



Red Hat Satellite 6.15

概要、概念、デプロイメントに関する考慮事項

Satellite アーキテクチャーの理解と Satellite デプロイメントの計画

Red Hat Satellite 6.15 概要、概念、デプロイメントに関する考慮事項

Satellite アーキテクチャーの理解と Satellite デプロイメントの計画

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

本書では、Red Hat Satellite 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関する推奨事項を説明します。

目次

多様性を受け入れるオープンソースの強化	4
RED HAT ドキュメントへのフィードバック (英語のみ)	5
パート I. SATELLITE の概要と概念	6
第1章 RED HAT SATELLITE によるコンテンツとパッチの管理	7
1.1. RED HAT SATELLITE のコンテンツフロー	7
1.2. RED HAT SATELLITE のコンテンツビュー	8
1.3. RED HAT SATELLITE のコンテンツタイプ	10
1.4. 関連情報	10
第2章 RED HAT SATELLITE によるサブスクリプション管理	11
2.1. RED HAT SATELLITE の SIMPLE CONTENT ACCESS (SCA)	11
2.2. 関連情報	11
第3章 RED HAT SATELLITE によるプロビジョニング管理	12
3.1. RED HAT SATELLITE のプロビジョニング方法	12
3.2. 関連情報	12
第4章 SATELLITE の主要コンポーネント	13
4.1. SATELLITE サーバーの概要	13
4.2. RED HAT SATELLITE の組織と場所	13
4.3. カプセルの概要	14
4.4. SATELLITE のホストの概要	14
4.5. SATELLITE SERVER の主要なオープンソースコンポーネントのリスト	14
4.6. カプセルの特徴	14
4.7. カプセルネットワーク	16
4.8. 関連情報	18
第5章 RED HAT SATELLITE の管理ツール	19
5.1. SATELLITE ウェブ UI の概要	19
5.2. HAMMER CLI の概要	19
5.3. SATELLITEAPI の概要	20
5.4. RED HAT SATELLITE でのリモート実行	20
5.5. ANSIBLE コレクションを使用した SATELLITE の管理	21
5.6. キックスタートワークフロー	21
第6章 サポートされている SATELLITE コンポーネントの使用法とバージョン	23
6.1. SATELLITE コンポーネントのサポートされている使用法	23
6.2. コンテンツ管理でサポートされているクライアントアーキテクチャー	24
6.3. ホストプロビジョニングでサポートされているクライアントアーキテクチャー	24
6.4. 設定管理でサポートされているクライアントアーキテクチャー	24
6.5. 関連情報	25
パート II. SATELLITE デプロイメント計画	26
第7章 一般的なデプロイメントシナリオ	27
7.1. 単一の場所	27
7.2. 分離されたサブネットを持つ単一の場所	27
7.3. 複数の場所	27
7.4. オフラインの SATELLITE	28
7.5. 外部サービスを備えたカプセル	29

第8章 デプロイメントに関する考慮事項	30
8.1. SATELLITE サーバーの設定	30
8.2. 外部データベースを備えた SATELLITE サーバー	31
8.3. 場所とトポロジー	32
8.4. コンテンツソース	32
8.5. コンテンツのライフサイクル	33
8.6. コンテンツのデプロイメント	35
8.7. プロビジョニング	35
8.8. ロールベースの認証	35
8.9. 追加タスク	36
第9章 組織、場所、ライフサイクル環境	37
9.1. 組織	37
9.2. ロケーション	38
9.3. ライフサイクル環境	38
第10章 ホストグループ化の概念	39
10.1. ホストグループの構造	39
第11章 プロビジョニングの概念	41
11.1. PXE ブート	41
11.2. HTTP ブート	42
付録A SATELLITE によって提供され、必要とされる技術ユーザー	45
付録B SATELLITE で使用される用語集	47
付録C CLI ヘルプ	54

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。この取り組みは膨大な作業を要するため、これらの変更による更新は可能な範囲で段階的に行われます。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

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。

パート I. SATELLITE の概要と概念

Red Hat Satellite は、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、および監視を行う集中型ツールです。Satellite を使用すると、物理環境、仮想環境、クラウド環境にわたってシステムを展開、設定、保守できます。

第1章 RED HAT SATELLITE によるコンテンツとパッチの管理

Red Hat Satellite を使用すると、ライフサイクルのすべての段階で体系的にコンテンツを提供し、ホストにパッチを適用できます。

1.1. RED HAT SATELLITE のコンテンツフロー

Red Hat Satellite のコンテンツフローには、外部ソースからホストへのコンテンツの管理と配信が含まれます。

Satellite 内のコンテンツは、**外部コンテンツソース** から **Satellite Server** に流れます。**Capsule Server** は、Satellite Server のコンテンツを **ホスト** にミラーリングします。

外部コンテンツソース

Satellite では多くのコンテンツソースを設定できます。サポートされているコンテンツソースには、Red Hat カスタマーポータル、Git リポジトリ、Ansible コレクション、Docker Hub、SCAP リポジトリ、または組織の内部データストアが含まれます。

Satellite Server

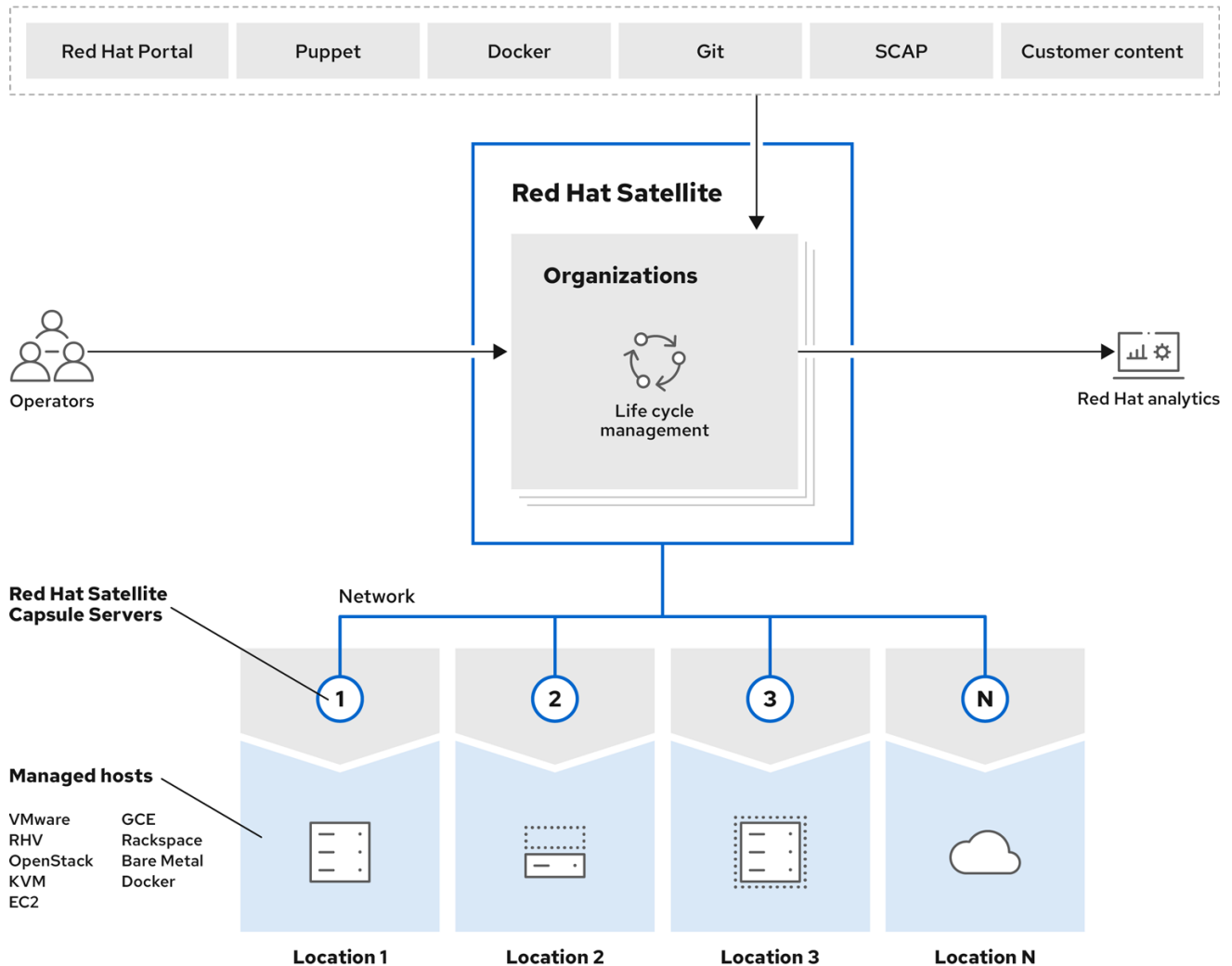
Satellite Server では、コンテンツのライフサイクルを計画および管理します。

Capsule Servers

Capsule Server を作成することで、ニーズに応じてさまざまな場所にコンテンツソースを確立できます。たとえば、地理的な場所ごとにコンテンツソースを確立したり、個別のネットワークを持つデータセンターに複数のコンテンツソースを確立したりできます。

Hosts

ホストシステムを Capsule Server に割り当てるか、Satellite Server に直接割り当てることで、ホストが提供するコンテンツをホストが確実に受信できるようになります。ホストは物理サーバーでも仮想サーバーでもかまいません。



278_Satellite_1022

関連情報

- 見る[4章Satellite の主要コンポーネント](#)詳細は。
- コンテンツ配信ネットワーク (CDN) の詳細は、[コンテンツの管理の Red Hat サブスクリプションの管理](#)を参照してください。

1.2. RED HAT SATELLITE のコンテンツビュー

コンテンツビューは、ホストがアクセスできるコンテンツの意図的にキュレーションされたサブセットです。コンテンツビューを作成することで、特定の環境または Capsule Server で使用されるソフトウェアバージョンを定義できます。

各コンテンツビューは、各環境にわたってリポジトリのセットを作成します。これらのリポジトリは Satellite Server によって保存および管理されます。たとえば、次の方法でコンテンツビューを作成できます。

- 実稼働環境用の古いパッケージバージョンを含むコンテンツビューと、**開発** 環境用の新しいパッケージバージョンを含む別のコンテンツビュー。
- オペレーティングシステムに必要なパッケージリポジトリを含むコンテンツビューと、アプリケーションに必要なパッケージリポジトリを含む別のコンテンツビュー。

- コンテンツビューを管理するためのモジュール方式の複合コンテンツビュー。たとえば、オペレーティングシステムを管理するためのコンテンツには1つのコンテンツビューを使用し、アプリケーションを管理するためのコンテンツには別のコンテンツビューを使用できます。両方のコンテンツビューを組み合わせた複合コンテンツビューを作成することにより、各コンテンツビューのリポジトリをマージする新しいリポジトリが作成されます。ただし、コンテンツビューのリポジトリは引き続き存在するため、個別に管理し続けることもできます。

デフォルトの組織ビュー

デフォルトの組織ビューは、Satellite に同期されるすべてのコンテンツに対するアプリケーション制御のコンテンツビューです。デフォルトの組織ビューを使用すると、コンテンツビューやライフサイクル環境を操作せずに、サブスクリプションを使用してホストを Satellite に登録し、コンテンツにアクセスできます。

環境間でコンテンツビューを促進する

アプリケーションライフサイクルでコンテンツビューをある環境から次の環境に昇格すると、Satellite はリポジトリを更新し、パッケージを公開します。

例1.1 パッケージを開発 からテスト へ昇格する

テスト および 本番 用のリポジトリには、`<my_software> -1.0-0.noarch.rpm` パッケージが含まれています。

	開発	テスト	実稼働
コンテンツビューのバージョン	バージョン 2	バージョン 1	バージョン 1
コンテンツビューの内容	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.0-0.noarch.rpm</code>	<code><my_software>-0-0.noarch.rpm</code>

コンテンツビューのバージョン 2 を 開発 から テスト に昇格すると、テスト のリポジトリが更新され、`<my_software> -1.1-0.noarch.rpm` パッケージが含まれるようになります。

	開発	テスト	実稼働
コンテンツビューのバージョン	バージョン 2	バージョン 2	バージョン 1
コンテンツビューの内容	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.1-0.noarch.rpm</code>	<code><my_software>-1.0-0.noarch.rpm</code>

これにより、ホストは特定の環境に指定されますが、その環境でコンテンツビューの新しいバージョンが使用されると更新を受け取るようになります。

関連情報

- 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。

1.3. RED HAT SATELLITE のコンテンツタイプ

Red Hat Satellite を使用すると、さまざまなコンテンツタイプをインポートして管理できます。

たとえば、Satellite は次のコンテンツタイプをサポートしています。

RPM パッケージ

Red Hat サブスクリプションに関連するリポジトリから RPM パッケージをインポートします。Satellite Server は、Red Hat Content Delivery Network から RPM ファイルをダウンロードし、ローカルに保存します。これらのリポジトリとその RPM ファイルをコンテンツビューで使用できません。

キックスタートツリー

キックスタートツリーをインポートしてホストをプロビジョニングします。新しいシステムは、ネットワーク経由でこれらの Kickstart ツリーにアクセスし、インストールの基本コンテンツとして使用します。Red Hat Satellite には、事前定義された Kickstart テンプレートが含まれています。独自の Kickstart テンプレートを作成することもできます。

ISO および KVM イメージ

インストールおよびプロビジョニングのメディアをダウンロードして管理します。たとえば、Satellite は特定の Red Hat Enterprise Linux および Red Hat 以外のオペレーティングシステムの ISO イメージとゲストイメージをダウンロード、保存、管理します。

カスタムファイルタイプ

SSL 証明書、ISO イメージ、OVAL ファイルなど、必要なあらゆる種類のファイルのカスタムコンテンツを管理します。

1.4. 関連情報

- Satellite を使用してコンテンツを管理する方法については、[コンテンツの管理](#) を参照してください。

第2章 RED HAT SATELLITE によるサブスクリプション管理

Red Hat Satellite を使用すると、ソフトウェアサブスクリプションの使用状況を追跡し、ニーズに応じてサブスクリプションを割り当てることができます。

2.1. RED HAT SATELLITE の SIMPLE CONTENT ACCESS (SCA)

Satellite の Simple Content Access (SCA) により、ソフトウェアエンタイトルメントの管理が簡素化されます。SCA を使用すると、有効なサブスクリプションをサブスクリプション割り当てまたはマニフェストに追加し、Satellite 内で更新できます。したがって、SCA を有効にすると、ホストにサブスクリプションを個別にアタッチする必要がなくなります。



注記

Simple Content Access (SCA) は、以前のエンタイトルメントベースのサブスクリプション管理方法に代わるものです。新しい組織では、デフォルトで SCA が有効になります。エンタイトルメントベースのサブスクリプション管理は非推奨です。詳細は、[リリースノート](#) を参照してください。

関連情報

- SCA の詳細は、[Simple Content Access - FAQ](#) を参照してください。
- SCA の有効化と使用の詳細は、[Simple Content Access](#) を参照してください。

2.2. 関連情報

- Satellite でサブスクリプションを管理する方法の詳細は、[コンテンツの管理](#) を参照してください。

第3章 RED HAT SATELLITE によるプロビジョニング管理

Red Hat Satellite を使用すると、統合されたインターフェイスからさまざまなプロビジョニング方法を使用して、さまざまなコンピューティングリソース上にホストをプロビジョニングできます。

3.1. RED HAT SATELLITE のプロビジョニング方法

Red Hat Satellite を使用すると、複数の方法を使用してホストをプロビジョニングできます。

ベアメタルプロビジョニング

Satellite は、主に PXE ブートと MAC アドレス識別を使用してベアメタルホストをプロビジョニングします。Satellite を使用してベアメタルホストをプロビジョニングする場合、次の操作を実行できます。

- ホストエントリーを作成し、プロビジョニングする物理ホストの MAC アドレスを指定します。
- プロビジョニング可能なホストのプールを作成する Satellite Discovery サービスを使用するには、空のホストを起動します。

Cloud Providers

Satellite はプライベートクラウドプロバイダーとパブリッククラウドプロバイダーに接続し、クラウド環境に保存されているイメージからホストのインスタンスをプロビジョニングします。Satellite を使用してクラウドからプロビジョニングする場合、次の操作を実行できます。

- 使用するハードウェアプロファイルまたはフレーバーを選択します。
- 特定のプロバイダーの API を使用してクラウドインスタンスをプロビジョニングします。

仮想化インフラストラクチャー

Satellite は、Red Hat Virtualization や VMware などの仮想化インフラストラクチャーサービスに接続します。Satellite を使用して仮想マシンをプロビジョニングする場合、次の操作を実行できます。

- 仮想イメージテンプレートから仮想マシンをプロビジョニングします。
- ベアメタルプロバイダーと同じ PXE ベースのブート方法を使用します。

3.2. 関連情報

- Satellite を使用してホストをプロビジョニングする方法については、[ホストのプロビジョニング](#)を参照してください。

第4章 SATELLITE の主要コンポーネント

一般的な Satellite デプロイメントは、Satellite Server、Satellite Server からのコンテンツをミラーリングする Capsule Server、および Satellite Server と Capsule Server からコンテンツと設定を受信するホストというコンポーネントで設定されます。

4.1. SATELLITE サーバーの概要

Satellite Server は、コンテンツのライフサイクルを計画および管理する Satellite デプロイメントの中心となるコンポーネントです。

一般的な Satellite デプロイメントには、次の操作を実行する 1 つの Satellite Server が含まれます。

- コンテンツライフサイクル管理
- カプセルサーバーの設定
- ホストの設定
- ホストのプロビジョニング
- パッチ管理
- サブスクリプションの管理

Satellite Server 委譲コンテンツ配信、ホストのプロビジョニング、Capsule Server への通信。Satellite Server 自体にも Capsule が含まれています。

Satellite Server には、きめ細かい認証システムも含まれています。Satellite ユーザーに、担当するインフラストラクチャーの部分のみにアクセスするための権限を付与できます。

関連情報

- 権限の管理の詳細は、[Red Hat Satellite の管理の ユーザーとロールの管理](#) を参照してください。

4.2. RED HAT SATELLITE の組織と場所

Satellite Server では、コンテンツ、ホスト、および設定を整理するために複数の **組織** と **場所** を定義できます。

組織

組織は通常、**財務**、**マーケティング**、**Web 開発** などのさまざまな事業単位、部門、またはチームを表します。

組織を作成することで、特定の要件に応じて設定を個別に分離して管理するための論理コンテナを作成できます。

場所

場所は通常、国や都市などの物理的な場所を表します。

場所を作成することで、ホストが配置されている地理的なサイトを定義できます。たとえば、複数のデータセンターがある環境ではこれが役立ちます。

4.3. カプセルの概要

Capsule Server を使用すると、Satellite デプロイメントの範囲とスケーラビリティを拡張できます。Capsule Server は、Red Hat Satellite デプロイメントで次の機能を提供します。

- Satellite Server からコンテンツをミラーリングして、さまざまな地理的または論理的な場所にコンテンツソースを確立します。ホストを Capsule Server に登録することで、このホストが中央の Satellite Server からではなく、その場所にある Capsule からコンテンツと設定を受信するように設定できます。
- ホストを検出、プロビジョニング、制御、および設定するためのローカライズされたサービスを実行します。

コンテンツビューを使用すると、Capsule Server がホストに提供するコンテンツの正確なサブセットを指定できます。詳細は、[1章 Red Hat Satellite によるコンテンツとパッチの管理](#) を参照してください。

4.4. SATELLITE のホストの概要

ホストは、Red Hat Satellite が管理する Linux クライアントを指します。ホストは物理サーバーでも仮想サーバーでもかまいません。

Amazon EC2、Google Compute Engine、KVM、libvirt、Microsoft Azure、OpenStack、Red Hat Virtualization、Rackspace Cloud Services、VMware vSphere など、Red Hat Satellite でサポートされている任意のプラットフォームに仮想ホストをデプロイできます。

Satellite を使用すると、監視、プロビジョニング、リモート実行、設定管理、ソフトウェア管理、サブスクリプション管理など、ホストを大規模に管理できます。

4.5. SATELLITE SERVER の主要なオープンソースコンポーネントのリスト

Satellite は、次のような相互に統合された複数のオープンソースプロジェクトで設定されています。

Foreman

Foreman は、物理システムと仮想システムのライフサイクル管理アプリケーションです。プロビジョニングと設定からオーケストレーションと監視まで、ライフサイクル全体にわたってホストを管理するのに役立ちます。

Katello

Katello は、コンテンツ、サブスクリプション、リポジトリ管理の追加機能によって Foreman の機能を拡張する Foreman のプラグインです。Katello により、Satellite 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できるようになります。

Candlepin

Candlepin はサブスクリプション管理サービスです。

Pulp

Pulp はリポジトリとコンテンツを管理するためのサービスです。

関連情報

- Satellite に統合されているアップストリームコンポーネントの完全なリストと、さまざまなバージョンの Satellite で提供されたアップストリームコンポーネントのバージョンに関する情報については、[Satellite 6 コンポーネントバージョン](#) を参照してください。

4.6. カプセルの特徴

Capsule Server はローカルホスト管理サービスを提供し、Satellite Server からのコンテンツをミラーリングできます。

Satellite Server のコンテンツをミラーリングするために、Capsule は次の機能を提供します。

リポジトリの同期

Capsule Server は、選択されたライフサイクル環境のコンテンツを Satellite Server からプルし、管理するホストでこのコンテンツを利用できるようにします。

コンテンツ配信

Capsule Server を使用するよう設定されたホストは、Satellite Server ではなく、その Capsule からコンテンツをダウンロードします。

ホストアクション配信

Capsule Server はホスト上でスケジュールされたアクションを実行します。

Red Hat Subscription Management (RHSM) プロキシ

ホストは、中央の Satellite Server や Red Hat カスタマーポータルではなく、関連付けられている Capsule Server に登録されます。

Capsule を使用して、インフラストラクチャーとホスト管理用の次のサービスを実行できます。

DHCP

Capsule は、ISC DHCP サーバー、Active Directory、Libvirt インスタンスなどの既存のソリューションとの統合を含め、DHCP サーバーを管理できます。

DNS

Capsule は、ISC バインドや Active Directory などの既存のソリューションとの統合を含め、DNS サーバーを管理できます。

TFTP

Capsule は、任意の 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できるため、プロビジョニング中にホストが自動的に参加できるようになります。Capsule は、Red Hat Identity Management および Active Directory などの既存のインフラストラクチャーと統合できます。

パペットサーバー

Capsule は、Puppet サーバーを実行することで設定管理サーバーとして機能できます。

パペット認証局

Capsule は Puppet 証明機関 (CA) と統合して、ホストに証明書を提供できます。

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

Capsule は、Intelligent Platform Management Interface (IPMI) または Redfish 標準を使用して、ホストの電源管理を提供できます。

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 はホスト上で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。

リモート実行 (REX)

Capsule はホスト上でリモートジョブ実行を実行できます。

Capsule で選択した機能のみを有効にすることで、Capsule Server を特定の限定された目的のために設定できます。一般的な設定は次のとおりです。

インフラストラクチャーカプセル: DNS + DHCP + TFTP

これらのサービスを備えたカプセルは、ホストにインフラストラクチャーサービスを提供し、新しいホストをプロビジョニングするために必要なすべてのサービスを備えています。

コンテンツカプセル: パルプ

このサービスを備えたカプセルは、Satellite Server からホストに同期されたコンテンツを提供します。

設定カプセル: Pulp + Puppet + PuppetCA

これらのサービスを備えたカプセルは、ホストにコンテンツと実行設定サービスを提供します。

DNS + DHCP + TFTP + Pulp + Puppet + PuppetCA を備えたカプセル

これらのサービスを備えたカプセルは、カプセル機能の完全なセットを提供します。これらすべての機能を備えたカプセルを設定すると、ホストに単一の接続ポイントを提供することで、そのカプセルに割り当てられたホストを分離できます。

4.7. カプセルネットワーク

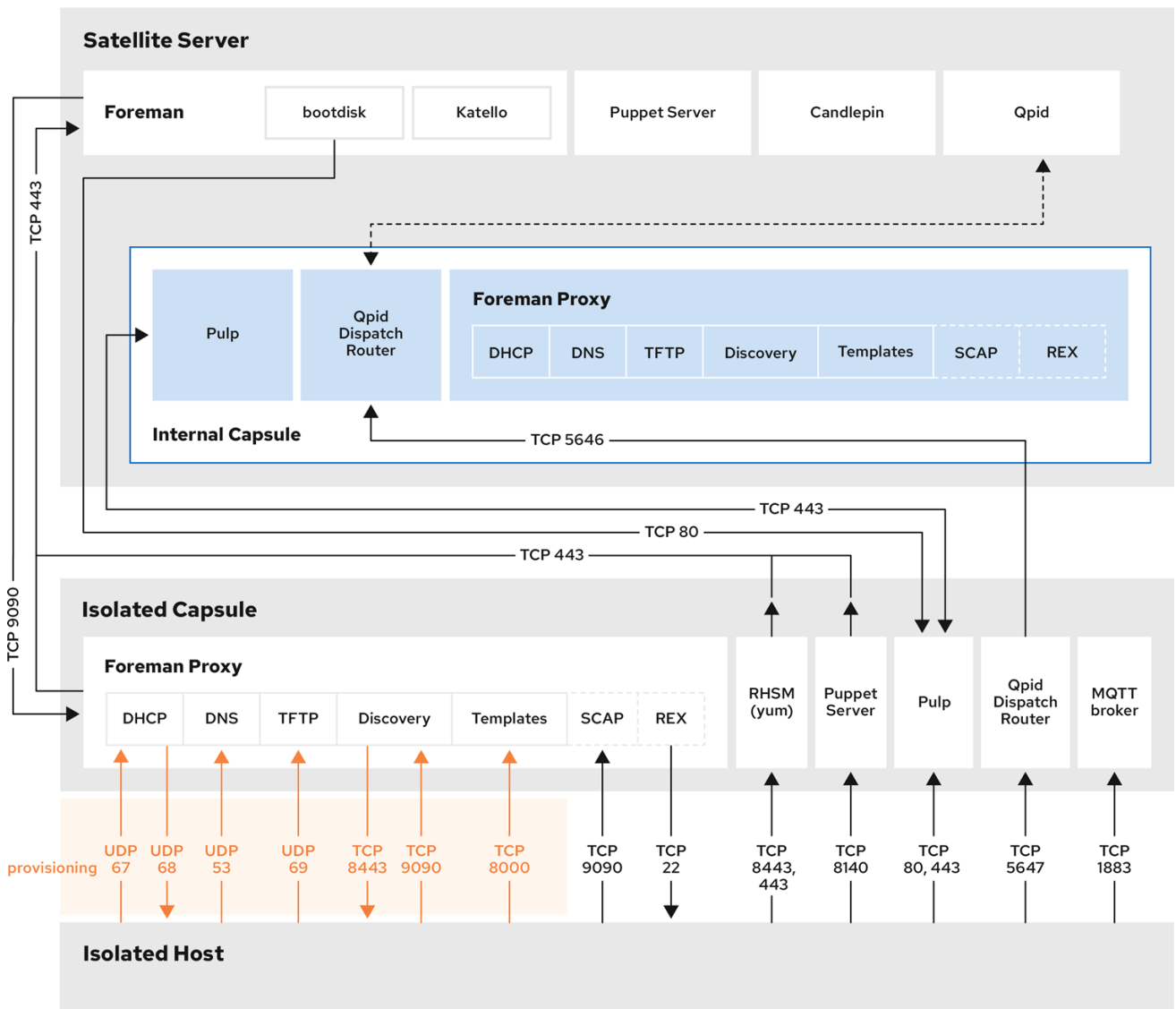
Satellite Server と Capsule Server に登録されたホスト間の通信は、その Capsule Server を介してルーティングされます。Capsule Server はホストに Satellite サービスも提供します。

Capsule Server が管理するサービスの多くは、専用のネットワークポートを使用します。ただし、Capsule Server は、ホストから Satellite Server へのすべての通信が単一のソース IP アドレスを使用するようにし、ファイアウォールの管理を簡素化します。

ホストがカプセルに接続する Satellite トポロジー

このトポロジーでは、Capsule はすべてのホストネットワーク通信に対して単一のエンドポイントを提供するため、リモートネットワークセグメントでは、Capsule 自体へのファイアウォールポートのみを開く必要があります。

図4.1 ホストがカプセルに接続したときに Satellite コンポーネントがどのように相互作用するか

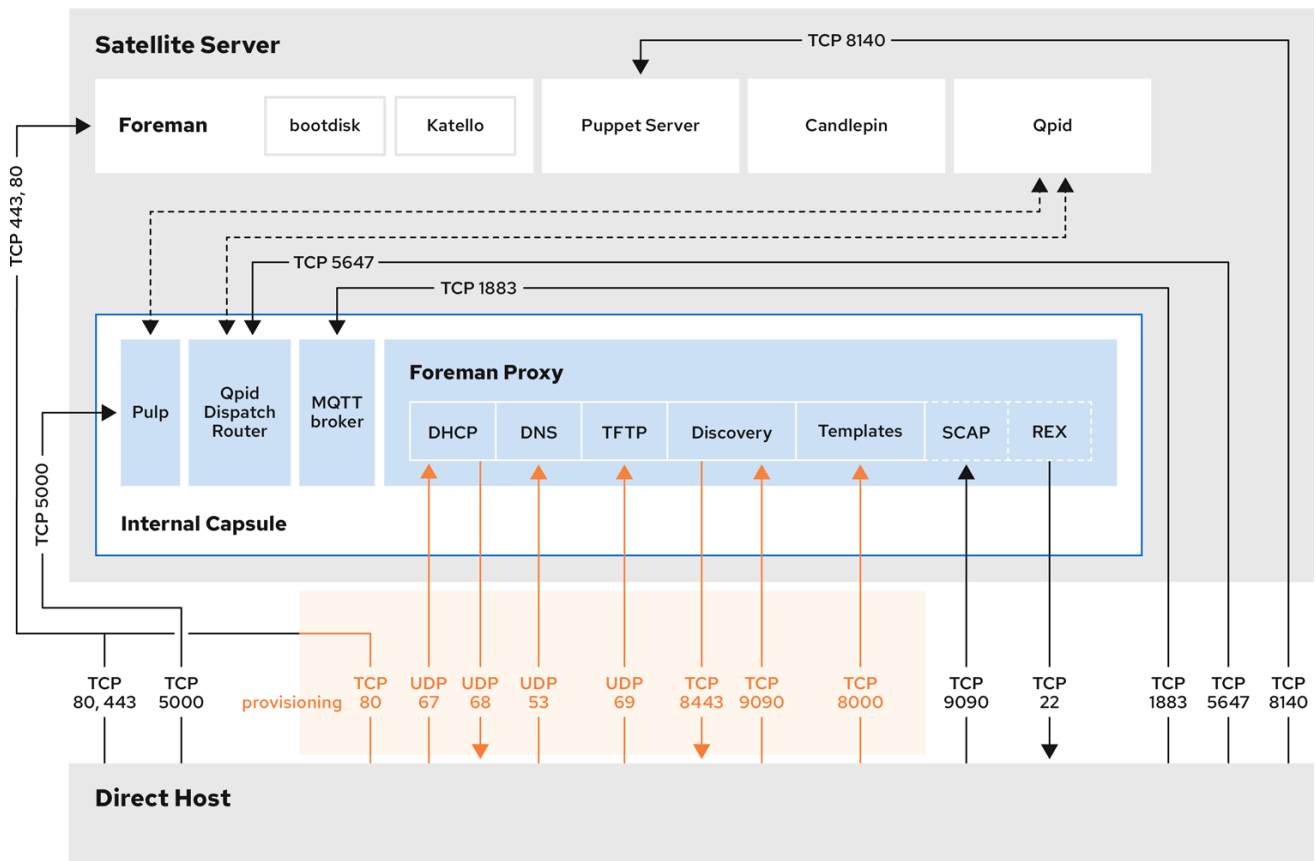


278_Satellite_0922

ホストが直接 Satellite Server に接続する Satellite トポロジー

このトポロジーでは、ホストは Capsule ではなく Satellite Server に接続します。Capsule Server は Satellite Server のホストであるため、これは Capsule 自体にも適用されます。

図4.2 ホストが Satellite Server に直接接続する場合の Satellite コンポーネントの相互作用



278_Satellite_0922

関連情報

必要なポートを開くようにホストベースのファイアウォールを設定するための全手順は、次のドキュメントを参照してください。

- [接続されたネットワーク環境に Satellite Server をインストールする場合のポートとファイアウォールの要件](#)
- [切断されたネットワーク環境での Satellite Server のインストールにおけるポートとファイアウォールの要件](#)
- [Capsule Server のインストールにおけるポートとファイアウォールの要件](#)

4.8. 関連情報

- Capsule Server の要件、インストール、およびスケーラビリティに関する考慮事項の詳細は、[Capsule Server のインストール](#)を参照してください。
- Capsule Server 間で負荷を分散する方法の詳細は、[ロードバランサーを使用した Capsule の設定](#)を参照してください。

第5章 RED HAT SATELLITE の管理ツール

Red Hat Satellite を管理するには複数のツールを使用できます。

5.1. SATELLITE ウェブ UI の概要

Satellite Web UI を使用して、ブラウザから Satellite インフラストラクチャーを管理および監視できます。たとえば、Satellite Web UI では次のナビゲーション機能を使用できます。

ナビゲーション機能	説明
組織 ドロップダウン	管理する組織を選択します。
場所 ドロップダウン	管理する場所を選択します。
Monitor	サマリーダッシュボードおよびレポートを表示します。
コンテンツ	コンテンツ管理ツールを提供します。これには、コンテンツビュー、アクティベーションキー、ライフサイクル環境が含まれます。
Hosts	ホストインベントリおよびプロビジョニング設定ツールを提供します。
設定	ホストグループや Ansible コンテンツなどの一般的な設定ツールとデータを提供します。
インフラストラクチャー	Satellite が環境と対話する方法を設定するツールを提供します。
	環境に対する重要な変更が管理者に通知されるようにイベントの通知が表示されません。
管理	ユーザー、ロールベースのアクセス制御 (RBAC)、一般設定などの設定の高度な設定を提供します。

関連情報

- Satellite Web UI の使用に関する詳細は、[Red Hat Satellite の管理](#)を参照してください。

5.2. HAMMER CLI の概要

Hammer を使用して CLI コマンドで Satellite Server を設定および管理できます。

Hammer を使用すると、次のような利点があります。

- 基本的なタスクの自動化のために、Hammer コマンドに基づいてシェルスクリプトを作成します。
- Hammer からの出力を他のツールにリダイレクトします。

- スクリプトで API 呼び出しを適用する前に、Hammer で **--debug** オプションを使用して API 呼び出しへの応答をテストします。たとえば、**hammer --debug organization list** です。

Hammer コマンドを発行するには、ユーザーが Satellite Server にアクセスできる必要があります。



注記

ユーザーフレンドリーで直感的なエクスペリエンスを実現するために、新しい機能を開発する際には Satellite Web UI が優先されます。したがって、Satellite Web UI で利用できる機能の一部は、Hammer ではまだ利用できない可能性があります。

バックグラウンドで、各 Hammer コマンドは最初に API へのバインドを確立し、要求を送信します。この動作は、大量の Hammer コマンドを順に実行する場合に、パフォーマンスに影響を与える可能性があります。対照的に、API コマンドを使用するスクリプトは Satellite API と直接通信し、バインドを 1 回だけ確立します。

関連情報

- Hammer CLI の使用に関する詳細は、[Hammer CLI ガイド](#)を参照してください。

5.3. SATELLITEAPI の概要

Satellite Server が提供する Representational State Transfer (REST) API を使用して、HTTP 経由で Satellite API にアクセスするカスタムスクリプトや外部アプリケーションを作成できます。REST API を使用して、企業の IT システムやサードパーティーのアプリケーションと統合し、自動メンテナンスやエラーチェックタスクを実行し、スクリプトを使用して反復タスクを自動化します。

REST API を使用すると、次のような利点があります。

- API を使用するには、HTTP プロトコルをサポートする任意のプログラミング言語、フレームワーク、またはシステムを設定します。
- ユーザーは実行時に多くの詳細を発見するため、Satellite インフラストラクチャーに関する最小限の知識しか必要としないクライアントアプリケーションを作成します。
- 仮想化プラットフォームを直感的に管理するために、リソースベースの REST モデルを採用します。

API コマンドに基づくスクリプトは Satellite API と直接通信するため、redhat.satellite 内のモジュールに依存する Hammer コマンドや Ansible Playbook に基づくスクリプトよりも高速になります。



重要

API コマンドは、Satellite のバージョンごとに異なります。Satellite Server のアップグレードを準備するときは、Satellite API コマンドを含むすべてのスクリプトを更新します。

関連情報

- Satellite API の使用に関する詳細は、[API ガイド](#)を参照してください。

5.4. RED HAT SATELLITE でのリモート実行

リモート実行を使用すると、シェルスクリプトまたは Ansible タスクと Playbook を使用して、Capsule からホスト上でジョブをリモートで実行できます。

Satellite でリモート実行を使用すると、次の利点が得られます。

- 複数のホストでジョブを同時に実行します。
- 実行するジョブをより細かく制御するには、コマンドで変数を使用します。
- ホストファクトとパラメーターを使用して変数値を入力します。
- コマンドを実行するときに、テンプレートのカスタム値を指定します。

リモート実行の通信は Capsule Server を介して行われるため、Satellite Server はターゲットホストに直接アクセスする必要がなく、多数のホストを管理するように拡張できます。

リモート実行を使用するには、ジョブテンプレートを定義する必要があります。ジョブテンプレートは、リモートホストに適用するコマンドです。ジョブテンプレートは複数回実行できます。

Satellite は、Embedded Ruby (ERB) 構文ジョブテンプレートを使用します。詳細は、[ホストの管理のテンプレート作成リファレンス](#) を参照してください。

デフォルトでは、Satellite にはシェルスクリプトと Ansible 用のジョブテンプレートがいくつか含まれています。詳細は、[ホストの管理のジョブテンプレートの設定](#) を参照してください。

関連情報

- [ホストの管理のリモートジョブの実行](#) を参照してください。
- [Ansible 統合を使用した設定の管理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。

5.5. ANSIBLE コレクションを使用した SATELLITE の管理

Satellite Ansible Collections は、Satellite API と対話する Ansible モジュールセットです。Satellite Ansible コレクションを使用すると、Satellite のさまざまな側面を管理および自動化できます。

関連情報

- [Ansible 統合を使用した設定の管理](#) を参照してください。
- [Red Hat Satellite の管理](#) を参照してください。

5.6. キックスタートワークフロー

インストールに必要なすべての情報を含む Kickstart ファイルを作成することにより、Satellite Server または Capsule Server のインストールプロセスを自動化できます。

Red Hat Satellite Kickstart スクリプトを実行すると、スクリプトは次のアクションを実行します。

1. Satellite Server または Capsule Server のインストールのロケーションを指定します。
2. 事前定義済みのパッケージをインストールします。
3. Subscription Manager をインストールします。

4. アクティベーションキーを使用して、ホストを Red Hat Satellite にサブスクライブします。
5. Puppet をインストールし、**puppet.conf** ファイルを設定して Red Hat Satellite または Capsule インスタンスを指定します。
6. Puppet を有効にして実行し、証明書を要求します。
7. ユーザー定義のスニペットを実行します。

関連情報

キックスタートの詳細は、高度な RHEL 8 インストールの実行の [キックスタートを使用した自動インストールの実行](#) を参照してください。

第6章 サポートされている SATELLITE コンポーネントの使用法とバージョン

Satellite は、次のユースケース、アーキテクチャー、バージョンをサポートしています。

6.1. SATELLITE コンポーネントのサポートされている使用法

すべての Red Hat Satellite コンポーネントの使用は、以下に説明するように、Red Hat Satellite のコンテキスト内でのみサポートされます。

Red Hat Enterprise Linux Server

各 Red Hat Satellite サブスクリプションには、サポートされる Red Hat Enterprise Linux Server 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 を実行する目的のみに予約してください。

サポートされていません: 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。

SELinux

SELinux が強制モードまたは許可モードになっていることを確認します。

サポートされていません: SELinux を無効にしたインストール。

Foreman

Red Hat Satellite にパッケージ化されたプラグインを使用して Foreman を拡張できます。サポートされている Foreman プラグインの詳細は、Red Hat ナレッジベースの [Satellite 6 コンポーネントバージョン](#) を参照してください。

サポートされていません: **Red Hat Satellite オプションリポジトリ**内のプラグインを使用して Foreman を拡張します。

Red Hat Satellite には、Red Hat Enterprise Linux 以外のオペレーティングシステムをプロビジョニングおよび設定するためのコンポーネント、設定、機能も含まれています。これらの機能は含まれていますが、Red Hat は Red Hat Enterprise Linux でのみその使用をサポートしています。

Pulp

Pulp と対話するには、Satellite Web UI、CLI、API のみを使用します。

サポートされていません: Pulp ローカル API またはデータベースとの直接的な変更または対話。これにより、Red Hat Satellite データベースに修復不可能な損傷が発生する可能性があります。

Candlepin

Candlepin と対話するには、Satellite Web UI、CLI、API のみを使用します。

サポートされていません: Candlepin、そのローカル API、またはデータベースとの直接のやり取り。これにより、Red Hat Satellite データベースに修復不可能な損傷が発生する可能性があります。

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と対話するには、Satellite Web UI、API、およびデータベースのみを使用します。

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のローカル API またはデータベースとの直接のやり取り。

Puppet

Satellite インストールプログラムを実行すると、Capsule Server の一部として Puppet サーバーをインストールして設定できます。Satellite Server または任意の Capsule Server 上の Puppet サーバーで実行される Puppet モジュールも、Red Hat によってサポートされています。

関連情報

- Red Hat はさまざまなスクリプトやその他のフレームワークをサポートしています。Red Hat ナレッジベースの [Red Hat はスクリプトフレームワークをどのようにサポートしているか](#) を参照してください。

6.2. コンテンツ管理でサポートされているクライアントアーキテクチャー

Satellite でホストを登録および管理するには、次の Red Hat Enterprise Linux のメジャーバージョンとハードウェアアーキテクチャーの組み合わせを使用できます。これらの組み合わせには、Satellite Client 6 リポジトリも利用できます。

表6.1 コンテンツ管理サポート

プラットフォーム	アーキテクチャー
Red Hat Enterprise Linux 9	x86_64, ppc64le, s390x, aarch64
Red Hat Enterprise Linux 8	x86_64, ppc64le, s390x
Red Hat Enterprise Linux 7	x86_64, ppc64 (BE), ppc64le, aarch64, s390x
Red Hat Enterprise Linux 6	x86_64, i386, s390x, ppc64 (BE)

6.3. ホストプロビジョニングでサポートされているクライアントアーキテクチャー

Satellite を使用したホストのプロビジョニングには、次の Red Hat Enterprise Linux のメジャーバージョンとハードウェアアーキテクチャーの組み合わせを使用できます。

表6.2 ホストプロビジョニングのサポート

プラットフォーム	アーキテクチャー
Red Hat Enterprise Linux 9	x86_64
Red Hat Enterprise Linux 8	x86_64
Red Hat Enterprise Linux 7	x86_64
Red Hat Enterprise Linux 6	x86_64, i386

6.4. 設定管理でサポートされているクライアントアーキテクチャー

Satellite による設定管理には、次の Red Hat Enterprise Linux のメジャーバージョンとハードウェアアーキテクチャーの組み合わせを使用できます。

表6.3 設定管理サポート

プラットフォーム	アーキテクチャー
Red Hat Enterprise Linux 9	x86_64
Red Hat Enterprise Linux 8	x86_64、 aarch64
Red Hat Enterprise Linux 7	x86_64
Red Hat Enterprise Linux 6	x86_64、 i386

6.5. 関連情報

- Red Hat Satellite リリースのサポート期間については、[Red Hat Satellite 製品ライフサイクル](#)を参照してください。

パート II. SATELLITE デプロイメント計画

第7章 一般的なデプロイメントシナリオ

このセクションでは、Red Hat Satellite に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ついては、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。

7.1. 単一の場所

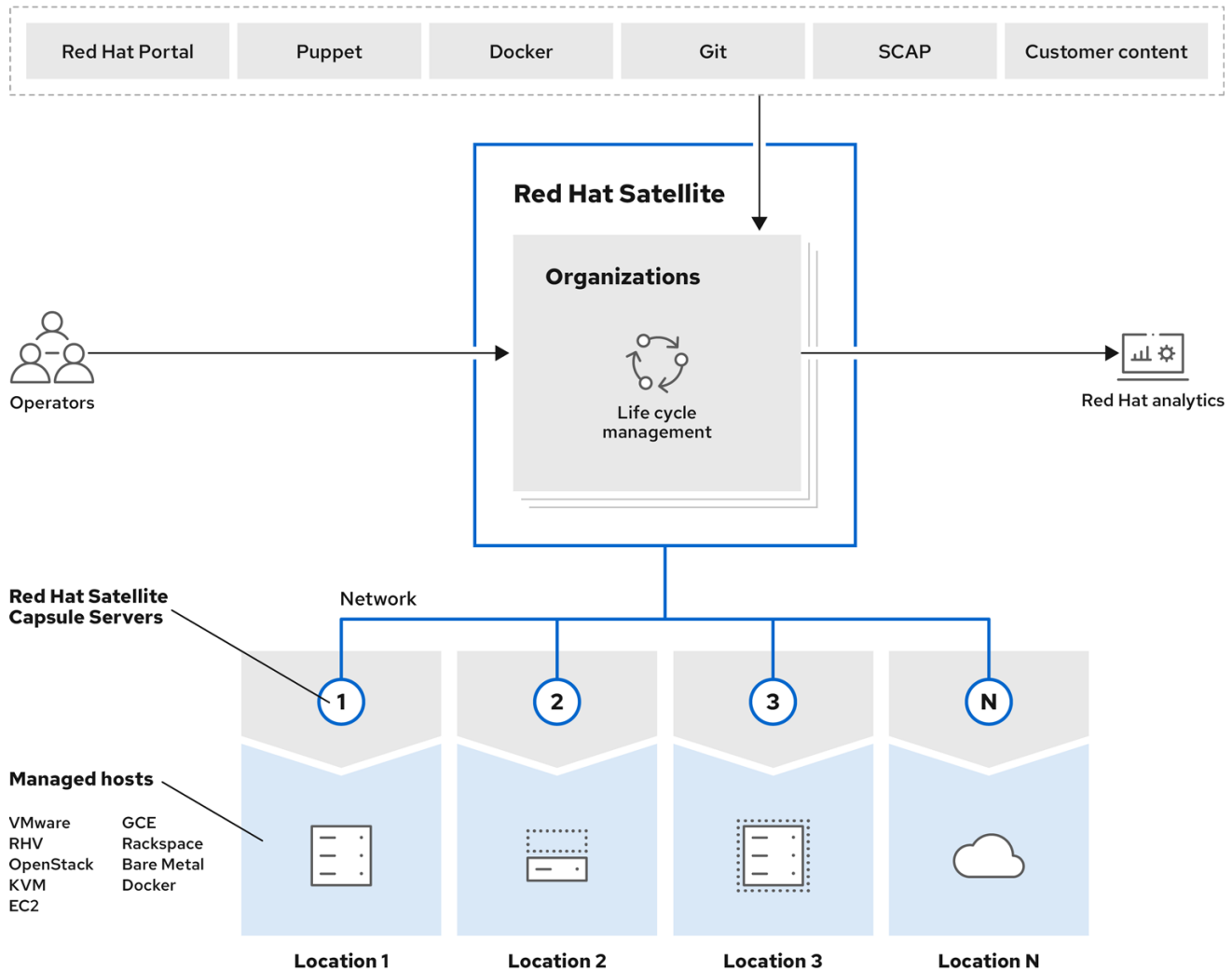
統合 Capsule は、インストールプロセス時に Satellite Server でデフォルトで作成される仮想の Capsule Server です。これは、Satellite Server を、1つの地理的なロケーションにある Satellite デプロイメントに直接接続したホストのプロビジョニングに使用できることを意味するため、必要になるのは物理サーバーが1台のみです。分離した Capsule のベースシステムは Satellite Server で直接管理できますが、このレイアウトを使用してリモート拠点にある他のホストを管理することは推奨されません。

7.2. 分離されたサブネットを持つ単一の場所

Red Hat Satellite が1つの地理的なロケーション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は分離したサブネットが複数必要になる場合があります。これは、たとえば DHCP および DNS サービスが設定されている複数の Capsule Server をデプロイすることで実現できますが、推奨される方法は Capsule を1つ使用して分離されたサブネットを作成することです。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とコンピュータリソースを管理し、Capsule にアクセスするだけでプロビジョニング、設定、エラータ、一般的な管理ができるようにします。サブネットの設定の詳細は、[ホストの管理](#) を参照してください。

7.3. 複数の場所

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。これにより、ホストはローカルの Capsule Server からコンテンツを取得するため、帯域幅を節約できます。リモートリポジトリとのコンテンツの同期は、各拠点の各ホストではなく、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の信頼性が上がり、設定が容易になります。



278_Satellite_1022

7.4. オフラインの SATELLITE

セキュリティー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ー更新、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。切断されたネットワークから Satellite Server をインストールする方法については、[切断されたネットワーク環境での Satellite Server のインストール](#) を参照してください。切断された Satellite のアップグレードについては、[接続された Red Hat Satellite 6.15 へのアップグレードの切断された Satellite Server のアップグレード](#) 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は 2 つあります。

- コンテンツ ISO を含む切断された Satellite**– 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を含む ISO イメージをダウンロードし、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 のコンテンツをローカルに同期します。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。コンテンツ ISO イメージが利用可能なサブスクリプションの製品を確認するには、<https://access.redhat.com> で Red Hat カスタマーポータルにログインし、**Downloads** > **Red Hat Satellite** の順に移動して **Content ISOs** をクリックします。切断された Satellite に

コンテンツ ISO をインポートする方法については、[コンテンツの管理](#)の [コンテンツをローカル CDN サーバーと同期するための Satellite の設定](#) を参照してください。Satellite Server へのインポート用に redhat.com で以前にホストされていたコンテンツ ISO 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。

- **切断された Satellite で、Satellite 間の同期** このセットアップでは、接続する 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、指定した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 で Satellite 間同期を設定する方法については、[コンテンツの管理](#)の [Satellite サーバー間でのコンテンツの同期](#) 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上記の方法を使用して、接続されている Satellite の初期設定処理を加速することもできます。

7.5. 外部サービスを備えたカプセル

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 デプロイメントに統合できます。外部サービスを使用して Capsule を設定する方法については、[Capsule Server のインストール](#)の [外部サービスを使用した Capsule Server の設定](#) を参照してください。

第8章 デプロイメントに関する考慮事項

本セクションでは、Red Hat Satellite 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と推奨事項を説明し、より具体的なドキュメントを紹介します。

8.1. SATELLITE サーバーの設定

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、Satellite Server のインスタンスを専用の Red Hat Enterprise Linux 8 Server にインストールします。

- 接続されたネットワークに Satellite Server をインストールする方法の詳細は、[接続されたネットワーク環境での Satellite Server のインストール](#) を参照してください。
大規模な Satellite のデプロイメントでは、事前定義済みの tuning プロファイルを使用して Satellite を設定するとパフォーマンスを向上できます。詳細は、[接続されたネットワーク環境での Satellite Server のインストールの定義済みプロファイルを使用した Satellite Server の調整](#) を参照してください。
- 切断されたネットワークでの Satellite Server のインストールの詳細は、[切断されたネットワーク環境での Satellite Server のインストール](#) を参照してください。
大規模な Satellite のデプロイメントでは、事前定義済みの tuning プロファイルを使用して Satellite を設定するとパフォーマンスを向上できます。詳細は、[切断されたネットワーク環境での Satellite Server のインストールの定義済みプロファイルを使用した Satellite Server のチューニング](#) を参照してください。

Red Hat サブスクリプションマニフェストを Satellite Server に追加する

Red Hat 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 はこの情報を使用して 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します。Red Hat サブスクリプションマニフェストを作成してインポートする方法については、[コンテンツの管理の Red Hat サブスクリプションの管理](#) を参照してください。

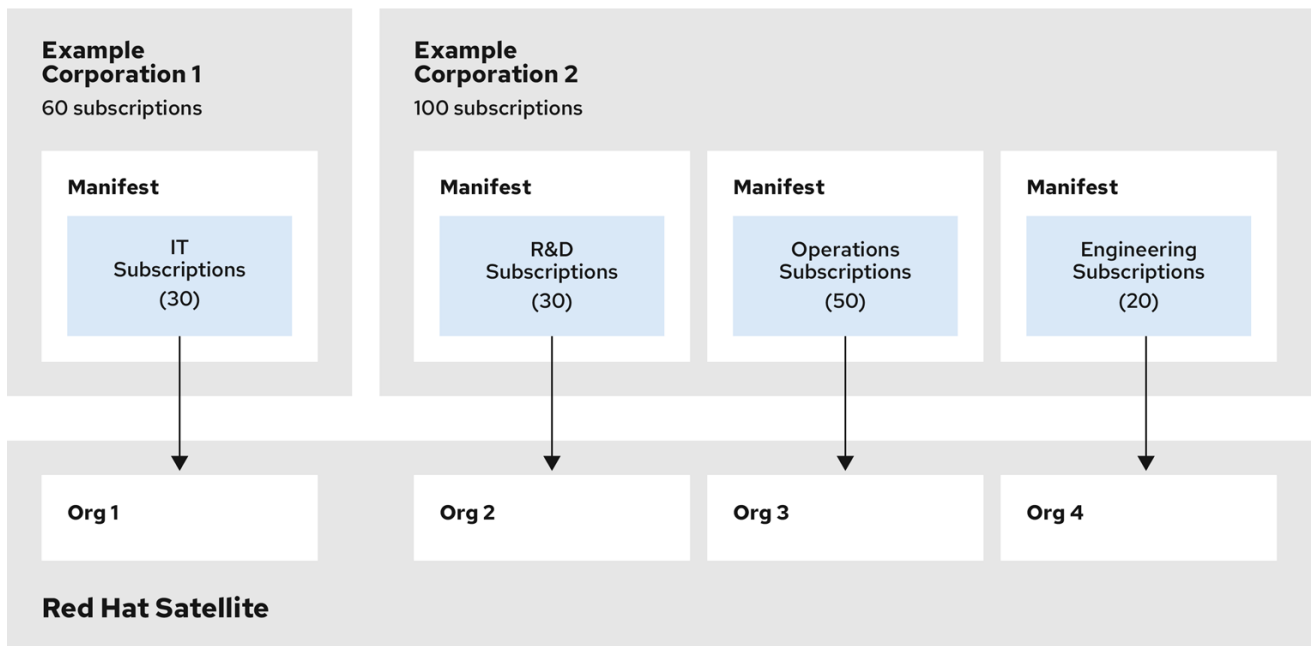
Red Hat Satellite では、Satellite に設定した各組織に対して、マニフェストが1つ必要になります。Satellite の組織機能を使用して、1つの Red Hat Network アカウントで、インフラストラクチャーを分けて管理する予定がある場合は、必要に応じて1つのアカウントが所有するサブスクリプションを、各組織のマニフェストに割り当てます。

Red Hat Network アカウントが複数になる予定がある場合や、Red Hat Network アカウントの所有者でもある別の組織に属するシステムを管理する必要がある場合には、ユーザーとは別のアカウント保持者が、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。Satellite サブスクリプションがなくても有効な別のサブスクリプションを所有している場合は、Subscription Asset Manager マニフェストを作成し、Satellite で使用できます。これにより、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。

システムの管理が必要であるにもかかわらず、RPM のサブスクリプションにアクセスできない場合は、Red Hat Enterprise Linux Satellite アドオンを使用する必要があります。詳細は、[Satellite アドオン](#) を参照してください。

以下の図では、1つの Satellite インストールで複数システムの管理を希望する、2組の 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。このマニフェストは、別の組織として Satellite にインポートできるので、システム管理者は Satellite 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 のシステムを管理できます。

図8.1 複数のマニフェストを持つ Satellite サーバー



278_Satellite_0922

Red Hat Subscription Manifest を作成する場合:

- 切断された Satellite Server または自己登録 Satellite Server を計画する場合は、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、ベースシステムで Subscription Manager ユーティリティを使用してサブスクライブしている、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 に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全 Red Hat 製品にサブスクリプションを追加します。
- サブスクリプションの有効期限が切れる日に注意し、有効期限が切れる前に更新するように計画します。
- 1つの組織につきマニフェストを1つ作成します。マニフェストは複数使用することができ、複数の Red Hat サブスクリプションからマニフェストを選択することができます。

Red Hat Satellite では、未来の日付が指定されたサブスクリプションをマニフェストで使用することができます。これにより、既存のサブスクリプションの有効期限の前に、未来の日付が指定されたサブスクリプションがマニフェストに追加され、継続的にリポジトリへアクセスすることができます。

Red Hat 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 で削除すると、コンテンツホストの登録がすべて解除されます。

8.2. 外部データベースを備えた SATELLITE サーバー

Satellite のインストール時に、**satellite-installer** コマンドで Satellite のインストール先の同じサーバーにデータベースを作成します。要件によっては、外部データベースに移行することで、Satellite の作業メモリーが増え、データベースの操作要求への対応時間が短縮される可能性があります。また、外部データベースに移行することで、負荷を分散し、パフォーマンスチューニングの容量を増やすことができます。

以下のシナリオで Satellite のデプロイメントを使用する予定の場合は、外部データベースの使用をご検討ください。

- リモート実行タスクが頻繁に行われる場合。このような場合には、PostgreSQL に大量のレコードが作成され、データベースの負荷が高くなります。
- リポジトリの同期やコンテンツビューの公開が頻繁に行われ、ディスクの I/O ワークロードが多い場合。このような場合には、Satellite はジョブ毎に PostgreSQL にレコードを作成します。
- ホストの容量が多い場合
- 同期したコンテンツの容量が多い場合

外部データベースの使用に関する詳細は、[接続されたネットワーク環境での Satellite Server のインストールの Satellite での外部データベースの使用](#) を参照してください。

8.3. 場所とトポロジー

このセクションでは、Satellite デプロイメントシナリオを指定する上で役立つ一般的な検討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は、[7章 一般的なデプロイメントシナリオ](#)に記載されています。以下はよくある質問です。

- **必要な Capsule Server の数は？** 組織が運営される地理的なロケーションの数が 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1つのロケーションのデプロイメントで使用でき、Capsule Server のベースシステムのプロビジョニングに使用できます。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 管理者のマネージド外である場合があります。このような場合には、Capsule を外部サービスと統合する必要があります (「[外部サービスを備えたカプセル](#)」を参照)。
- **Satellite Server をインターネットから切断している必要はありますか？** 切断された Satellite は一般的なデプロイメントシナリオです (「[オフラインの 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、Inter-Satellite Synchronization 向けの追加の 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トリソースは何ですか？** - ベアメタルホストのプロビジョニングとは別に、Satellite でサポートされているさまざまなコンピュートリソースを使用できます。さまざまなコンピュートリソースのプロビジョニングの詳細は、[ホストのプロビジョニング](#) を参照してください。

8.4. コンテンツソース

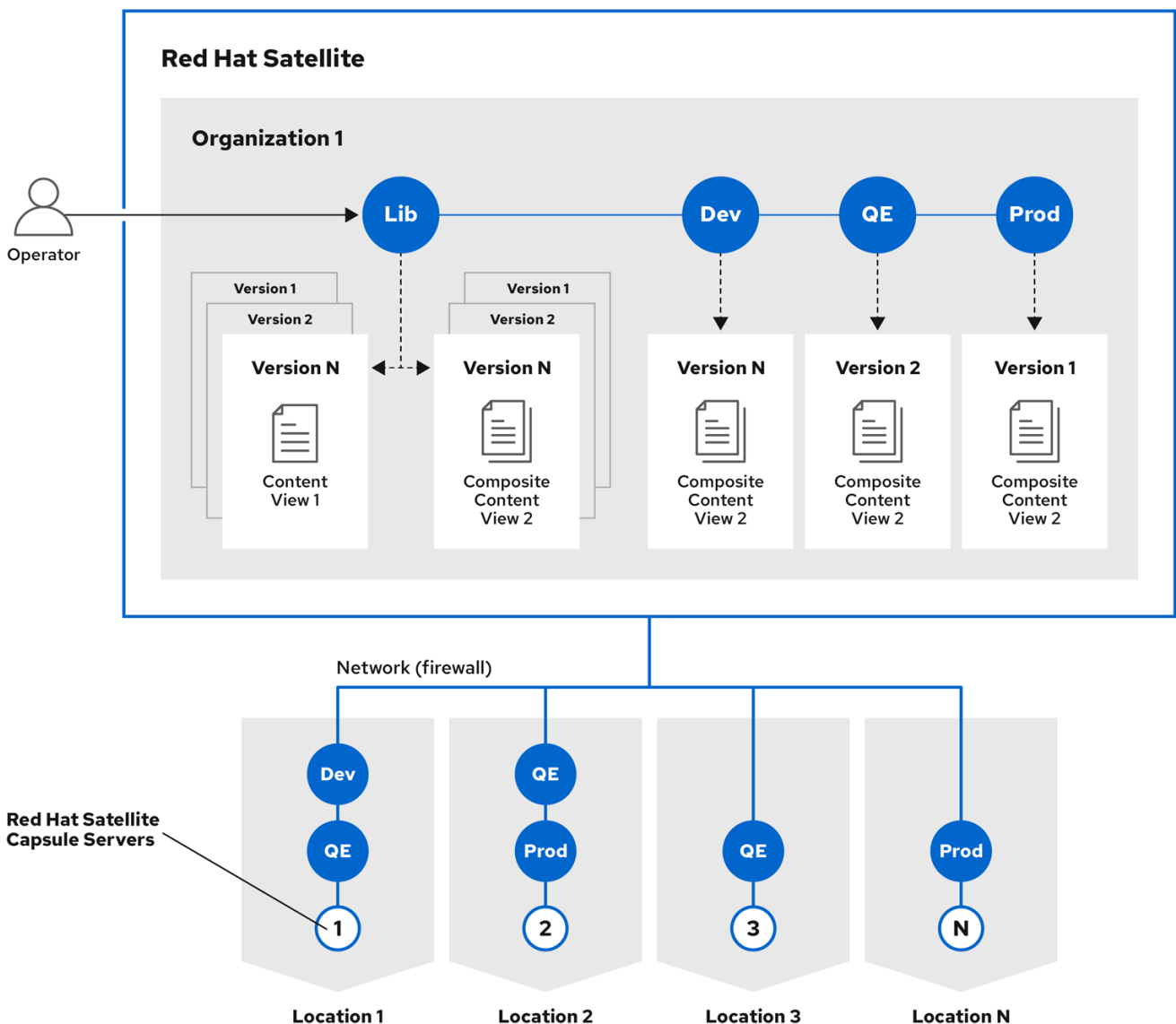
サブスクリプションマニフェストは Red Hat Satellite Server からアクセスできる Red Hat リポジトリを決定します。Red Hat リポジトリを有効にすると、関連付けられた Satellite 製品が自動的に作成されます。カスタムソースからコンテンツを配信するには、製品およびリポジトリを手動で作成する必要があります。Red Hat リポジトリは、デフォルトでは GPG キーで署名されるため、カスタムリポジトリ用にも GPG キーを作成することが推奨されます。

RPM パッケージのみを含む Yum リポジトリーは、同期時間とストレージ容量を削減する **オンデマンド** ダウンロードポリシーをサポートします。**オンデマンド** ダウンロードポリシーは、ホストから要求された場合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。コンテンツソースの設定に関する詳細な手順については、コンテンツの **管理** の **コンテンツのインポート** を参照してください。

Satellite Server 内のカスタムリポジトリーには、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すが、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。

8.5. コンテンツのライフサイクル

Satellite には、コンテンツライフサイクルを詳細に管理する各種機能が含まれています。**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、フィルター機能があるコンテンツセットであり、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ことにより、定義された方法でホストがコンテンツを利用できるようにします。



278_Satellite_0922

コンテンツ管理プロセスの詳細な概要については、コンテンツの管理の [カスタムコンテンツのインポート](#) 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一般的なシナリオを説明します。

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、接続したすべての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、ホストが利用できる前にコンテンツをテストできなくなるため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。

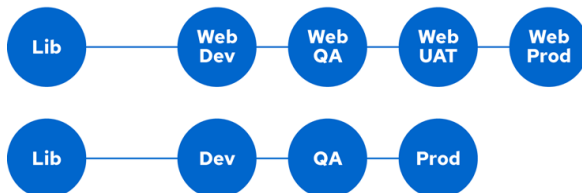
- **単一ライフサイクル環境:** ライブラリーのコンテンツは、実稼働ステージに直接プロモートされます。このアプローチでは、複雑さの面で制限がありますが、ホストで利用可能にする前に、ライブラリー内でコンテンツをテストできます。



- **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 (品質保証)**、**Production (実稼働)**)で設定されており、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異なるパスがあるため、アプリケーション別の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、このシナリオでは、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が制限されます。コンテンツソースを変更すると、一部のホストに影響を及ぼします。
- **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、ホストタイプの数が少ない(最高 30)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、オペレーティングシステムやアプリケーションコンテンツ専用のコンテンツ

ビューを使用できます。複合を使用すると、オペレーティングシステムとアプリケーションをさまざまな頻度で別々に管理できます。

- **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。コンテンツビューを作成しすぎないようにする必要がありますが、コンテンツビューのサイズも関連する操作 (公開、プロモート) の速度に影響を及ぼす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、依存関係もすべて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ーは、ホストのプロビジョニングにのみ使用されるため、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。

8.6. コンテンツのデプロイメント

コンテンツデプロイメントは、コンテンツホスト上のエラータとパッケージを管理します。Satellite は、MQTT/HTTPS (プルベース) または SSH (プッシュベース) 経由でリモート実行を実行するように設定できます。リモート実行は Satellite Server でデフォルトで有効になっていますが、Capsule Server およびコンテンツホストでは無効になっています。これは手動で有効にする必要があります。

8.7. プロビジョニング

Satellite には、プロビジョニングテンプレート、Puppet を使用した設定管理、ホストロールを標準化してプロビジョニングするホストグループなど、ホストプロビジョニングの自動化を容易にする機能が複数含まれています。プロビジョニングワークフローの説明については、[ホストのプロビジョニングのプロビジョニングワークフロー](#) を参照してください。このガイドには、各種のコンピュートリソースでのプロビジョニング方法が記載されています。

8.8. 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いて Satellite 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わたります。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なる 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、および実稼働の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を定義する:** ロールへアクセス可能な Satellite インフラストラクチャーのサブセットと、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を使用する (可能な場合):** Satellite 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。既存のロールをコピーおよび編集して、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、関連する分野は広範囲に及ぶ場合があります。そのため、ロールに読み取り専用アクセスを付与する際に、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部分のみに限定できます。これにより、ユーザーは今後の変更の可能性に関する情報を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ことが効果的な方法です。

ロールを定義してユーザーに割り当てる手順については、[Red Hat Satellite の管理のユーザーとロールの管理](#)を参照してください。このガイドには、外部の認証ソースを設定する方法が記載されています。

8.9. 追加タスク

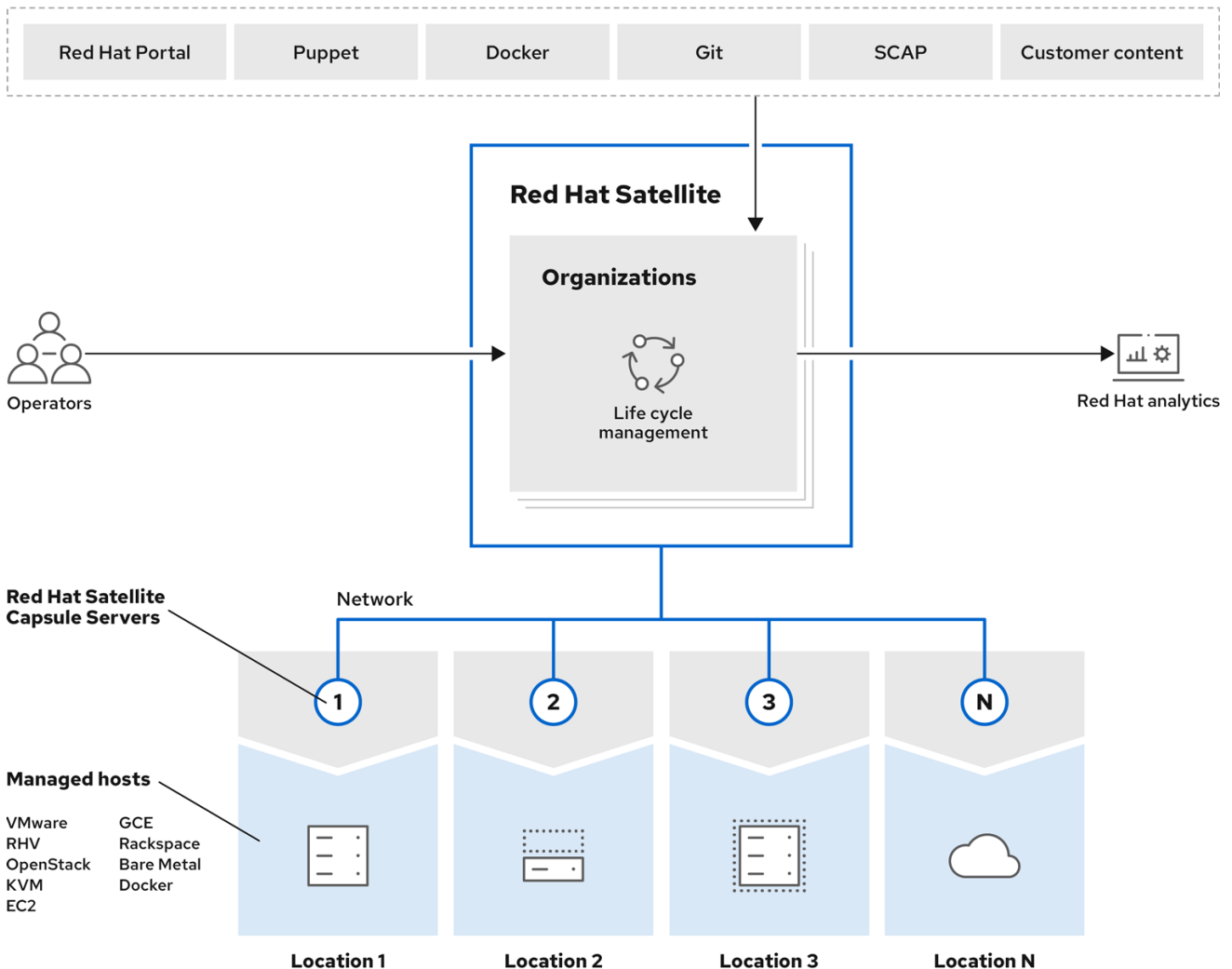
このセクションでは、特定のタスクを自動化したり、Satellite のコアの使用率を拡張したりするために使用できる Satellite 機能の一部を簡単に概説します。

- **ベアメタルホストの検出** - Satellite Discovery プラグインを使用すると、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が可能になります。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イス、メモリー、およびディスク情報などの Factor 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。詳細は、[ホストのプロビジョニングの検出されたホストからのホストの作成](#)を参照してください。
- **バックアップ管理** - バックアップと障害復旧の手順については、[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。リモート実行を使用すると、ホスト上で定期的なバックアップタスクを設定することもできます。リモート実行の詳細は、[ホストの管理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。
- **セキュリティ管理:** Satellite は、セキュリティ管理をさまざまな方法でサポートします。たとえば、更新およびエラー管理、システム検証のための OpenSCAP 統合、更新およびセキュリティコンプライアンスのレポート作成、詳細なロールベースの認証が挙げられます。エラー管理と OpenSCAP の概念の詳細は、[ホストの管理](#)を参照してください。
- **インシデント管理:** Satellite は、レポート作成やメール通知など、全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。最近の変更を示すイベント履歴をはじめ、各ホストの詳細情報は、Satellite Server からアクセスできます。また、Satellite は [Red Hat Insights](#) と統合されています。
- **Hammer および API を使用したスクリプト:** Satellite 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、[Hammer CLI ガイド](#)と[API ガイド](#)を参照してください。

第9章 組織、場所、ライフサイクル環境

Red Hat Satellite は、組織とロケーションの管理に対して統合的なアプローチを取ります。システム管理者は、1台の Satellite Server に、複数の組織とロケーションを定義します。たとえば、3つの国(米国、英国、日本)に3つの組織(Finance(財務)、Marketing(マーケティング)、Sales(営業))がある会社について考えてみましょう。この例では、Satellite Server がすべての地理的な場所(ロケーション)にあるすべての組織を管理しているため、システムを管理するためのコンテキストを9つ作成します。さらに、ユーザーは特定のロケーションを定義し、そのロケーションをネストして階層を作成できます。たとえば、Satellite 管理者が、米国のロケーションをさらにボストン、フェニックス、サンフランシスコなどの都市に分けるような場合です。

図9.1 Red Hat Satellite のトポロジー例



278_Satellite_1022

Satellite Server は、ロケーションと組織をすべて定義します。各 Satellite Capsule Server がコンテンツを同期し、ロケーションが異なるシステムの設定を処理します。

コンテンツと設定は、中央の Satellite Server と、特定のロケーションに割り当てられた Satellite Capsule Server との間で同期され、中央の Satellite Server が管理機能を保持します。

9.1. 組織

組織は、所有者、目的、コンテンツ、セキュリティーレベルなどの区分に基づいて Red Hat Satellite リソースを論理グループに分割します。Red Hat Satellite では複数の組織を作成して管理し、サブスクリ

プッシュを分けて各組織に割り当てることができます。これにより、1つの管理システムで複数の個別の組織のコンテンツを管理できるようになります。

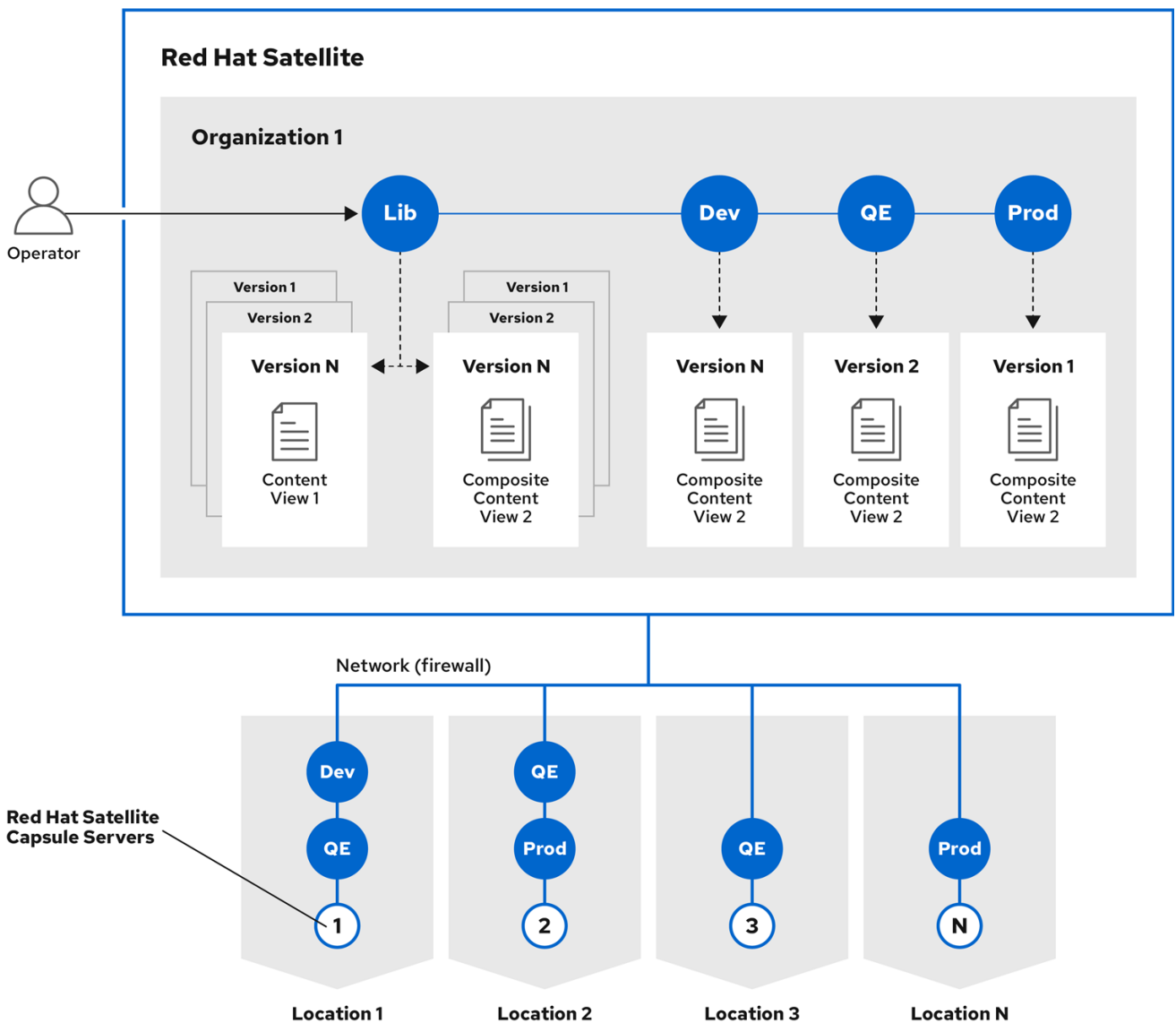
9.2. ロケーション

ロケーションは、地理的なロケーションに基づいて組織を論理グループに分割します。各ロケーションは、1つのアカウントで作成および使用されますが、1つのアカウントで管理できるロケーションと組織の数は複数になります。

9.3. ライフサイクル環境

アプリケーションライフサイクルは、アプリケーションライフサイクルの各ステージを表す **ライフサイクル環境** に分割されます。ライフサイクル環境はリンクされて **環境パス** を形成します。コンテンツは、必要に応じて環境パスの次のライフサイクル環境にプロモートできます。たとえば、アプリケーションのあるバージョンの開発が終了したら、そのバージョンをテスト環境にプロモートし、次のバージョンの開発を開始できます。

図9.2 4つの環境を含む環境パス



278_Satellite_0922

第10章 ホストグループ化の概念

Capsule Server の物理トポロジーとは別に、Red Hat Satellite はホストのグループ化に複数の論理ユニットを提供します。こ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。

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

以下は Red Hat Satellite の主な論理グループです。

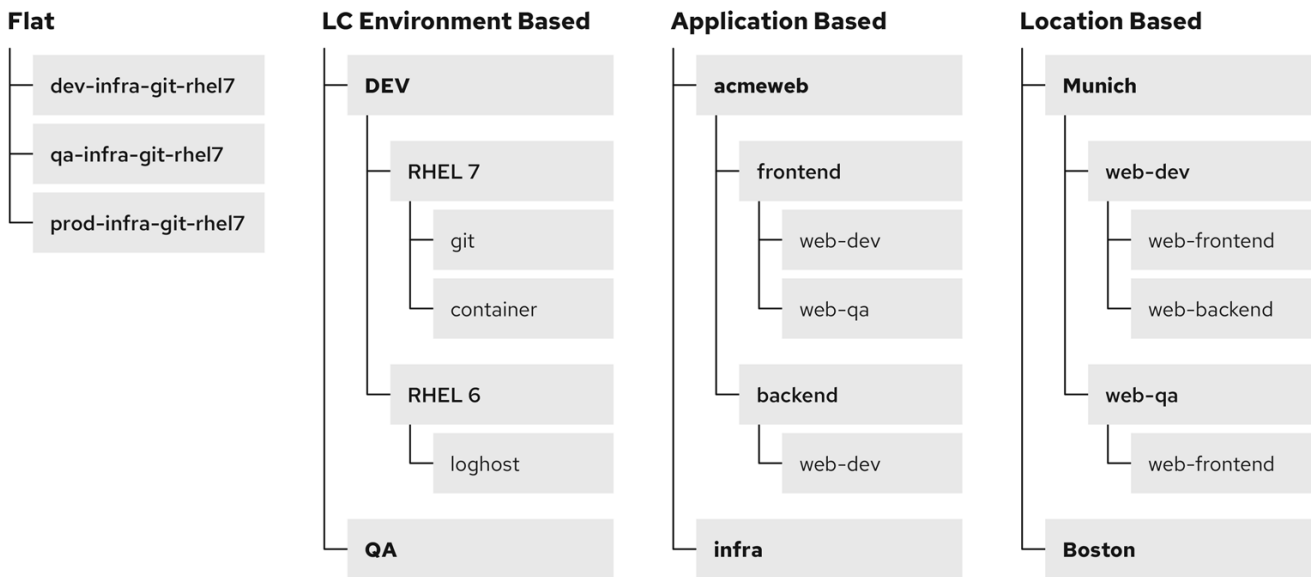
- **組織**: ホストの高位の論理グループです。組織により、コンテンツと設定が明確に区別されます。各組織には個別のサブスクリプションマニフェストが必要であり、それぞれを Red Hat Satellite Server の個別の仮想インスタンスと見なすことができます。低位のホストグループが適用可能な場合には組織を使用しないようにしてください。
- **ロケーション**: 物理的なロケーションに一致するホストのグループ分けです。ロケーションは、ネットワークインフラストラクチャーのマッピングに使用して、間違ったホストの配置または設定を防ぐことができます。たとえば、サブネット、ドメイン、またはコンピュータリソースを直接 Capsule Server に割り当てることはできず、ロケーションにのみ割り当てられます。
- **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。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。新規ホストの設定は、適切なホストグループへの追加が主な作業になります。ホストグループはネスト化できるため、要件に最も適した設定を作成できます ([「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の管理目的で 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分類できるため、パッケージ管理やエラーインストールなどの一括処理が可能になります。

場所およびホストグループは入れ子にすることができます。組織とホストコレクションはフラットです。

10.1. 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。ホストグループの構造を入念に計画すれば、ホスト設定の保守が容易になります。本セクションでは、ホストグループを体系化するための4つのアプローチを説明します。

図10.1 ホストグループの構造例



278_Satellite_1022

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、あまり複雑にはならないことです。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、ホストグループの設定が重複す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使用します。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。この構造は、ライフサイクル環境ごとに責任が分けられているシナリオで役に立ちます (たとえば、ライフサイクルのDevelopment (開発)、QA (品質保証)、Production (実稼働)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に対するホストのロールに基づいています。たとえば、これにより、バックエンドおよびフロントエンドのサーバーグループに対するネットワーク設定の定義が可能になります。ホストの一部の特徴を、Puppet ベースの、複雑な設定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、この構造はロケーション間でのパラメーターの共有を複雑にします。そのため、アプリケーションが多数ある複雑な環境では、設定を変更するたびに、変更が必要になるホストグループの数が大幅に増加します。

第11章 プロビジョニングの概念

Red Hat Satellite の重要な機能は、ホストの無人プロビジョニングです。これを実現するために、Red Hat Satellite では、DNS および DHCP インフラストラクチャー、PXE ブート、TFTP、および Kickstart が使用されます。本章では、これらの概念の動作原理を説明します。

11.1. PXE ブート

PXE (preboot execution environment) は、ネットワーク上でシステムを起動することができます。PXE は、ローカルハードディスクまたは CD-ROM を使用する代わりに、DHCP を使用してネットワークに関する一般情報をホストに提供し、TFTP サーバーを検出し、ブートイメージをダウンロードします。PXE サーバー設定の詳細は、Red Hat ナレッジベースのソリューション [How to set-up/configure a PXE Server](#) を参照してください。

11.1.1. PXE シーケンス

1. ホストは、ブート可能なイメージが見つからない場合に、PXE イメージをブートします。
2. ホストの NIC が、DHCP サーバーにブロードキャスト要求を送ります。
3. DHCP サーバーが、要求を受け取り、ネットワークに関する一般情報 (IP アドレス、サブネットマスク、ゲートウェイ、DNS、TFTP サーバーのロケーション、ブートイメージ) を送ります。
4. ホストが、TFTP サーバーから、ブートローダー **image/pxelinux.0** および設定ファイル **pxelinux.cfg/00:MA:CA:AD:D** を取得します。
5. ホスト設定が、カーネルイメージ **initrd** のロケーションとキックスタートを指定します。
6. ホストが、ファイルをダウンロードして、イメージをインストールします。

Satellite Server による PXE ブートの使用例については、[ホストのプロビジョニングの プロビジョニングネットワークフロー](#) を参照してください。

11.1.2. PXE ブートの要件

PXE ブートでマシンをプロビジョニングするには、以下の要件を満たしていることを確認してください。

ネットワーク要件

- オプション: ホストおよび DHCP サーバーがルーターで隔てられている場合は、DHCP リレーエージェントを設定し、DHCP サーバーを指定しておく。

クライアント要件

- すべてのネットワークベースのファイアウォールで、サブネットのクライアントが Capsule にアクセスするように設定すること。詳細は、「[カプセルネットワーク](#)」を参照してください。
- クライアントが DHCP サーバーと TFTP サーバーにアクセスできること。

Satellite 要件

- Satellite Server と Capsule の両方に DNS が設定され、プロビジョニングされたホスト名を解決できること。
- クライアントがブートオプションで DHCP オファーを利用できるように、クライアントが UDP ポート 67 および 68 にアクセス可能であること。
- クライアントが Capsule で TFTP サーバーにアクセスできるように、UDP ポート 69 にアクセス可能であること。
- クライアントが Capsule からファイルおよびキックスタートテンプレートをダウンロードできるように、TCP ポート 80 にアクセス可能であること。
- ホストプロビジョニングインターフェイスサブネットに DHCP Capsule セットがあること。
- ホストプロビジョニングインターフェイスサブネットに TFTP Capsule セットがあること。
- ホストプロビジョニングインターフェイスサブネットに Templates Capsule セットがあること。
- Satellite インストーラーを使用して、正しいサブネットを指定した DHCP を有効にしておくこと。
- Satellite インストーラーを使用して TFTP を有効にしておくこと。

11.2. HTTP ブート

HTTP ブートを使用して、HTTP を使用するネットワーク経路でシステムをブートできます。

11.2.1. 管理された DHCP による HTTP ブート要件

HTTP ブートでマシンをプロビジョニングするには、以下の要件を満たしていることを確認してください。

クライアント要件

HTTP ブートが機能するには、お使いの環境に以下のクライアント側の設定があることを確認します。

- すべてのネットワークベースのファイアウォールで、サブネットのクライアントが Capsule にアクセスするように設定すること。詳細は、「[カプセルネットワーク](#)」を参照してください。
- クライアントが DHCP サーバーと DNS サーバーにアクセスできること。
- クライアントが HTTP UEFI ブート Capsule にアクセスできること。

ネットワーク要件

- オプション: ホストおよび DHCP サーバーがルーターで隔てられている場合は、DHCP リレーエージェントを設定し、DHCP サーバーを指定しておく。

Satellite 要件

TFTP プロトコルは HTTP UEFI ブートには使用されませんが、Satellite は TFTP Capsule API を使用してブートローダー設定をデプロイします。

HTTP ブートが機能するには、Satellite で以下の設定があることを確認します。

- Satellite Server と Capsule の両方に DNS が設定され、プロビジョニングされたホスト名を解決できること。
- クライアントが DHCP の要求とオファーを送受信できるように、クライアントが UDP ポート 67 および 68 にアクセス可能であること。
- Capsule からブートローダーおよびキックスタートテンプレートをダウンロードできるように、クライアントに対して TCP ポート 8000 が解放してあること。
- HTTPS プロトコルを使用して Capsule からブートローダーをダウンロードできるように、クライアントに対して TCP ポート 9090 が解放してあること。
- ホストのプロビジョニングインターフェイスとして機能するサブネットには、DHCP カプセル、HTTP ブートカプセル、TFTP カプセル、テンプレートカプセルがあります。
- **grub2-efi** パッケージが最新版に更新されていること。**grub2-efi** パッケージを最新バージョンに更新し、インストーラーを実行して最新のブートローダーを `/boot` から `/var/lib/tftpboot` ディレクトリーにコピーするには、以下のコマンドを入力します。

```
# satellite-maintain packages update grub2-efi
# satellite-installer
```

11.2.2. 管理されていない DHCP での HTTP ブート要件

マネージド DHCP を使用せずに HTTP ブートでマシンをプロビジョニングするには、以下の要件を満たしていることを確認してください。

クライアント要件

- HTTP UEFI ブート URL は以下のいずれかに設定する必要があります。
 - **http://capsule.example.com:8000**
 - **https://capsule.example.com:9090**
- クライアントが DHCP サーバーと DNS サーバーにアクセスできること。
- クライアントが HTTP UEFI ブート Capsule にアクセスできること。
- すべてのネットワークベースのファイアウォールで、サブネットのクライアントが Capsule にアクセスするように設定すること。詳細は、「[カプセルネットワーク](#)」を参照してください。

ネットワーク要件

- クライアントで利用可能なマネージド外の DHCP サーバー。
- クライアントで利用可能なマネージド外の DNS サーバー。DNS が利用できない場合は、IP アドレスを使用してクライアントを設定します。

Satellite 要件

TFTP プロトコルは HTTP UEFI ブートには使用されませんが、Satellite は TFTP Capsule API を使用してブートローダー設定をデプロイします。

- Satellite Server と Capsule の両方に DNS が設定され、プロビジョニングされたホスト名を解決できること。

- クライアントが DHCP の要求とオファーを送受信できるように、クライアントが UDP ポート 67 および 68 にアクセス可能であること。
- Capsule からブートローダーおよびキックスタートテンプレートをダウンロードできるように、クライアントに対して TCP ポート 8000 が解放してあること。
- クライアントが HTTPS 経由で Capsule からブートローダーをダウンロードできるように、TCP ポート 9090 が開いていることを確認します。
- ホストプロビジョニングインターフェイスサブネットに HTTP ブートカプセルが設定されていることを確認します。
- ホストプロビジョニングインターフェイスサブネットに TFTP Capsule セットがあること。
- ホストプロビジョニングインターフェイスサブネットに Templates Capsule セットがあること。
- **grub2-efi** パッケージを最新バージョンに更新し、インストーラーを実行して、**/boot** ディレクトリーから **/var/lib/tftpboot** ディレクトリーに最新のブートローダーをコピーすること。

```
# satellite-maintain packages update grub2-efi  
# satellite-installer
```


付録A SATELLITE によって提供され、必要とされる技術ユーザー

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。このアカウントは、ファイルの管理や Satellite に統合されたコンポーネントの所有権を処理する時に使用します。これらのアカウントには、固定の UID と GID が割り当てられている場合と、システム上で次に利用可能な UID と GID が割り当てられている場合があります。アカウントに割り当てられた UID と GID を制御するには、Satellite のインストール前にアカウントを定義してください。アカウントによっては UID と GID がハードコードされている場合があるので、Satellite のインストール時に作成された全アカウントで、これができるわけではありません。

以下の表は、インストール時に Satellite が作成した全アカウントのリストです。**柔軟な UID および GID** コラムで **はい** と指定されているアカウントは、Satellite のインストール前にカスタムの UID と GID を事前定義できます。

システムアカウントのホームディレクトリーとシェルディレクトリーは、Satellite を正常に動作させるのに必要であるため、変更しないでください。

Satellite が作成するローカルユーザーと競合する可能性があるため、Satellite のベースオペレーティングシステムのシステムユーザーには外部アイデンティティプロバイダーを使用できません。

表A.1 Satellite によって提供され、必要とされる技術ユーザー

ユーザー名	UID	グループ名	GID	柔軟な UID および GID	Home	Shell
foreman	該当なし	foreman	該当なし	はい	/usr/share/foreman	/sbin/nologin
foreman-proxy	該当なし	foreman-proxy	該当なし	はい	/usr/share/foreman-proxy	/sbin/nologin
apache	48	apache	48	いいえ	/usr/share/httpd	/sbin/nologin
postgres	26	postgres	26	いいえ	/var/lib/pgsql	/bin/bash
pulp	該当なし	pulp	該当なし	いいえ	該当なし	/sbin/nologin
puppet	52	puppet	52	いいえ	/opt/puppetlabs/server/data/puppetserver	/sbin/nologin
ssslauth	該当なし	ssslauth	76	いいえ	/run/ssslauth	/sbin/nologin
tomcat	53	tomcat	53	いいえ	/usr/share/tomcat	/bin/nologin

ユーザー名	UID	グループ名	GID	柔軟な UID および GID	Home	Shell
unbound	該当なし	unbound	該当なし	はい	/etc/unbound	/sbin/nologin

付録B SATELLITE で使用される用語集

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、新た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。

回答ファイル

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。回答ファイルはYAML形式で定義され、`/etc/foreman-installer/scenarios.d/`ディレクトリーに保存されます。

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関する概要を示すものです。

Audit (監査)

特定のユーザーが加えた変更について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。

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 では、選択したホストを管理する BMC インターフェイスを作成できます。

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。ブートディスクの種類には、**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** があります。

Capsule

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 ミラーとして機能する) ために、そしてローカライズされた他のサービス (Puppet サーバー、DHCP、DNS、TFTP など) を実行するために、Red Hat Satellite デプロイメントで使用できる追加サーバーです。Capsule は、複数の地理的なロケーションでの 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。ここでは、管理が必要な全リソースと、そのリソース間の依存関係がリスト表示されます。カタログは Puppet マニフェストから Puppet サーバ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。

コンプライアンスポリシー

指定されたホストに SCAP コンテンツに対するコンプライアンスチェックを実行する、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。

コンピュータプロファイル

コンピュータ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。

コンピュータリソース

Red Hat Satellite 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 Virtualization、Red Hat OpenStack Platform、EC2、VMWare などがあります。

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 は、専用のコンピュータ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。

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 は、コンテンツビューでイメージをホストに配信するだけでなく、コンテナイメージをインポートする各種の方法をサポートします。

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) または Docker イメージが含まれます。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用する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。

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。

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。

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できます。

発見されたホスト

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上で検出されたベアメタルホスト。

発見イメージ

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。

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。プラグインは、Satellite Server と Capsule Server で実行されるサービスと、ホストで実行される Discovery イメージの 3 つのコンポーネントで設定されます。

発見ルール

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

Docker タグ

コンテナイメージを、通常はイメージに保存されるアプリケーションのバージョンで区別するために使用するマークです。Satellite Web UI の **コンテンツ > Docker Tag** で、タグを使用してイメージをフィルタリングできます。

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。

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は **適用可能** となり、ホストのコンテンツビューにある場合は **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)。

外部ノード分類器

ホストの設定時に使用するサーバーへの追加データを提供するコンストラクト。Red Hat Satellite は、Satellite デプロイメント内の Puppet サーバーへの外部ノードの分類子として機能します。外部ノード分類子は、将来の Satellite バージョンでは削除される予定です。

Facter

Facter は一種のプログラムであり、それを実行するシステムに関する情報 (ファクト) を提供します。たとえば、Facter は合計メモリー、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できます。Puppet モジュールは、Facter が収集したホストのデータに基づいて、特定の設定を有効にします。

Fact (ファクト)

合計メモリー、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コンポーネントです。Foreman は、Red Hat Satellite の主要なアップストリーム対応製品です。

フォアマンフック

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。

Foreman フック機能は非推奨であり、将来の Satellite バージョンでは削除される予定です。

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。

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 を管理する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、スクリプトで実行したりできます。Hammer は対話式のシェルも提供します。

ホスト

Red Hat Satellite 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。

ホストコレクション

エラータのインストールなど、1つ以上のホストへの一括動作に使用されるユーザー定義グループです。

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が継承する、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。

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。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に必要なブートファイルのみが含まれます。

増分アップグレード (コンテンツビュー)

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、すでに公開されている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。セキュリティー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)

リモートで、Satellite Server からホストに実行されるコマンドです。すべてのジョブはジョブテンプレートで定義されます。

Katello

サブスクリプションとリポジトリの管理を担当する Foreman プラグイン。

遅延同期

リポジトリのデフォルトのダウンロードポリシーを **即時** から **オンデマンド** に変更する機能。オンデマンド設定では、ホストから要求されたときにのみパッケージをダウンロードすることで、ストレージスペースと同期時間を節約します。

場所

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 で同期済みの全リポジトリのコンテンツのコンテナです。ライブラリーは、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として、各組織にデフォルトで存在します。

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。

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。一般的なプロモートパス (例: 開発からテスト、テストから実稼働へ) で、コンテンツビューをプロモートすることができます。

マニフェスト (Red Hat 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 にサブスクリプションを転送するためのメカニズム。 [Puppet マニフェスト](#) と混同しないでください。

Satellite の移行

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。

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、ホストのコンプライアンス監査を提供するために Satellite に統合されています。

Organization

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

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。

パラメーター化されたクラス (スマートクラスパラメーター)

Puppet サーバ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。

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のパーミッションは、**view_architectures**、**create_architectures**、**edit_architectures**、および **destroy_architectures** となります。パーミッションはロールに分類し、ロール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、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。

プロモーション (コンテンツビュー)

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コンテンツを移行する動作を指します。詳細は、[コンテンツの管理](#) の [コンテンツビューの昇格](#) を参照してください。

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

ホストのプロビジョニング設定を定義します。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。

公開 (コンテンツビュー)

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。

パルプミラー

コンテンツをミラーリングする Capsule Server コンポーネント。

Puppet

Satellite の設定管理コンポーネント。

puppet-agent

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。

パペット環境

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。

パペットマニフェスト

Puppet スクリプト (拡張子 `.pp`) を参照します。このファイルには、パッケージ、サービス、ファイル、ユーザー、グループなど、必要なリソースセットを定義するコードが含まれており、各属性はキーと値のペアを使用します。

[マニフェスト \(Red Hat サブスクリプションマニフェスト\)](#) と混同しないでください。

パペットサーバー

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。

パペットモジュール

ユーザー、ファイル、サービスなどのリソースを管理するの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。

繰り返しのロジック

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 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 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。

Repository (リポジトリー)

コンテンツのコレクションにストレージを提供します。

リソースタイプ

ホスト、カプセル、アーキテクチャーなど、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使用されます。

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 は事前に定義されたロールを数多く提供します。

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。

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け取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 ライブラリーにミラーリングすることを指します。

同期プラン

コンテンツ同期の実行をスケジュールします。

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。タスクのステータスは、Satellite Web UI の **モニター** > **Satellite タスク** > **タスク** で監視できます。

Updating Satellite

Satellite Server および Capsule Server のインストールを z-stream リリースから次のリリース (例: Satellite 6.15.0 から Satellite 6.15.1) に進めるプロセス。

Upgrading Satellite

Satellite Server および Capsule Server のインストールを y-stream リリースから次のリリース (例: Satellite 6.14 から Satellite 6.15) に進めるプロセス。

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ができるロールのコレクション。

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 を使用し、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 で使用すると、virt-who はその 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にサブスクリプションを提供できます。

付録C CLI ヘルプ

Satellite は、Satellite Web UI、Hammer CLI、API、および Ansible コレクション `redhat.satellite` を通じて複数のユーザーインターフェイスを提供します。コマンドラインで Satellite を管理する場合は、次のヘルプ出力を参照してください。

Satellite サービス

Satellite Server および Capsule Server が操作に使用するサービスセット。**satellite-maintain** ツールを使用して、これらのサービスを管理できます。サービスの完全リストを表示するには、Satellite または Capsule Server がインストールされているマシンで、**satellite-maintain service list** コマンドを実行します。詳細は、Satellite Server または Capsule Server で **satellite-maintain --help** を実行してください。

Satellite プラグイン

プラグインをインストールすることで Satellite を拡張できます。詳細は、Satellite Server または Capsule Server で **satellite-installer --full-help** を実行してください。

ハンマー CLI

hammer を使用してコマンドラインで Satellite を管理できます。Hammer CLI の使用に関する詳細は、[Hammer CLI ガイド](#)を参照するか、Satellite Server または Capsule Server で **hammer --help** を実行してください。