Red Hat Satellite 6.14

Satellite の概要、概念、およびデプロイメントの考慮事項

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

Last Updated: 2024-02-15

Red Hat Satellite 6.14 Satellite の概要、概念、およびデプロイメントの考慮事項

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

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

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 アーキテクチャー

第1章 RED HAT SATELLITE 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のシステムのデプロイ、設定、および保守を可能にするシステム管理ソリューションです。Satellite では、一元化された単一のツールを使用して複数の 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、監視が可能です。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。

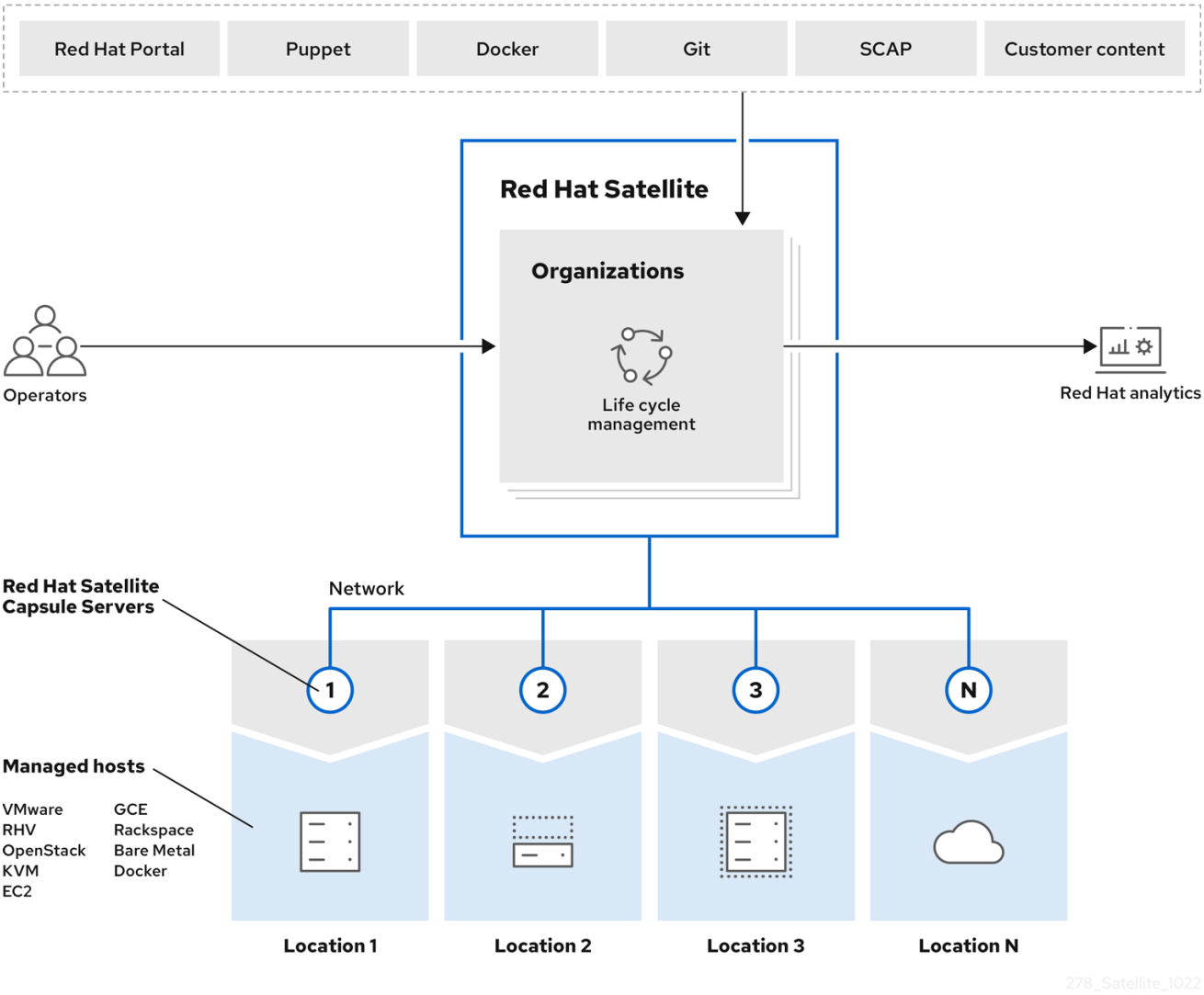
Capsule Server は、Satellite Server のコンテンツをミラーリングして、さまざまな地理的場所にわたるコンテンツのフェデレーションを容易にします。ホストシステムは中央の Satellite Server からではなく、それぞれのロケーションの Capsule Server からコンテンツおよび設定をプルできます。また、Capsule Server は Puppet サーバー、DHCP、DNS、TFTP などのローカライズされたサービスも提供します。Capsule Server は、マネージドシステムの数が増えたときに、Satellite 環境のスケーリングをサポートします。

Capsule Server により中央サーバーの負荷が減少して冗長性が増加し、帯域幅の使用率が低下します。詳細は、2章*Capsule Server の概要* を参照してください。

1.1. システムアーキテクチャー

以下の図は、Red Hat Satellite のアーキテクチャー全体の概要を示しています。

図1.1 Red Hat Satellite システムアーキテクチャー



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

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使用します。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、およびコンテナーイメージの主要なソースです。さらに、サポートされている他のコンテンツソース (Git リポジトリー、Docker ハブ、Puppet Forge、SCAP リポジトリー) 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できます。

Satellite Server

Satellite Server により、GUI、CLI、または API を使用して、コンテンツライフサイクルと、Capsule Server およびホストの設定を計画および管理できます。
Satellite Server では、ライフサイクル管理の分類に組織を主要な分類単位として使用します。組織によりホストグループのコンテンツが分離され、特定の要件および管理タスクが設定されます。たとえば、OS ビルドチームと Web 開発チームに別の組織を指定します。

Satellite Server には、詳細に設定された認証システムも含まれます。これにより、Satellite オペレーターには、各自の責任範疇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 Servers

Capsule Server は、さまざまな地理的なロケーションにコンテンツソースを設定するために Satellite Server 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の Satellite Server ではなく、それぞれのロケーションの 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 の数は、Satellite を使用する組織が運営される地理的なロケーションの数と同じかそれ以上に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 でホストが利用でき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は、図1.2「Red Hat Satellite のコンテンツライフサイクル」 を参照してください。

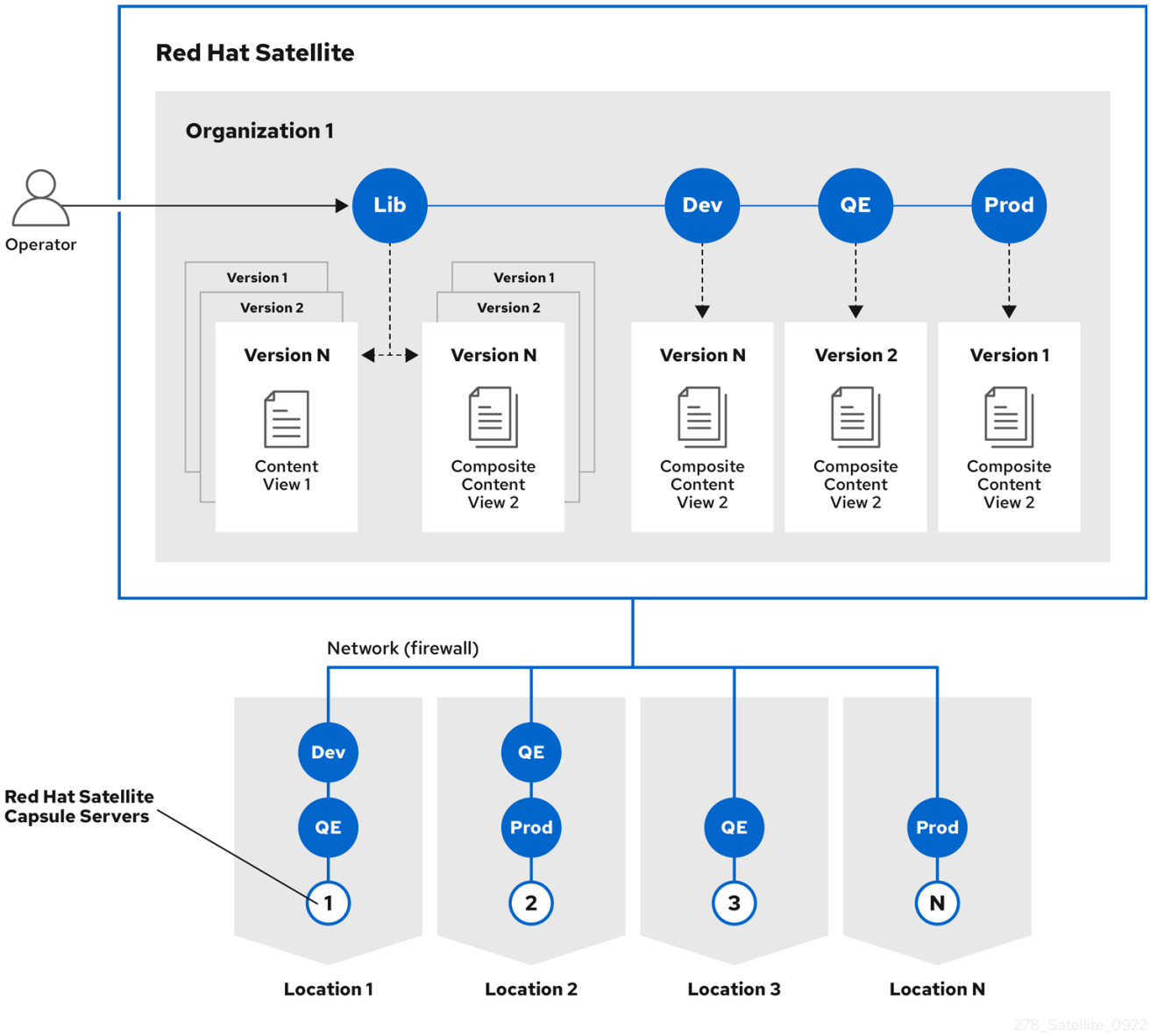
マネージドのホストと Satellite Server 間の通信経路は、ホストの代わりに複数のサービスを管理できる Capsule Server 経由で指定されます。このサービスの多くは専用のネットワークポートを使用しますが、Capsule Server は、ホストから Satellite Server への全通信にソース IP アドレスを 1 つ使用するようにします。これにより、ファイアウォールの管理が容易になります。Capsule Server の詳細は、2章*Capsule Server の概要* を参照してください。

マネージドホスト

ホストは、Capsule Server からコンテンツを受け取ります。ホストは物理ホストまたは仮想ホストになります。Satellite Server でホストを直接管理できます。Capsule Server を実行しているベースシステムも、Satellite Server のマネージドホストとなります。

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 のコンテンツライフサイクル



デフォルトでは、各組織には外部ソースのコンテンツのライブラリーがあります。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 (品質保証)、および Production (実稼働) 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、マネージド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て複合コンテンツビューを作成できます。オペレーティングシステムで必要なパッケージのリポジトリーと、アプリケーションで必要なパッケージのリポジトリーで、コンテンツビューを分けることには利点があります。たとえば、1 つのリポジトリーに含まれるパッケージに対する更新はすべて、関連するコンテンツビューを再公開するだけで完了します。したがって、複合コンテンツビューを利用すれば、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ため、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 で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、DNS、DHCP、および TFTP をインフラストラクチャーサービスとして実行でき、たとえばコンテンツサービスや設定サービスで補完することができます。

Capsule Server は、ライブラリーの同期したコンテンツを使用して、新規バージョンのコンテンツ

ビューを作成して更新することができます。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。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るので、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します。たとえば、Production (実稼働) のコンテンツビューにセキュリティーエラータを適用する必要がある場合に、コンテンツビューを他のライフサイクルにプロモートせずに直接更新できます。コンテンツ管理の詳細は、**コンテンツの管理** 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 は、Satellite として統合、検証、提供、およびサポートされる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。この情報は、Red Hat カスタマーポータルで管理され、定期的に更新されます。詳細は、Satellite 6 Component Versions を参照してください。

Red Hat Satellite は、以下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用して、これらのシステムを自動的に設定します。また、レポート、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ーを管理する Foreman プラグインです。Katello を使用して Red Hat リポジトリーをサブスクライブし、コンテンツをダウンロードできます。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、リポジトリーおよびコンテンツの管理を行う Katello 内のサービスです。Pulp を使用すれば、組織の異なるコンテンツビューが RPM パッケージを要求した場合にパッケージが重複しなくなるため、保存スペースを効率的に使用できます。

Hammer

Hammer は、コマンドラインおよびシェルを提供する CLI ツールで、Satellite Web UI とほぼ同様の機能を提供します。

REST API

Red Hat Satellite には RESTful API サービスが含まれます。このサービスにより、システム管理者および開発者は、カスタムスクリプトや Red Hat Satellite へのインターフェイスとな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、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 を実行するために取っ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。

Red Hat Satellite コンポーネントのサポートは、以下のようになります。

SELinux は、Enforcing モードまたは Permissive モードのいずれかである必要があります。無効化された SELinux でのインストールはサポートされません。

Puppet

Red Hat Satellite 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用すると、ユーザーは Puppet サーバーを Capsule Server の一部としてインストールおよび設定できます。Satellite Server または Satellite Capsule Server 上の Puppet サーバーで実行されている Puppet モジュールも Red Hat でサポートされています。サポート対象の Puppet バージョンについては、Red Hat ナレッジベースの記事 Satellite 6 Component Versions を参照してください。
Red Hat は、Puppet モジュールを含むスクリプトおよびフレームワークを多数サポートしています。フレームワークのサポートについては、Red Hat ナレッジベースの記事 スクリプトフレームワークのサポート状況 を参照してください。

Pulp

Pulp の使用は、Satellite 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を直接変更したり対話したりすると、Red Hat Satellite データベースに修復不能な破損が生じる可能性があるため、サポートされません。

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 リポジトリーのプラグインはサポートされません。
また、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コンポーネント、設定、および機能が含まれます。これらはすでに組み込まれており、使用することはできますが、Red Hat でサポートされるのは Red Hat Enterprise Linux での使用となります。

Candlepin

Candlepin の使用は、Satellite 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 データベースに修復不能な破損が生じる可能性があるため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、Satellite Web UI、API およびデータベースでの使用のみがサポートされます。Red Hat では、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はサポートしていません。

注記

すべての Red Hat Satellite コンポーネントの使用は Red Hat Satellite のコンテキスト内でのみサポートされます。コンポーネントをサードパーティーで使用した場合はサポート対象外となります。

1.4. サポート対象のクライアントアーキテクチャー

1.4.1. コンテンツ管理

Satellite でホストの登録と管理を行う場合に、サポート対象となる Red Hat Enterprise Linux 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。これには、Satellite Client 6 リポジトリーが含まれます。

表1.1 コンテンツ管理のサポート

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

1.4.2. ホストのプロビジョニング

Satellite でのホストのプロビジョニング時に、サポート対象となる Red Hat Enterprise Linux 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。

表1.2 ホストのプロビジョニングサポート

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

1.4.3. 設定管理

Satellite で設定管理を行う場合に、サポート対象となる Red Hat Enterprise Linux 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。

表1.3 Puppet Agent 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、aarch64
Red Hat Enterprise Linux 7	x86_64
Red Hat Enterprise Linux 6	x86_64、i386

第2章 CAPSULE SERVER の概要

Capsule Server には、**コンテンツフェデレーション** があり、**ローカライズされたサービス** を実行してホストの検出、プロビジョニング、制御および設定を行います。Capsule を使用して、さまざまな地理的ロケーションに Satellite デプロイメントを拡張できます。このセクションでは、Capsule で有効にできる機能の概要と、その簡単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、スケーラビリティーの留意事項に関する詳細は、Capsule Server のインストール を参照してください。

2.1. CAPSULE の機能

Capsule Server は 2 種類の機能を提供します。まず、ホスト管理に必要なサービスの実行に Capsule を使用できます。そして、Satellite Server からのコンテンツをミラーリングするように Capsule を設定することもできます。

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、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など、DHCP サーバーを管理できます。
- **DNS**: Capsule は、ISC BIND や Active Directory などの既存のソリューションとの統合など、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し、プロビジョニング時にホストが自動的に参加できるようにします。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。
- **Puppet サーバー** – Capsule は、Puppet サーバーを実行することにより設定管理サーバーとして機能できます。
- **Puppet 認証局**: Capsule は Puppet の CA と統合して、ホストに証明書を提供できます。
- **ベースボード管理コントローラー (BMC):** Capsule は、IPMI または Redfish を使用して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ー**: 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 はホストでリモートジョブを実行できます。

以下はコンテンツに関連する機能になります。

- **リポジトリーの同期**: コンテンツ配信のために、Satellite Server (具体的には選択したライフサイクル環境) のコンテンツを Capsule Server にプルします (Pulp により有効になります)。
- **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、中央の Satellite Server ではなく、Capsule からコンテンツをダウンロードします (Pulp により有効になります)。
- **ホストアクションの配信**: Capsule Server はホスト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) プロキシー**: 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、関連付けられている Capsule Server に登録されます (Candlepin で提供されます)。

2.2. CAPSULE のタイプ

すべての Capsule 機能を一度に有効にする必要はありません。Capsule Server は限定し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ーを有効にすると、インフラストラクチャー Capsule で、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が利用できます。

- **コンテンツ Capsule** [Pulp]: Satellite Server から同期したコンテンツをホストに提供します。

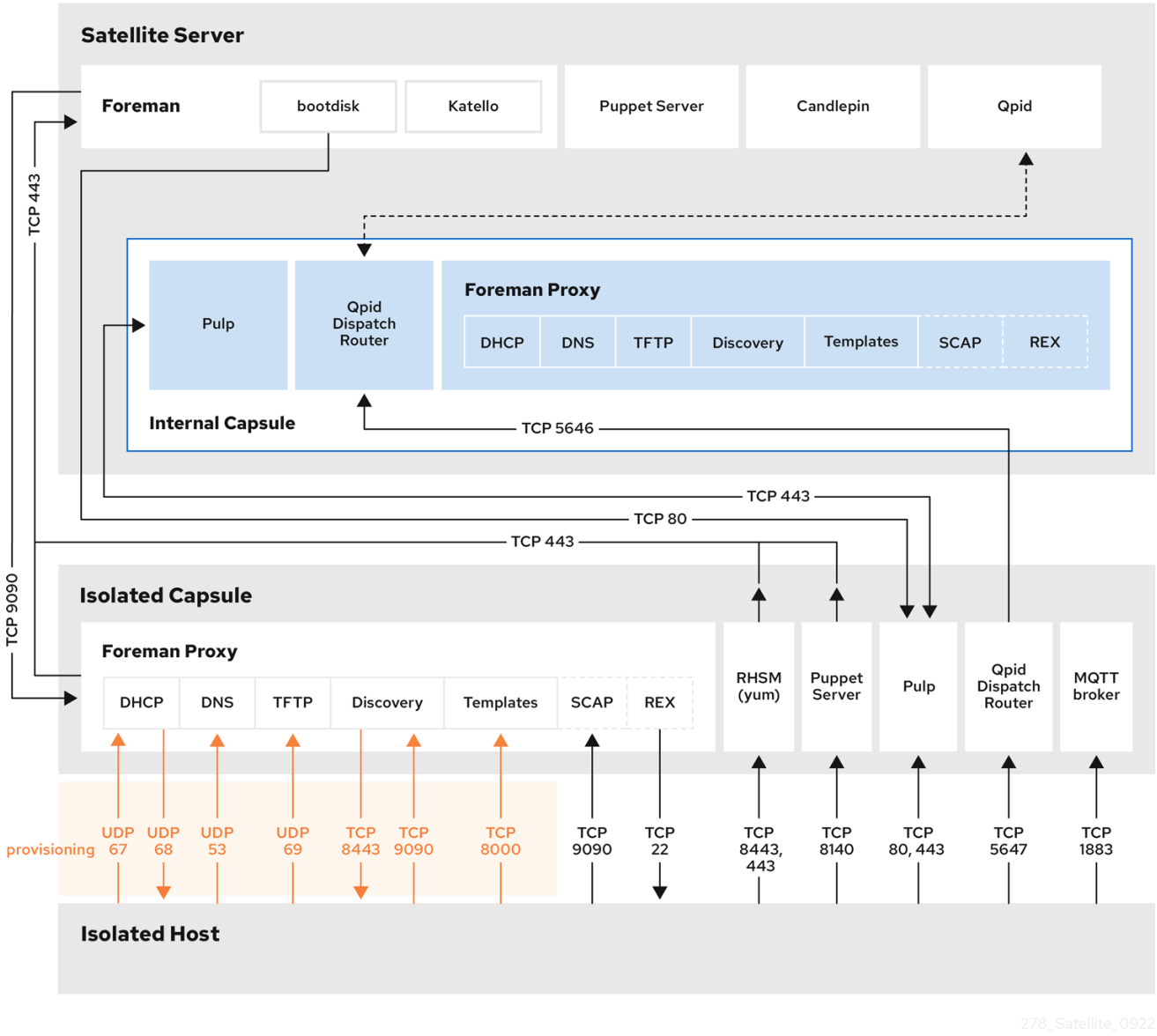
- **設定 Capsule** [Pulp + Puppet + PuppetCA]: コンテンツを提供し、ホストの設定サービスを実行します。

- **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、マネージド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

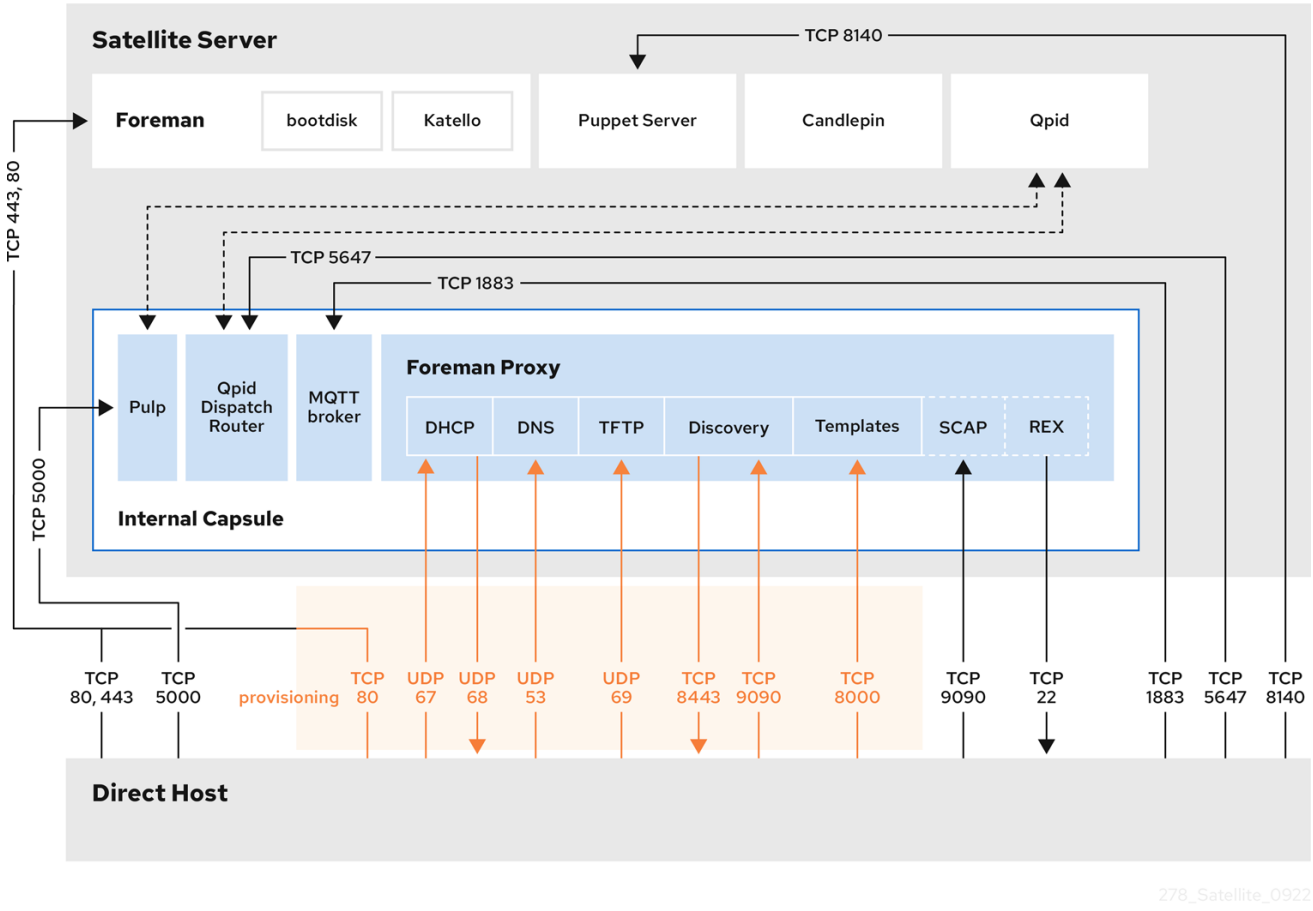
Capsule を分離する目的は、リモートネットワークセグメントでファイアウォールポートを Capsule 自体に開くだけで済むように、ホストの全ネットワーク通信のエンドポイントを 1 つだけ提供することにあります。以下の図は、分離した Capsule にホストが接続しているシナリオで、Satellite コンポーネントがどのように対話するかを示しています。

図2.1 分離した Capsule を含む Satellite トポロジー



以下の図は、ホストが Satellite Server に直接接続する場合に、Satellite コンポーネントがどのように対話するかを示しています。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しない場合でも関連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



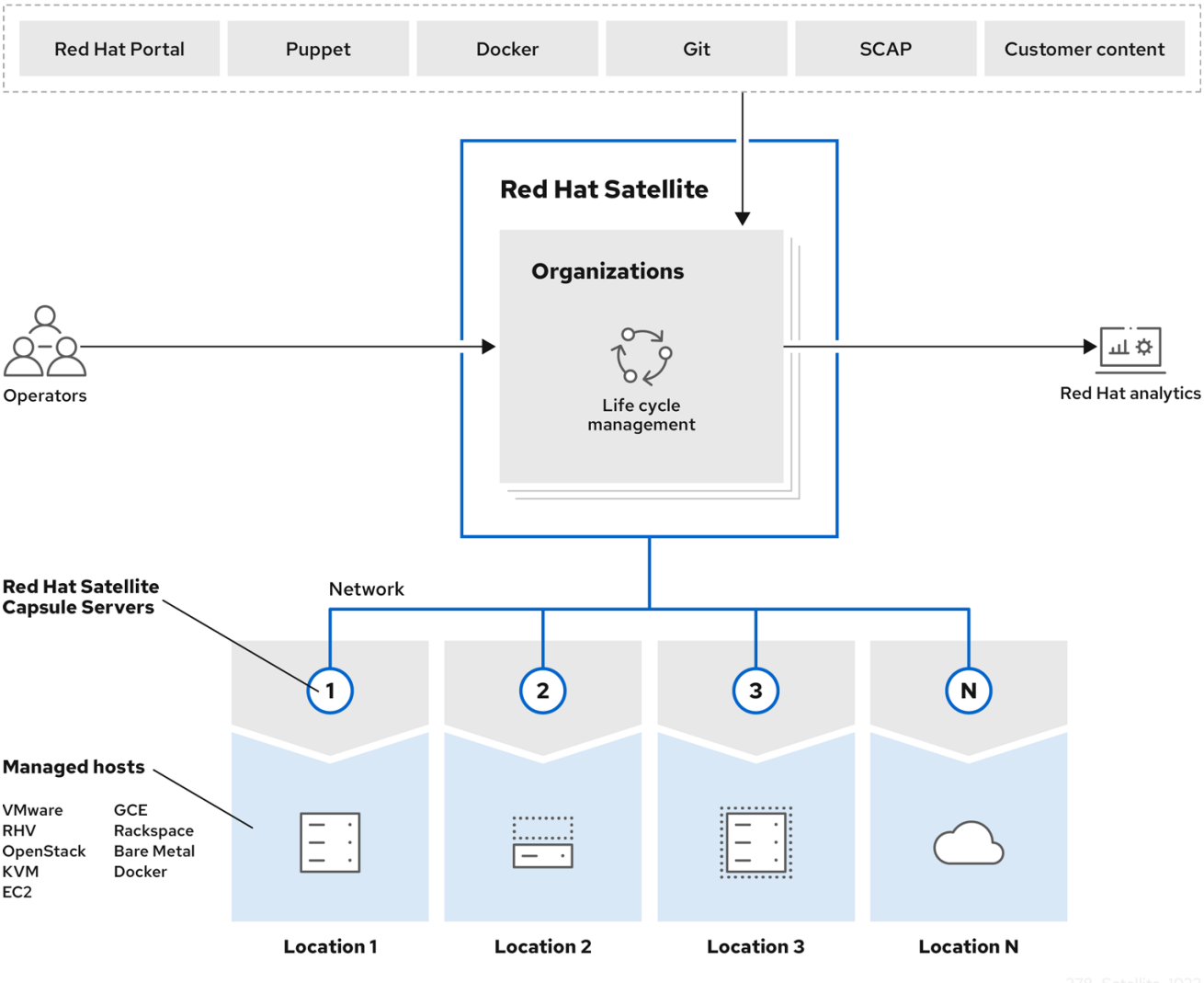
必要なポートを開くようにホストベースのファイアウォールを設定するための全手順は、次のドキュメントを参照してください。

- **オンラインネットワーク環境での Satellite Server のインストール** の ポートとファイアウォールの要件
- **オフラインネットワーク環境での Satellite Server のインストール** の ポートとファイアウォールの要件
- **Capsule Server のインストール** の ポートとファイアウォールの要件

第3章 組織、ロケーション、および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3.1 Red Hat Satellite のトポロジー例



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。

3.1. 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

プション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。

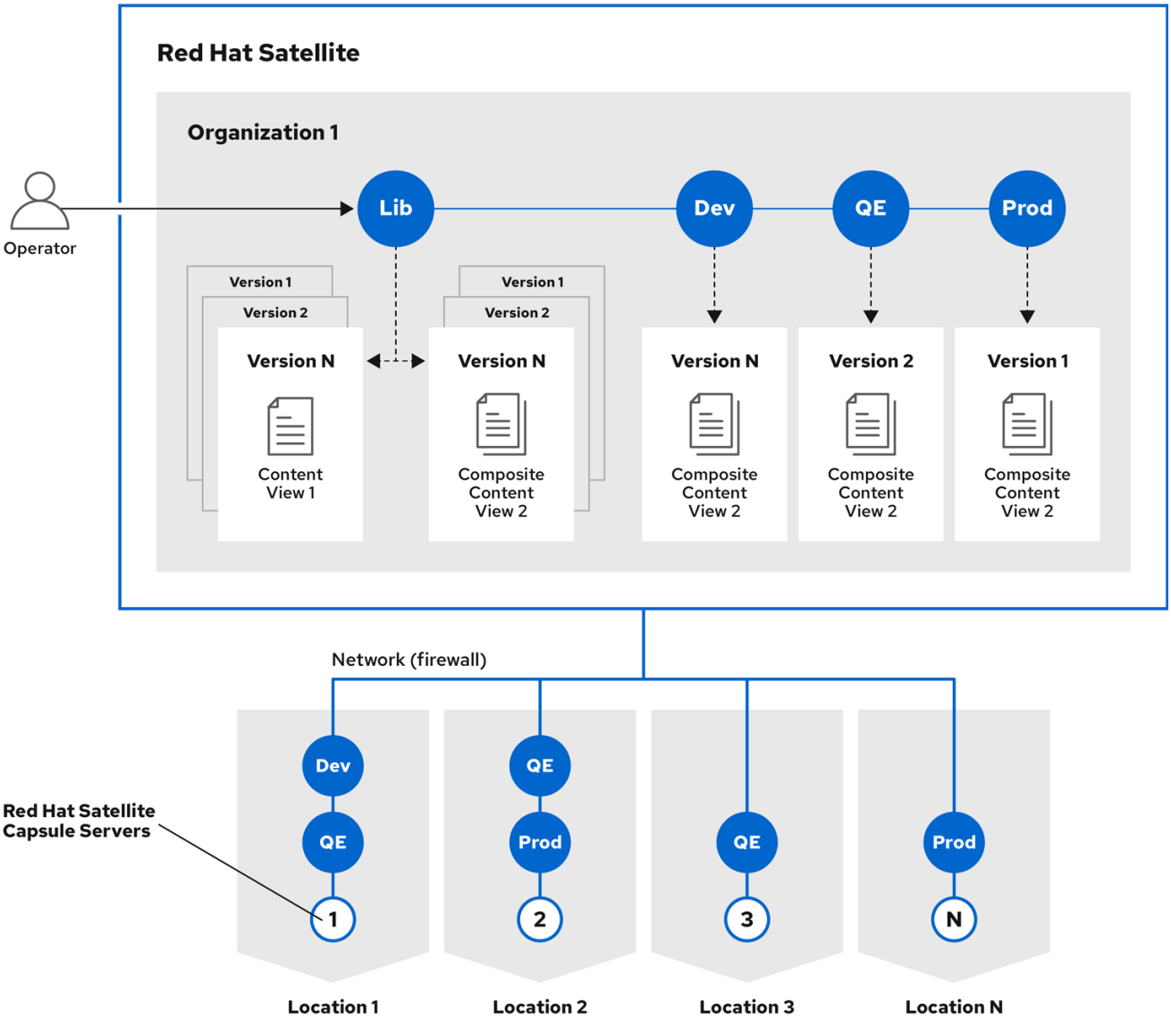
3.2. ロケーション

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。

3.3. 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** に分類されます。ライフサイクル環境はそれぞれがリンクし **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3.2 4つの環境を含む環境パス



第4章 ホストのグループ化の概念

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、またはコンピュートリソースを直接 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)。

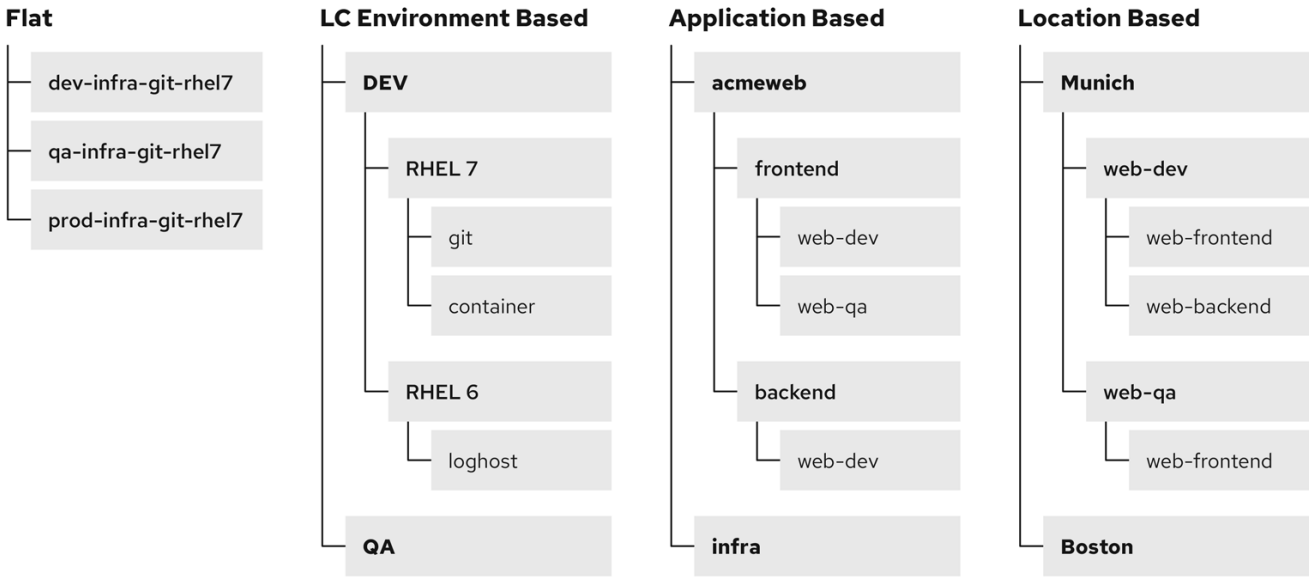
- **ホストコレクション**: サブスクリプションおよびコンテンツの管理目的で Satellite Server に登録されたホストは **コンテンツホスト** と呼ばれます。コンテンツホストはホストコレクションに分類できるため、パッケージ管理やエラータインストールなどの一括処理が可能になります。

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。

4.1. 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4.1 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、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5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。

5.1. 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。

5.1.1. 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。

5.1.2. PXE ブート要件

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、図2.1「分離した Capsule を含む Satellite トポロジー」 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。

5.2. 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由でシステムをブートできます。

5.2.1. 管理 DHCP を使用した HTTP ブートの要件

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、図2.1「分離した Capsule を含む Satellite トポロジー」 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、DHCP Capsule、HTTP ブート Capsule、TFTP Capsule、およびテンプレート Capsule があること。
- **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
```

5.2.2. マネージド外 DHCP を使用した HTTP ブートの要件

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、図2.1「分離した Capsule を含む Satellite トポロジー」 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。
- 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 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。
- **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
```

5.3. キックスタート

キックスタートを使用して、Red Hat Satellite または Capsule Server のインストールプロセスを自動化できます。これは、インストールに必要な情報をすべて含むキックスタートファイルを作成して行います。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。

5.3.1. ワークフロー

Red Hat Satellite キックスタートスクリプトを実行すると、以下のワークフローが発生します。

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。

第6章 コンテンツ配信ネットワーク構造

Red Hat コンテンツ配信ネットワーク (CDN) (**cdn.redhat.com**) は、地理的に分散された一連の静的 Web サーバーで、システムが使用することを目的としたコンテンツおよびエラータが含まれます。このコンテンツには、Subscription Manager を使用して登録されたシステムまたは Satellite Web UI を介して直接アクセスできます。この CDN のアクセス可能なサブネットは、Red Hat サブスクリプション管理 または Satellite Server を使用して、システムに割り当てられているサブスクリプションから設定します。

Red Hat コンテンツ配信ネットワークは、有効なユーザーのみがアクセスできるように、X.509 証明書認証によって保護されます。

CDN のディレクトリー構造

```
$ tree -d -L 11
└── content 1
    ├── beta 2
    │   └── rhel 3
    │       └── server 4
    │           └── 7 5
    │               └── x86_64 6
    │                   └── sat-tools 7
    └── dist
        └── rhel
            └── server
                └── 7
                    ├── 7.2
                    │   └── x86_64
                    │       └── kickstart
                    └── 7Server
                        └── x86_64
                            └── os
```

1. **content** ディレクトリー。
2. コンテンツのライフサイクルに関するディレクトリー。**beta** (ベータコードの場合)、**dist** (実稼働環境の場合)、**eus** (延長更新サポートの場合) などが一般的なディレクトリーとなります。
3. 製品名に関するディレクトリー。通常は、Red Hat Enterprise Linux を表す **rhel**。
4. 製品の種類に関するディレクトリー。Red Hat Enterprise Linux の場合は **server**、**workstation**、**computenode** などになります。
5. リリースバージョンに関するディレクトリー (例: **7**、**7.2**、**7Server**)。
6. ベースアーキテクチャーに関するディレクトリー (例: **i386** または **x86_64**)。
7. リポジトリー名に関するディレクトリー (例: **sat-tools**、**kickstart**、**rhscl**)。一部のコンポーネントには、変更可能な追加サブディレクトリーが含まれます。

このディレクトリー構造は、Red Hat Subscription Manifest でも使用されます。

パート II. SATELLITE デプロイメントの計画

第7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 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、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. サブネットが分離されている単一ロケーション

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トリソースを管理し、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ー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あがり、設定が容易になります。このアプローチを示した図については、図1.1「Red Hat Satellite システムアーキテクチャー」 を参照してください。

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は、オフラインネットワーク環境での Satellite Server のインストール を参照してください。ネットワーク接続なしで Satellite をアップグレードする方法は、**Red Hat Satellite 6.14 へのアップグレード** の 接続されていない 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。

- **切断された Satellite でコンテンツ ISO を使用**: 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、Satellite Server またはローカル Web サーバーにデプロイメントします。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。コンテ

ンツ ISO を切断された Satellite にインポートする方法は、**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 Inter-Satellite Synchronization を設定する方法については、**コンテンツの管理** の Satellite Server 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. 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 デプロイメントに統合できます。外部サービスを使用した Capsule Server の設定方法の詳細は、**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 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、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は、**オンラインネットワーク環境での Satellite Server のインストール** 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、**オンラインネットワーク環境での Satellite Server のインストール** の 事前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に Satellite Server をインストールする方法については、**オフラインネットワーク環境での Satellite Server のインストール** 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、**オフラインネットワーク環境での Satellite Server のインストール** の 事前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。

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

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ーを検索します。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。

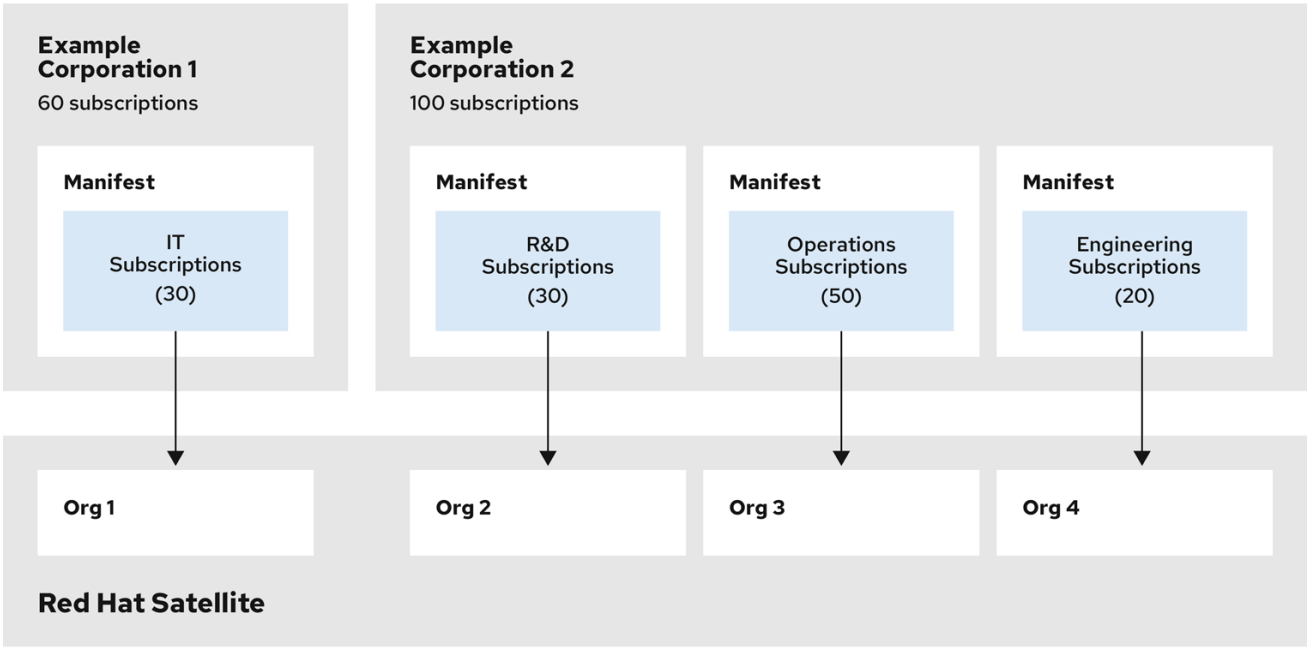
Red Hat Satellite では、Satellite に設定した各組織に対して、マニフェストが 1 つ必要になります。Satellite の組織機能を使用して、1 つの Red Hat Network アカウントで、インフラストラクチャーを分けて管理する予定がある場合は、必要に応じて 1 つのアカウントが所有するサブスクリプションを、各組織のマニフェストに割り当てます。

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 アカウントの保持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含む Satellite Server



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 ユーティリティー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ー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使用する SATELLITE SERVER

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。
- リポジトリー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、**オンラインネットワーク環境での Satellite Server のインストール** の Satellite での外部データベースの使用 を参照してください。

8.3. ロケーションおよびトポロジー

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 共通の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 の数であると考えることができます。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 (「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？**: ベアメタルホストのプロビジョニングのほかに、Satellite でサポートされる各種の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でのプロビジョニングについては、**ホストのプロビジョニング** を参照してください。

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 リポジトリーを決定します。Red Hat リポジトリーを有効にすると、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ーを手動で作成する必要があります。Red Hat リポジトリーは、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ー用にも 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ーの設定は、それらが保持するコンテンツのタイプ (RPM パッケージまたは Docker イメージ) によって異なります。

RPM パッケージのみを含む **yum** リポジトリーとして設定されるリポジトリーは、同期時間とストレージ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** および **オンデマンド** から選択できます。**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。コンテンツソースのセットアップに関する詳細な説明は、**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

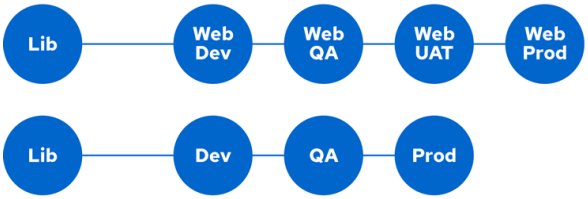
Satellite には、コンテンツライフサイクルを詳細に管理する各種機能が含まれています。**ライフサイクル環境** は、コンテンツライフサイクルのステージを表し、**コンテンツビュー** は、フィルター機能があるコンテンツセットであり、コンテンツの定義済みのサブセットとみなすことができます。コンテンツビューをライフサイクル環境に関連付けることにより、定義された方法でホストがコンテンツを利用できるようにします。プロセスの視覚化については、 を参照してください。コンテンツ管理プロセスの詳細な概要は、**コンテンツの管理** の カスタムコンテンツのインポート を参照してください。以下のセクションでは、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。特定のコンピュート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

コンテンツのデプロイメントは、コンテンツホストにおけるエラータおよびパッケージの管理です。Satellite でコンテンツをデプロイする方法は 2 つあります。デフォルトの方法は **リモート実行** で、2 つ目の方法は非推奨の **Katello エージェント** です。

- **リモート実行**: リモート実行では、パッケージのインストール、更新、または削除、設定管理エージェントのブートストラップ、および Puppet 実行のトリガーが可能になります。SSH (プッシュベース) または MQTT/HTTPS (プルベース) を介してリモート実行をするように、Satellite を設定できます。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。
  これは、コンテンツのデプロイメントで推奨される方法です。

- **Katello エージェント**: Katello エージェントはパッケージのインストールおよび更新を行います。このエージェントは、Satellite Server と通信する **goferd** サービスを使用します。**katello-agent** パッケージのインストールに成功すると、コンテンツホストで有効になり、自動的に起動します。
  Katello エージェントは非推奨で、今後の Satellite リリースで削除されることに注意してください。

8.7. PROVISIONING

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。プロビジョニングのワークフローの詳細は **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、定義したロールをユーザーに割り当てる方法は、**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 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イス、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**: Satellite Server のバックアップおよび障害復旧の手順を参照してください (**Red Hat Satellite の管理** の Satellite Server および Capsule Server のバックアップ を参照)。リモート実行を利用すると、マネージド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。リモート実行の詳細は、**ホストの管理** の リモートジョブの設定およびセットアップ を参照してください。

- **セキュリティー管理**: Satellite は、セキュリティー管理をさまざまな方法でサポートします。たとえば、更新およびエラータ管理、システム検証のための OpenSCAP 統合、更新およびセキュリティー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。エラータ管理および 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 とも統合されています。

- **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 および API ガイド を参照してください。

付録A SATELLITE で提供される、必須の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。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ので、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ープロバイダーを使用できません。

表A.1 Satellite で提供される、必須のテクニカルユーザー

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
qdrouterd	該当なし	qdrouterd	該当なし	はい	/var/lib/qdrouterd	/sbin/nologin
qpidd	該当なし	qpidd	該当なし	はい	/var/lib/qpidd	/sbin/nologin

ユーザー名	UID	グループ名	GID	柔軟な UID および GID	Home	Shell
saslauth	該当なし	saslauth	76	いいえ	/run/saslauthd	/sbin/nologin
tomcat	53	tomcat	53	いいえ	/usr/share/tomcat	/bin/nologin
unbound	該当なし	unbound	該当なし	はい	/etc/unbound	/sbin/nologin

付録B 用語集

この用語集では、Red Hat Satellite に関連する用語を紹介します。

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。

Answer file (応答ファイル)

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。応答ファイルは 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 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。

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 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。

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 (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 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 (スマートプロキシー) 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。

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 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。

Container (Docker Container) (コンテナー (Docker コンテナー))

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、専用のコンピュートリソースでコンテナーのプロビジョニングをサポートします。

Container Image (コンテナーイメージ)

コンテナー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、コンテナー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。

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できます。

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 で実行するサービス、Capsule Server で実行するサービス、ホスト上で実行する Discovery イメージの 3 つのコンポーネントで設定されます。

Discovery Rule (検出ルール)

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 Tag (Docker タグ)

コンテナーイメージを、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

セキュリティー修正、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)。

External Node Classifier (外部ノードの分類子)

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。
外部ノードの分類子は、次の Satellite バージョンで削除される点に注意してください。

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 Fater 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 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。

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。

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。
Foreman Hook 機能は非推奨となり、次の Satellite バージョンで削除される点に注意してください。

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 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。

Host Collection (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。

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。

Host Image (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。

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。

Job Template (ジョブテンプレート)

ジョブのプロパティーを定義します。

Katello

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

Lazy Sync (遅延同期)

即時 という **yum** リポジトリーのデフォルトのダウンロードポリシーを **オンデマンド** に変更する機能です。**オンデマンド** 設定は、クライアントからの要求時にパッケージをダウンロードすることのみで、ストレージ領域と同期時間を節約します。

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 で同期済みの全リポジトリーのコンテンツのコンテナーです。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ーです。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 Subscription Manifest)

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、マネージドホストのコンプライアンス監査を実行するために Satellite に統合されています。

組織

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。

Parametrized Class (Smart Class Parameter) (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ー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ー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。

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

リポジトリーおよびコンテンツ管理を行う Katello 内のサービス。

Pulp Mirror

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 (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)

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 Subscription Manifest) と混同しないでください。

Puppet 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。

Puppet Module (Puppet 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。

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 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)

コンテナー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、Capsule、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ー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独自の応答ファイルがあります。

Smart Proxy (スマートプロキシー)

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

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。

Synchronization Plan (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)

リポジトリー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 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。

Updating Satellite

z-stream を基準にして、Satellite Server および Capsule Server のインストールを次のリリースに上げるプロセスです (例: Satellite 6.14.0 から Satellite 6.14.1)。

Upgrading Satellite

y-stream を基準にして、Satellite Server および Capsule Server のインストールを次のリリースに上げるプロセスです (例: Satellite 6.13 から Satellite 6.14)。

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。