guide](#)を参照してください。移行プロセスの概要については、Red Hat ナレッジベースのアーティクル [Red Hat Satellite 5 から Satellite 6 への移行](#)を参照してください。
- ※ **ベアメタルホストの検出:** Satellite 6 検出プラグインはプロビジョニングネットワーク上の不明なホストのベアメタルの自動検出を可能にします。これらの新規ホストは、シリアル ID、ネットワークインターフェース、メモリー、およびディスク情報などの Facter が収集するクライアントアップロードシステムのファクトに基づいて Satellite Server および Puppet エージェントに自己登録します。登録後は、それらの検出されたホストのプロビジョニングを初期化できます。詳細については、[Red Hat Satellite Host Configuration Guide](#)の [Discovering Bare-metal Hosts on Satellite](#)を参照してください。
- ※ **バックアップ管理:** Satellite Server のバックアップおよび障害回復の手順についての情報を参照してください ([Red Hat Satellite サーバー管理ガイド](#)の [バックアップおよび障害回復](#)を参照)。リモート実行を利用すると、管理対象ホストで繰り返されるバックアップタスクを設定することもできます。リモート実行についての詳細は、[Red Hat Satellite Host Configuration Guide](#)の [Running Jobs on Satellite Hosts](#)を参照してください。
- ※ **セキュリティー管理:** Satellite 6 は、更新およびエラータ管理、システム検証のための OpenSCA 統合、更新およびセキュリティーコンプライアンスのレポート作成、および詳細なロールベースの認証を含む、セキュリティー管理を各種の方法でサポートします。エラータ管理および OpenSCAP 概念の詳細については、[Red Hat Satellite Host Configuration Guide](#)を参照してください。
- ※ **インシデント管理:** Satellite 6 は、レポート作成およびメール通知を含むすべてのシステムの一元化された概要を提供することで、インシデント管理プロセスをサポートします。各ホストの詳細情報は、最近の変更のイベント履歴を含め、Satellite Server からアクセスできます。また、Satellite 6 は [Red Hat Insights](#)と統合しています。
- ※ **Hammer および API を使用したスクリプト:** Satellite 6 は、大半の Web UI の手順と同等の CLI を提供する Hammer というコマンドラインツールを提供します。さらに、Satellite API へのアクセスを使用して、選択したプログラミング言語で自動化スクリプトを作成できます。詳細は、[Red Hat Hammer CLI Guide](#) および [Red Hat API Guide](#)を参照してください。

## 第5章 共通のデプロイメントシナリオ

このセクションでは、Red Hat Satellite の共通のデプロイメントシナリオの概要を簡潔に説明します。以下のレイアウトに関連して数多くのバリエーションや組み合わせが可能である点に注意してください。

### 5.1. 単一ロケーション

**統合 Capsule** は、インストールプロセス時に Satellite Server でデフォルトで作成される仮想の Capsule Server です。これは、Satellite Server を、単一の地理的な場所の Satellite デプロイメント用の直接接続されたホストのプロビジョニングに使用できることを意味するため、1つの物理サーバーのみが必要になります。分離した Capsule のベースシステムは Satellite Server で直接管理できますが、このレイアウトを使用してリモート拠点にある他のホストを管理することは推奨されません。

### 5.2. サブネットが分類されている単一ロケーション

Red Hat Satellite が単一の地理的な場所にデプロイされている場合でも、インフラストラクチャーに複数の分離したサブネットが必要になる場合があります。この分離は、たとえば DHCP および DN サービスが設定された複数の Capsule Server をデプロイすることによって実行できますが、単一の Capsule を使用して分離したサブネットを作成することが推奨されます。次にこの Capsule を使用して分離されたネットワークでホストおよびコンピュータリソースを管理し、それらが Capsule にアクセスすればプロビジョニング、設定、エラータ、およびその他の一般的な管理ができるようにします。サブネットの設定についての詳細は、[Red Hat Satellite Host Configuration Guide](#) を参照してください。

### 5.3. 複数ロケーション

地理的な拠点ごとに 1 つ以上の Capsule Server を作成することが推奨されます。この方法により、ホストはローカルの Capsule Server からコンテンツを取得するので、帯域幅を節約できます。リモートリポジトリからのコンテンツの同期は、各拠点の各ホストによってではなく Capsule によってのみ実行されます。さらに、このレイアウトにより、プロビジョニングインフラストラクチャーがより信頼性があり、設定しやすいものになります。このアプローチを示した図については、[図1.1 「Red Hat Satellite 6 システムアーキテクチャー」](#) を参照してください。

### 5.4. 接続されていない SATELLITE

セキュリティレベルの高い環境ではインターネットから切断された、閉じられたネットワークでホストを機能させることが必要になりますが、Red Hat Satellite は、システムに対して最新のセキュリティ更新、エラータ、パッケージおよびその他のコンテンツをプロビジョニングできます。この場合、Satellite Server にはインターネットへの直接のアクセスはありませんが、他のインフラストラクチャーコンポーネントのレイアウトは影響を受けません。切断された Satellite のインストールまたはアップグレードを実行する方法についての詳細は、[Red Hat Satellite インストールガイド](#) を参照してください。

切断された Satellite Server にコンテンツをインポートする方法として、2つのオプションがあります。

- ✦ **切断された Satellite とコンテンツ ISO** このセットアップでは、Red hat カスタマーポータルからコンテンツと共に ISO イメージをダウンロードし、それらを Satellite Server またはローカル web サーバーに抽出します。その後 Satellite Server のコンテンツはローカルに同期されます。これにより、Satellite Server のネットワークの完全な分離が可能になりますが、コンテン



ツ ISO イメージのリリースは約 6 週間ごとに行われており、すべての製品コンテンツが組み込まれる訳ではありません (現在は、Red Hat Enterprise Linux と RHEL-OSP7、RHDS、および Red Hat Enterprise Linux for Real Time などのレイヤー化された製品のみ)。コンテンツ ISO を切断された Satellite にインポートする方法についての詳細は、**Red Hat Satellite Content Management Guide** の [Importing Content ISOs into a Disconnected Satellite](#) を参照してください。

- ✳ **切断された Satellite と Satellite 間の同期** このセットアップでは、接続された Satellite Server をインストールし、そこからコンテンツをエクスポートして、ストレージデバイスを使用して切断された Satellite にコンテンツを設定します。これにより、Red Hat が提供するコンテンツとカスタムコンテンツの両方を、各自が選択する頻度でエクスポートできますが、別のサブスクリプションで追加のサーバーをデプロイする必要があります。Satellite 間の同期を設定する方法については、**Red Hat Satellite Content Management Guide** の [Synchronizing Content Between Satellite Servers](#) を参照してください。

コンテンツを切断された Satellite Server にインポートするための上記の方法を使用して、接続された Satellite の初期設定を加速することもできます。

## 5.5. CAPSULE と外部サービス

外部 DNS、DHCP、または TFTP サービスを使用できるように Capsule Server (統合またはスタンドアロン) を設定できます。これらのサービスを提供するサーバーが環境内にすでに存在する場合は、それを Satellite デプロイメントに簡単に統合できます。外部サービスを使用できるように Capsule を設定する方法についての詳細は、**Red Hat Satellite インストールガイド** の [外部サービスの設定](#) を参照してください。

## 付録A SATELLITE で提供される必要になるテクニカルユーザー

Satellite のインストール時にシステムアカウントが作成されます。それらは、ファイルや Satellite に統合されるコンポーネントのプロセス所有権を管理するために使用されます。これらのアカウントの一部には固定された UID があり、その他のアカウントは単純にシステム上で次に利用可能な UID を取得します。各種のアカウントに割り当てられた UID を管理するために、それらのアカウントを事前に定義し UID を固定することができます。アカウントの一部にはハードコーディングされた UID があるため、Satellite のインストール時に作成されるすべてのアカウントについてこれを実行することはできません。

以下の表は、インストール時に Satellite で作成されるすべてのアカウントの概要を示しています。**flex UID** とマークが付けられたアカウントは、Satellite のインストール前にカスタム UID を使って事前に定義することができます。

home または shell などのフィールドは Satellite が正常に機能するための要件になるため、UID 以外に指定されたアカウントのパラメーターや値を変更することは推奨されません。

表A.1 Satellite で提供される必要になるテクニカルユーザー

ユーザー名	UID	Flex UID	Home	Shell
qpidd	なし	はい	/var/lib/qpidd	/sbin/nologin
foreman	なし	はい	/usr/share/foreman	/sbin/nologin
unbound	なし	はい	/etc/unbound	/sbin/nologin
foreman-proxy	なし	はい	/usr/share/foreman-proxy	/sbin/nologin
qdrouterd	なし	はい	/var/lib/qdrouterd	/sbin/nologin
puppet	52	いいえ	/var/lib/puppet	/sbin/nologin
postgres	26	いいえ	/var/lib/pgsql	/bin/bash
mongodb	184	いいえ	/var/lib/mongodb	/sbin/nologin

ユーザー名	UID	Flex UID	Home	Shell
apache	48	いいえ	/usr/share/httpd	/sbin/nologin
hsqldb	96	いいえ	/var/lib/hsqldb	/sbin/nologin
tomcat	91	いいえ	/usr/share/tomcat	/bin/nologin
saslauth	76	はい	/run/saslauthd	/sbin/nologin

## 付録B 用語集

この用語集では、Red Hat Satellite 6 に関連する各種の用語について記載します。

### Activation Key (アクティベーションキー)

ホストの登録およびサブスクリプションの割り当てに使用するトークンです。アクティベーションキーは、新規に作成されるホストに関連付けられるサブスクリプション、製品、コンテンツビュー、およびその他のパラメーターを定義します。

### Answer File (Answer ファイル)

インストールシナリオの設定を定義する設定ファイルです。Answer ファイルは YAML 形式で定義され、`/etc/foreman-installer/scenarios.d/` ディレクトリに保管されます。

### ARF Report (ARF レポート)

OpenSCAP 監査の結果です。Red Hat Satellite で管理されるホストのセキュリティーコンプライアンスについての概要を示すものです。

### Audit (監査)

特定のユーザーが加える変更についてのレポートを提供します。監査は Satellite web UI の **モニター > 監査** で確認できます。

### Baseboard Management Controller (BMC) (ベースボード管理コントローラー: BMC)

ベアメタルホストのリモートの電源管理を有効にします。Satellite 6 では、選択したホストを管理するための BMC インターフェースを作成できます。

### Boot Disk (ブートディスク)

PXE なしのプロビジョニングに使用される ISO イメージです。この ISO により、ホストは Satellite Server に接続でき、インストールメディアを起動し、オペレーティングシステムをインストールできます。ブートディスクには、**ホストイメージ**、**完全ホストイメージ**、**汎用イメージ**、および **サブネットイメージ**などのいくつかの種類があります。

### Capsule (Capsule Server)

コンテンツのフェデレーションおよび配信を容易にする (Pulp ノードとしての機能する) ために、またローカライズされた他のサービス (Puppet マスター、**DHCP**、**DNS**、**TFTP** など) を実行するために Red Hat Satellite 6 デプロイメントで利用できる追加サーバーです。Capsule は、複数の地理的な場所での Satellite のデプロイメントに役立ちます。アップストリームの Foreman 用語では、Capsule は Smart Proxy (スマートプロキシ) と呼ばれています。

### Catalog (カタログ)

Puppet が管理する 1 つの特定のホストのあるべきシステム状態について説明するドキュメントです。これは、管理する必要のあるすべてのリソースと、それらリソース間の依存関係を一覧表示します。カタログは Puppet マニフェストから Puppet マスターによってコンパイルされ、また Puppet エージェントのデータが使用されます。

### Candlepin

サブスクリプションの管理を行う Katello 内のサービスです。



**Compliance Policy (コンプライアンスポリシー)**

SCAP コンテンツに対するコンプライアンスチェックを指定されたホストに実行する、Satellite Server で実行されるスケジュールされたタスクを指します。

**Compute Profile (コンピュートプロファイル)**

コンピュートリソース上で新規仮想マシンのデフォルトの属性を指定します。

**Compute Resource (コンピュートリソース)**

Red Hat Satellite 6 がホストやシステムのデプロイに使用する仮想またはクラウドのインフラストラクチャーです。例として、Red Hat Enterprise Virtualization、OpenStack、EC2、および VMWare などがあります。

**Container (Docker Container) (コンテナ (Docker コンテナ))**

アプリケーションで必要とされるすべてのランタイム依存関係が含まれる分離されたアプリケーションサンドボックスです。Satellite 6 は、専用のコンピュートリソースでコンテナのプロビジョニングをサポートします。

**Container Image (コンテナイメージ)**

コンテナの設定の静的なスナップショットです。Satellite 6 は、コンテンツビューでイメージをホストに配信すると共にコンテナイメージをインポートするために各種の方法をサポートします。

**Content (コンテンツ)**

Satellite がホストに配信するものの総称です。ソフトウェアパッケージ (RPM ファイル)、Puppet モジュール、Docker イメージ、または OSTree スナップショットが含まれます。コンテンツはライブラリーに同期され、コンテンツビューを使ってライフサイクル環境にプロモートされ、ホストでコンテンツを使用できるようにします。

**Content Delivery Network (CDN) (コンテンツ配信ネットワーク: CDN)**

Red Hat コンテンツを Satellite Server に配信するために使用されるメカニズムです。

**Content Host (コンテンツホスト)**

コンテンツとサブスクリプションに関連するタスクを管理するホストの一部です。

**Content View (コンテンツビュー)**

インテリジェントなフィルタリングによって作成されるライブラリーコンテンツのサブセットです。コンテンツビューが公開されると、ライフサイクル環境パスでプロモートしたり、増分アップグレードを使用して変更したりすることができます。コンテンツビューは、Red Hat Satellite 5 のチャンネルとクローン作成の組み合わせをさらに改良したものです。

**Discovered Host (検出ホスト)**

検出プラグインによってプロビジョニングネットワークで検出されるベアメタルホストです。

**Discovery Image (検出イメージ)**

プロビジョニングプロセスの開始前に、初期のハードウェア情報を取得し、Satellite

Server と通信するためにホスト上で PXE で起動した Red Hat Enterprise Linux ベースの最小オペレーティングシステムを指します。

### Discovery Plug-in (検出プラグイン)

プロビジョニングネットワーク上の不明なホストのベアメタルの自動検出を可能にします。このプラグインは、Satellite Server および Capsule Server で実行されるサービス、およびホスト上で実行される Discovery イメージの 3 つのコンポーネントで構成されます。

### Discovery Rule (検出ルール)

ホストグループを検出されたホストに割り当て、プロビジョニングを自動的にトリガーする事前に定義されたプロビジョニングルールのセットです。

### Docker Tag (Docker タグ)

コンテナイメージを、通常はイメージに保存されたアプリケーションのバージョンで区別するために使用されるマークです。Satellite 6 web UI の **コンテンツ > Docker Tags** 以下でタグ別にイメージをフィルターできます。

### ERB

Embedded Ruby (ERB) は、プロビジョニングおよびジョブテンプレートで使用されるテンプレートの構文です。

### Errata (エラータ)

セキュリティ修正、バグ修正、および機能拡張を含む更新された RPM パッケージです。ホストとの関連では、エラータはホストにインストールされているパッケージを更新する場合にのみ **適用可能** となり、ホストのコンテンツビューにある場合にのみ **インストール可能** になります (つまり、ホストでのインストールのためにアクセス可能になります)。

### External Node Classifier (外部ノードの分類子)

ホストの設定時に Puppet マスターが使用する追加データを提供する Puppet コンストラクトです。Red Hat Satellite 6 は、Satellite デプロイメントで Puppet マスターに対する外部ノードの分類子として機能します。

### Facter

Facter は一種のプログラムであり、それが実行されるシステムについての情報 (ファクト) を提供します。たとえば、Facter は合計メモリ、オペレーティングシステムのバージョン、アーキテクチャーなどをレポートすることができます。Puppet モジュールは、Facter によって収集されるホストデータに基づいて特定の設定を有効にします。

### Fact (ファクト)

合計メモリ、オペレーティングシステムのバージョン、またはアーキテクチャーなどのホストのパラメーターです。ファクトは Facter によってレポートされ、Puppet で使用されます。

### Foreman

主にプロビジョニングおよびコンテンツのライフサイクル管理を行う Red Hat Satellite 6 コンポーネントです。Foreman は Red Hat Satellite 6 に対応する主なアップストリームのコンポーネントです。

## Foreman Hook

ホストの作成時やホストのプロビジョニングの完了時など、オーケストレーションイベントが発生する際に自動的にトリガーされる実行可能プログラムです。

## Full Host Image (完全ホストイメージ)

特定のホストの PXE なしのプロビジョニングに使用されるブートディスクです。この完全なホストイメージには、組み込み Linux カーネルと、関連付けられたオペレーティングシステムインストーラーの init RAM ディスクが含まれます。

## Generic Image (汎用イメージ)

特定ホストに関連付けられていない PXE なしのプロビジョニングのブートディスクです。この汎用イメージは、ホストの MAC アドレスを Satellite Server に送信し、これをホストエントリーに対してマッチングします。

## Hammer

Red Hat Satellite 6 を管理するためにコマンドラインツールです。コマンドラインから Hammer コマンドを実行したり、それらをスクリプトで実行したりできます。また、Hammer は対話式シェルも提供します。

## Host (ホスト)

Red Hat Satellite 6 が管理するすべてのシステム (物理または仮想システム) を指します。

## Host Collection (ホストコレクション)

エラータのインストールなど、一括動作に使用される 1 つ以上のホストのユーザー定義グループです。Satellite 5 システムグループに相当します。

## Host Group (ホストグループ)

ホストをビルドするためのテンプレートです。ホストグループは、ホストグループメンバーによって継承されるサブネットやライフサイクル環境などの共有パラメーターを保持します。ホストグループはネスト化して階層構造を作成できます。

## Host Image (ホストイメージ)

特定ホストの PXE なしのプロビジョニングに使用されるブートディスクです。ホストイメージには、Satellite Server のインストールメディアにアクセスするため必要なブートファイルのみが含まれます。

## Incremental Upgrade (of a Content View)((コンテンツビューの) 増分アップグレード)

ライフサイクル環境に新規 (マイナー) コンテンツビューバージョンを作成する動作です。増分アップグレードにより、すでに公開されたコンテンツビューのインプレースの変更を行うことが可能になります。セキュリティエラータの適用時など、迅速な更新に役立ちます。

## Job (ジョブ)

Satellite Server からホスト上でリモートに実行されるコマンドです。すべてのジョブはテンプレートで定義されます。Satellite 5 のリモートコマンドに似ています。

## Job Template (ジョブテンプレート)

ジョブのプロパティを定義します。

## Katello

サブスクリプションおよびリポジトリ管理を行う Foreman プラグインです。

## Lazy Sync (遅延同期)

**即時** という yum リポジトリのデフォルトのダウンロードポリシーを**オンデマンド** または **バックグラウンド** に変更する機能です。オンデマンド設定は、クライアントによる要求時にのみパッケージをダウンロードすることでスペースと同期時間を節約し、**バックグラウンド** 設定は、リポジトリのメタデータの同期後にパッケージをダウンロードして同期時間を節約します。

## Location (ロケーション)

物理的な場所を表すデフォルト設定のコレクションです。

## Library (ライブラリー)

Satellite Server のすべての同期済みリポジトリのコンテンツのコンテナです。各組織にデフォルトで存在する主なライフサイクル環境、すべてのライフサイクル環境パスのルート、およびすべてのコンテンツビューのコンテンツのソースです。

## Life Cycle Environment (ライフサイクル環境)

コンテンツホストによって消費されるコンテンツビューのバージョンのコンテナです。ライフサイクル環境はライフサイクル環境パスのステップを表します。コンテンツはコンテンツビューを公開し、プロモートすることによりライフサイクル環境を通過します。

## Life Cycle Environment Path (ライフサイクル環境パス)

コンテンツビューがプロモートされるライフサイクル環境の順序です。たとえば、開発からテスト、また本番へなど、コンテンツビューをプロモートパスでプロモートすることができます。Red Hat Satellite 5 では、チャンネルのクローン作成がこの概念を実行するものとなりました。

## Manifest (Subscription Manifest)(マニフェスト (サブスクリプションマニフェスト))

Red Hat カスタマーポータルから Red Hat Satellite 6 にサブスクリプションを送信するメカニズムです。これは Red Hat Satellite 5 で使用される証明書に機能的に似ています。

## OpenSCAP

Security Content Automation Protocol (SCAP) に基づいてセキュリティーコンプライアンスの監査を実施するプロジェクトです。OpenSCAP は、管理対象ホストのコンプライアンス監査を実行するために Satellite 6 に統合されています。

## Organization (組織)

Satellite 6 デプロイメント内の複数のシステム、コンテンツおよびその他の機能からなる単独のコレクションです。

## OSTree

起動可能で不変なバージョン管理されたシステムツリーを管理するためのツールです。Satellite 6 は OSTree スナップショットのミラーリングと、それらのコンテンツビューでの配信をサポートします。

### Parameter (パラメーター)

プロビジョニング時の Red Hat Satellite コンポーネントの動作を定義します。パラメーターの範囲に応じて、グローバル、ドメイン、ホストグループ、およびホストパラメーターの間で区別します。パラメーターの複雑度にもよりますが、単純なパラメーター(キーと値のペア) およびスマート変数(条件付き引数、検証、オーバーライド) 間で区別できます。

### Parametrized Class (Smart Class Parameter) (パラメーター化されたクラス (Smart Class パラメーター))

Puppet マスターからクラスをインポートして作成されるパラメーターです。

### Permission (パーミッション)

Satellite インフラストラクチャーの選択された部分(リソースタイプ)に関連するアクションを定義します。各リソースタイプはパーミッションのセットに関連付けられます。たとえば、**Architecture** というリソースタイプには以下のパーミッションがあります:  
**view\_architectures**、**create\_architectures**、**edit\_architectures**、および  
**destroy\_architectures**。パーミッションはロールに分類し、それらをユーザーまたはユーザーグループに関連付けることができます。

### Product (製品)

コンテンツリポジトリのコレクションです。製品は Red Hat CDN から提供されるか、または Satellite 管理者によってカスタムリポジトリを分類するために作成されます。

### Promote (a Content View)((コンテンツビューの) 公開)

1つのライフサイクル環境から別のライフサイクル環境へとコンテンツを移行する動作を指します。

### Provisioning Template (プロビジョニングテンプレート)

ホストプロビジョニングの設定を定義します。プロビジョニングテンプレートはホストグループ、ライフサイクル環境、またはオペレーティングシステムに関連付けることができます。Satellite 6 では、Red Hat Satellite 5 のキックスタープロファイルおよび Cobbler スニペットと同様の機能を提供します。

### Publish (a Content View)((コンテンツビューの) 公開)

コンテンツビューのバージョンをライフサイクル環境で利用可能にし、ホストが利用できるようにする動作です。

### Pulp

リポジトリおよびコンテンツ管理を行う Katello 内のサービスです。

### Pulp Node (Pulp ノード)

コンテンツをミラーリングする Capsule Server のコンポーネントです。これは Red Hat Satellite 5 Proxy に似ています。主な違いは、Pulp ノードの場合には、コンテンツがホストによって使用される前にそのノード上でコンテンツをステージングできる点にあります。

## **Puppet**

Satellite 6 の設定および管理コンポーネントです。

## **Puppet Agent (Puppet エージェント)**

設定変更をホストに適用するホスト上で実行されるサービスです。

## **Puppet Environment (Puppet 環境)**

Puppet モジュールの特定のセットに関連付けることのできる Puppet エージェントノードの単独のセットです。

## **Puppet Manifest (Puppet マニフェスト)**

.pp 拡張子を使用したファイル名を持つ Puppet スクリプトを指します。

## **Puppet Master (Puppet マスター)**

Puppet エージェントが実行する Puppet マニフェストをホストに提供する Capsule Server のコンポーネントです。

## **Puppet Module (Puppet モジュール)**

ユーザー、ファイル、およびサービスなどのリソースを管理するために利用できるコード (Puppet マニフェスト) およびデータ (ファクト) の自己完結型のバンドルです。

## **Recurring Logic (再帰論理)**

スケジュールに従って自動的に実行されるジョブです。Satellite 6 web UI では、これらのジョブを **モニター > 再帰論理** で確認できます。

## **Registry (レジストリー)**

コンテナイメージのアーカイブです。Satellite 6 は、ローカルおよび外部のレジストリーからのイメージのインポートをサポートします。Satellite 自体はホストのイメージレジストリーとして機能できます。ただし、ホストは変更をレジストリーに戻すことができません。

## **Red Hat Access Insights**

Satellite web UI から選択された Red Hat カスタマーポータルサービスへのアクセスを提供するモジュールです。

## **Repository (リポジトリ)**

コンテンツのコレクションのストレージを提供します。

## **Resource Type (リソースタイプ)**

ホスト、Capsule、またはアーキテクチャーなどの Satellite インフラストラクチャーの一部を指します。パーミッションのフィルターで 사용됩니다。

## Role (ロール)

ホストなどのリソースのセットに適用されるパーミッションのコレクションを指します。ロールはユーザーおよびユーザーグループに割り当てることができます。Satellite は数多くの事前に定義されたロールを提供します。

## SCAP content (SCAP コンテンツ)

ホストのチェックに使用される設定およびセキュリティのベースラインを含むファイルです。コンプライアンスポリシーで使用されます。

## Scenario (シナリオ)

Satellite CLI インストーラーの事前に定義された設定のセットです。シナリオでは、インストールのタイプを定義します。たとえば、Capsule Server をインストールするには、**satellite-installer --scenario capsule** を実行します。すべてのシナリオには、シナリオの設定を保存するための独自の Answer ファイルがあります。

## Smart Proxy (スマートプロキシ)

DNS または DHCP などの外部サービスと統合できる Capsule Server です。アップストリームの Foreman の用語では、Smart Proxy (スマートプロキシ) は Capsule の同意語です。

## Smart Variable (Smart 変数)

Puppet モジュールのクラスで使用される設定値です。

## Standard Operating Environment (SOE) (標準運用環境 (SOE))

アプリケーションがデプロイされるオペレーティングシステムの制御されたバージョンです。

## Subnet Image (サブネットイメージ)

Capsule Server 経由で通信する PXE なしのプロビジョニングの汎用イメージのタイプです。

## Subscription (サブスクリプション)

Red Hat からコンテンツおよびサービスを受信するためのエンタイトルメントです。

## Synchronization (同期)

外部リソースのコンテンツを Red Hat Satellite 6 ライブラリーにミラーリングすることを指します。

## Synchronization Plan (同期プラン)

コンテンツ同期のスケジュールされた実行を可能にします。

## Task (タスク)

リポジトリの同期またはコンテンツビューの公開など、Satellite または Capsule Server で実行されるバックグラウンドプロセスです。タスクステータスは、Satellite web UI の **モニター > タスク** でモニターできます。

## Trend (トレンド)

Satellite 6 インフラストラクチャーの特定の部分で変更を追跡する手段です。トレンドを、Satellite web UI の **モニター** > **トレンド** で設定します。

## User Group (ユーザーグループ)

ユーザーのコレクションに割り当てることのできるロールのコレクションです。これは、Red Hat Satellite 5 のロールに似ています。

## User (ユーザー)

Red Hat Satellite を使用できるように登録されたすべてのユーザーを指します。認証および認可については、外部リソース (LDAP、Identity Management、または Active Directory) または Kerberos を使ってビルトインロジックで実行できます。

## virt-who

ハイパーバイザーから仮想マシンの ID を取得するためのエージェントです。Satellite 6 で使用すると、virt-who はそれらの ID を Satellite Server に報告するため、仮想マシンにプロビジョニングされているホストのサブスクリプションを提供できます。