Red Hat Satellite 6.3

アーキテクチャーガイド

Satellite 6 デプロイメント計画

Last Updated: 2018-07-17

Red Hat Satellite 6.3 アーキテクチャーガイド

Satellite 6 デプロイメント計画

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

法律上の通知

概要

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

目次

パート I. SATELLITE 6 アーキテクチャー

第1章 RED HAT SATELLITE 6 について

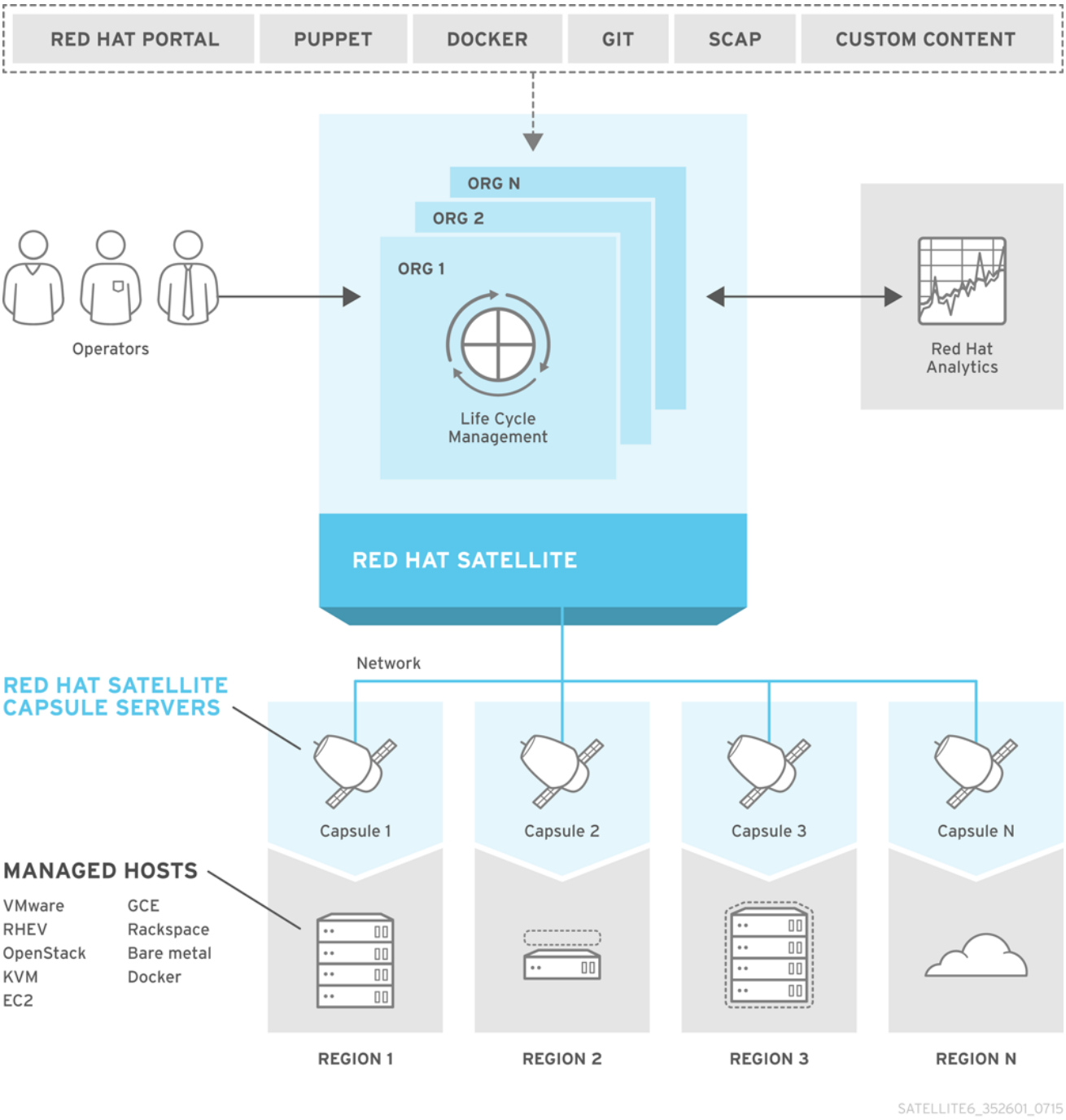
Red Hat Satellite は、物理環境、仮想環境、およびクラウド環境でシステムのデプロイ、設定、および保守を行うことを可能にするシステム管理ソリューションです。Satellite は、複数の Red Hat Enterprise Linux デプロイメントのプロビジョニング、リモート管理、および監視を、一元化された 1 つのツールで行うことを可能にします。**Red Hat Satellite Server** は、Red Hat カスタマーポータルおよびその他のソースから得たコンテンツを同期し、詳細なライフサイクル管理、ユーザーおよびグループのロールベースのアクセス制御、サブスクリプションの統合管理、高度な GUI、CLI、または API アクセスなどの機能を提供します。

Red Hat Satellite Capsule Server は、Red Hat Satellite Server のコンテンツをミラーリングして、複数の地理的な場所にわたるコンテンツフェデレーションを実現します。ホストシステムは、中央 Satellite Server ではなくローカルの Capsule Server からコンテンツおよび設定をプルできます。また、Capsule Server は、Puppet マスター、DHCP、DNS、TFTP などのローカライズされたサービスも提供します。Capsule Server を使用すると、環境内で管理対象システムの数が増えたときに、Red Hat Satellite のスケーリングが容易になります。

1.1. システムアーキテクチャー

以下の図は、Red Hat Satellite 6 のアーキテクチャー全体の概要を示しています。

図1.1 Red Hat Satellite 6 システムアーキテクチャー



このアーキテクチャーでは、コンテンツが 4 つのステージを通過します。

外部コンテンツソース

Red Hat Satellite Server は、各種ソースからのさまざまなタイプのコンテンツを使用します。これには、Red Hat カスタマーポータルへの接続が必要です。Red Hat カスタマーポータルは、ソフトウェアパッケージ、エラータ、Puppet モジュール、およびコンテナーイメージの主な提供元 (ソース) になります。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。

Red Hat Satellite Server

Red Hat 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類され、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 Server

Capsule Server は、さまざまな地理的な場所にコンテンツソースを設定するために Satellite Server 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、ローカルの Capsule Server からコンテンツおよび設定をプルできます。そのため Capsule Server の数は、Satellite を使用する組織が機能する地理的な場所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 6 におけるコンテンツのライフサイクル」 を参照してください。

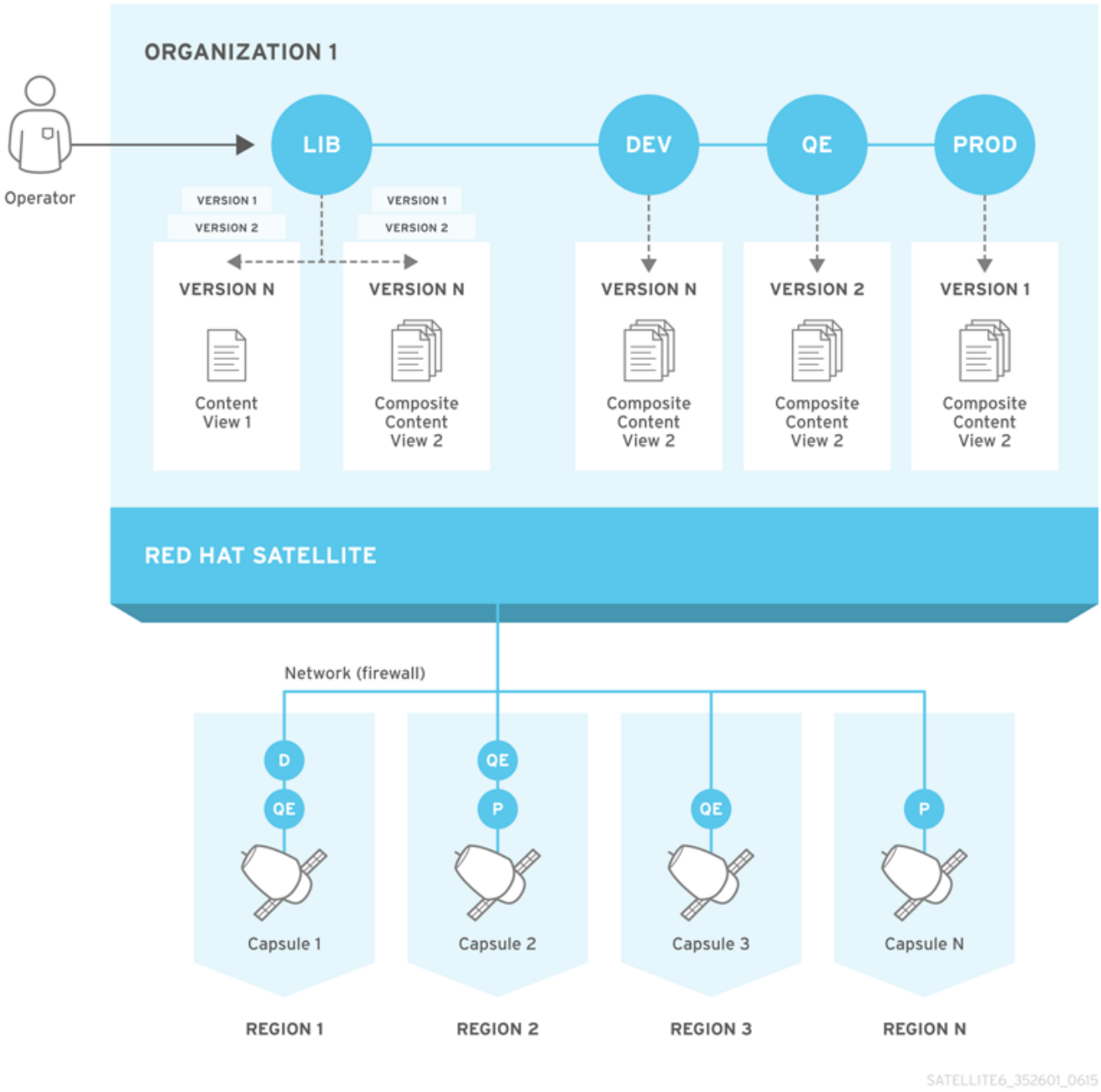
管理対象のホストと Satellite Server 間の通信経路は、ホストの代わりに複数のサービスを管理できる Capsule Server 経由で指定されます。このサービスの多くは専用のネットワークポートを使用しますが、Capsule Server は、ホストから Satellite Server への全通信にソース IP アドレスを 1 つ使用するようになります。これによりファイアウォールの管理が容易になります。Capsule Server の詳細は「2章 *Capsule Server の概要*」を参照してください。

管理対象ホスト

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 (KVM、VMware vSphere、OpenStack、Amazon EC2、Rackspace Cloud Services、Google Compute Engine、または Docker コンテナーにデプロイされる) 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 の管理対象となります。

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 6 におけるコンテンツのライフサイクル



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

コンテンツビューを組み合わせ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として実行できますが、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 の更新は、ライブラリーと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行います。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ます。つまり、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 (本番)」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、直接更新することができます。コンテンツ管理の詳細は『Red Hat Satellite コンテンツ管理ガイド』を参照してください。

1.2. システムコンポーネント

Red Hat Satellite 6 は、Satellite 6 として統合され、検証され、提供され、サポート対象となるいくつかのオープンソースプロジェクトで構成されています。詳細は、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。このページは定期的に更新されています。

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

Foreman

Foreman は、物理システムと仮想システムのプロビジョニングとライフサイクル管理に使用されるオープンソースのアプリケーションです。Foreman は、キックスタートや Puppet モジュールなどの様々な方法を使用してシステムを自動的に設定します。さらに Foreman はレポート、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 は、Red Hat リポジトリーをサブスクライブしてコンテンツをダウンロードする手段となります。コンテンツは、複数の異なるバージョンを作成して管理でき、各バージョンは、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を管理する Katello のサービスです。

Pulp

Pulp は、リポジトリーおよびコンテンツを管理する Katello のサービスです。Pulp を使用すれば、組織が異なるコンテンツビューが RPM パッケージをリクエスト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、Web UI とほぼ同様の機能を提供します。

REST API

Red Hat Satellite 6 には RESTful API サービスが含まれます。このサービスにより、システム管理者や開発者は、カスタムスクリプトや 、Red Hat Satellite へのインターフェー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 およびそのアップストリームコンポーネントで使用される用語は広範囲に及びます。よく使われる用語の説明は「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。

Puppet

Red Hat Satellite 6 には、サポートされる Puppet パッケージが含まれます。インストールプログラムを使用すると、Puppet マスターを Red Hat Satellite Capsule Server の一部としてインストールし、設定することができます。Red Hat Satellite Server または Satellite Capsule Server の Puppet マスターで実行する Puppet モジュールも、Red Hat のサポート対象となります。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 Server Web UI、CLI、および API 経由でのみサポートされます。Pulp のローカル API またはデータベースを直接変更したり対話したりすると、Red Hat Satellite 6 データベースに修復不能な破損が生じる可能性があるため、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、Red Hat Satellite 6 Web UI、CLI、および API 経由でのみサポートされます。Candlepin、そのローカル API またはデータベースとの直接的な対話については、Red Hat Satellite 6 データベースに修復不能な破損が生じる可能性がある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、Red Hat Satellite 6 WebUI、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。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されていません。

1.4. サポートされている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 6.3 でホストの登録および管理を行うときに、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 (Satellite Tools リポジトリーなど)。

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 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 6.3 でのホストのプロビジョニングを行う際に、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 7	x86_64
Red Hat Enterprise Linux 6	x86_64、i386

1.4.3. 設定管理

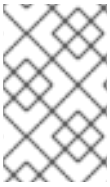
Satellite 6.3 で設定管理を行う際に、サポート対象となる Red Hat Enterprise Linux メジャーバージョンとハードウェアアーキテクチャーの組み合わせ。

表1.3 Puppet 4 サポート

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

表1.4 Puppet 3 サポート

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



注記

すべての Red Hat Satellite コンポーネントの使用は Red Hat Satellite の環境内でのみサポートされます。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。

第2章 CAPSULE SERVER の概要

Capsule Server は、**コンテンツフェデレーション** を提供し、**ローカライズされたサービス** を使用してホストを検出し、プロビジョニングを実行し、制御および設定を実行します。Capsule を使用して、地理的な様々な場所に Satellite デプロイメントを拡張することができます。このセクションでは、Capsule で有効にできる機能の概要と、そ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考慮事項などの詳細情報は『インストールガイド』を参照してください。

2.1. CAPSULE の機能

Capsule Server は 2 種類の機能を提供します。まず、Satellite Server からのコンテンツをミラーリングするように Capsule を設定できます。そして、ホスト管理に必要なサービスを実行する際にも Capsule を使用できます。

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 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)。
- **ホスト動作のデリバリー:** Capsule Server は、パッケージ更新など、ホストでスケジュールされている動作を実行します (ホストの Katello エージェントおよび Capsule の Qpid Dispatch Router で提供されます)。
- **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)。

以下はインフラストラクチャーおよびホスト管理サービスになります。

- **DHCP:** Capsule は DHCP サーバーとして機能します。または 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統合できます。
- **DNS:** Capsule は DNS サーバーとして機能します。または ISC DNS、Active Directory、BIND などの既存のソリューションと統合できます。
- **TFTP:** Capsule は TFTP サーバーとして使用として機能します。または UNIX ベースの TFTP サーバーと統合できます。
- **レルム:** Capsule は、Kerberos レルムまたはドメインを管理し、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、IdM、FreeIPA、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 マスター:** Capsule は、Puppet マスターを実行することにより設定管理サーバーとして管理できます。
- **Puppet 認証局:** Capsule は、ホストに証明書を提供する Puppet 認証局 (CA) として機能します。
- **ベースボード管理コントローラー (BMC):** Capsule は、ホストの電源管理を行います。

- **プロビジョニングテンプレートプロキシー:** 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。

2.2. CAPSULE のタイプ

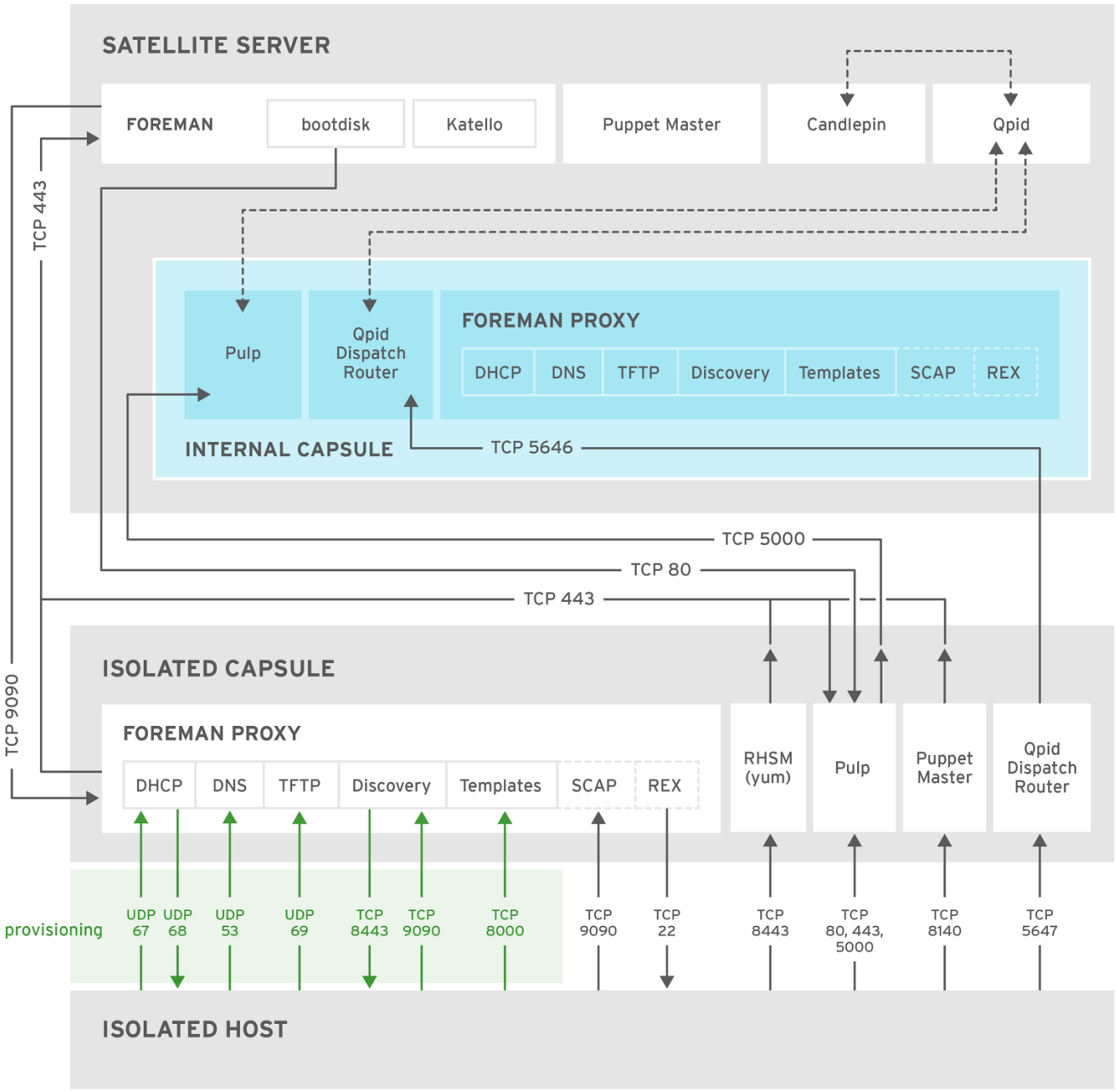
すべての Capsule 機能を一度に有効にする必要はありません。Capsule Server は限定した目的のために設定できます。一部の共通の設定には以下が含まれます。

- **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、管理対象ホストの単一接続点を提供することでホストの分離を有効にします。

2.3. CAPSULE のネットワーク

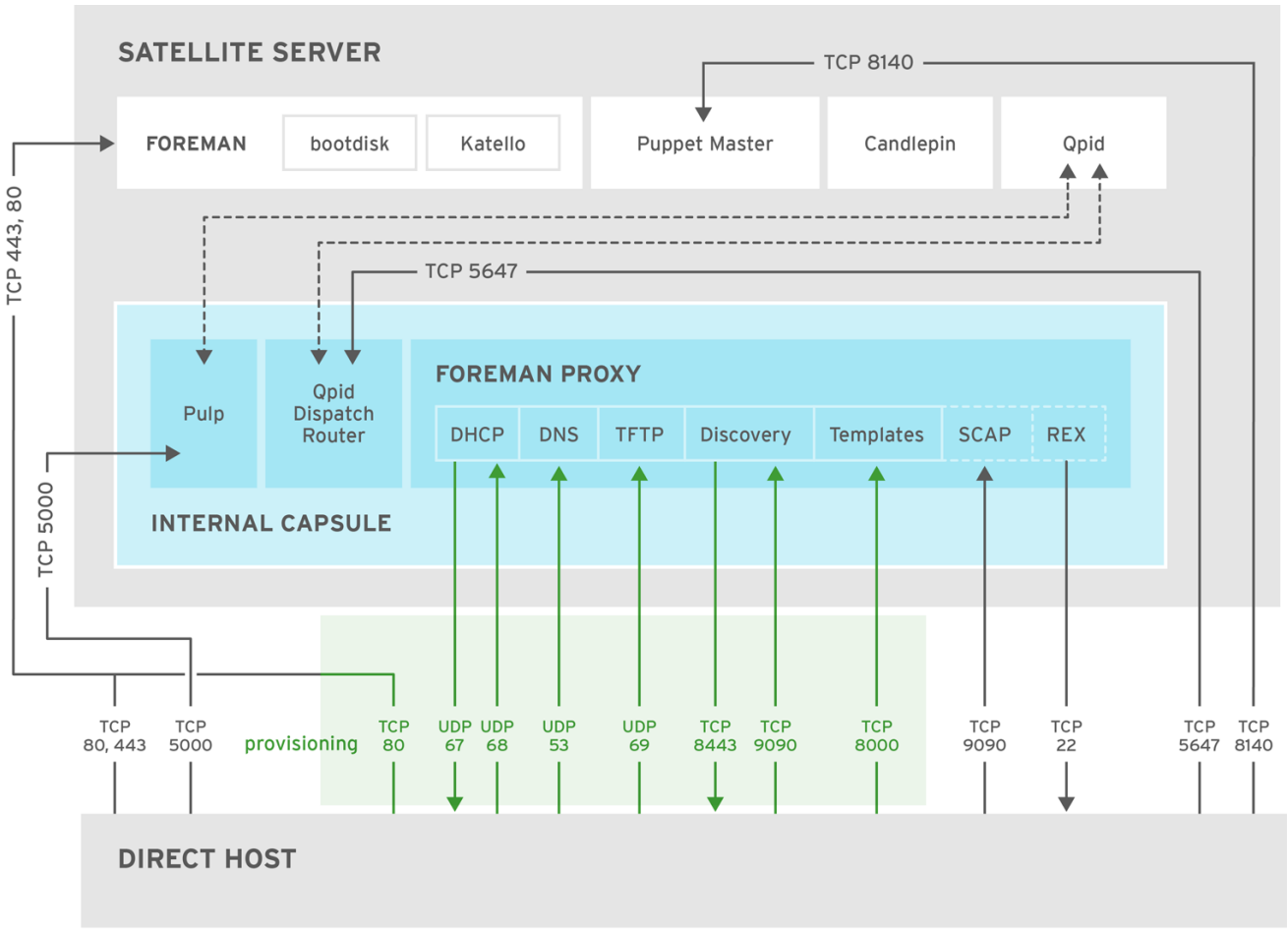
Capsule を分離する目的は、リモートネットワークセグメントでファイアウォールポートを Capsule 自体に開くだけで済むように、ホストの全ネットワーク通信のエンドポイントを 1 つだけ提供することにあります。以下の図は、分離した Capsule にホストが接続しているシナリオで、Satellite コンポーネントがどのように対話するかを示しています。

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てい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

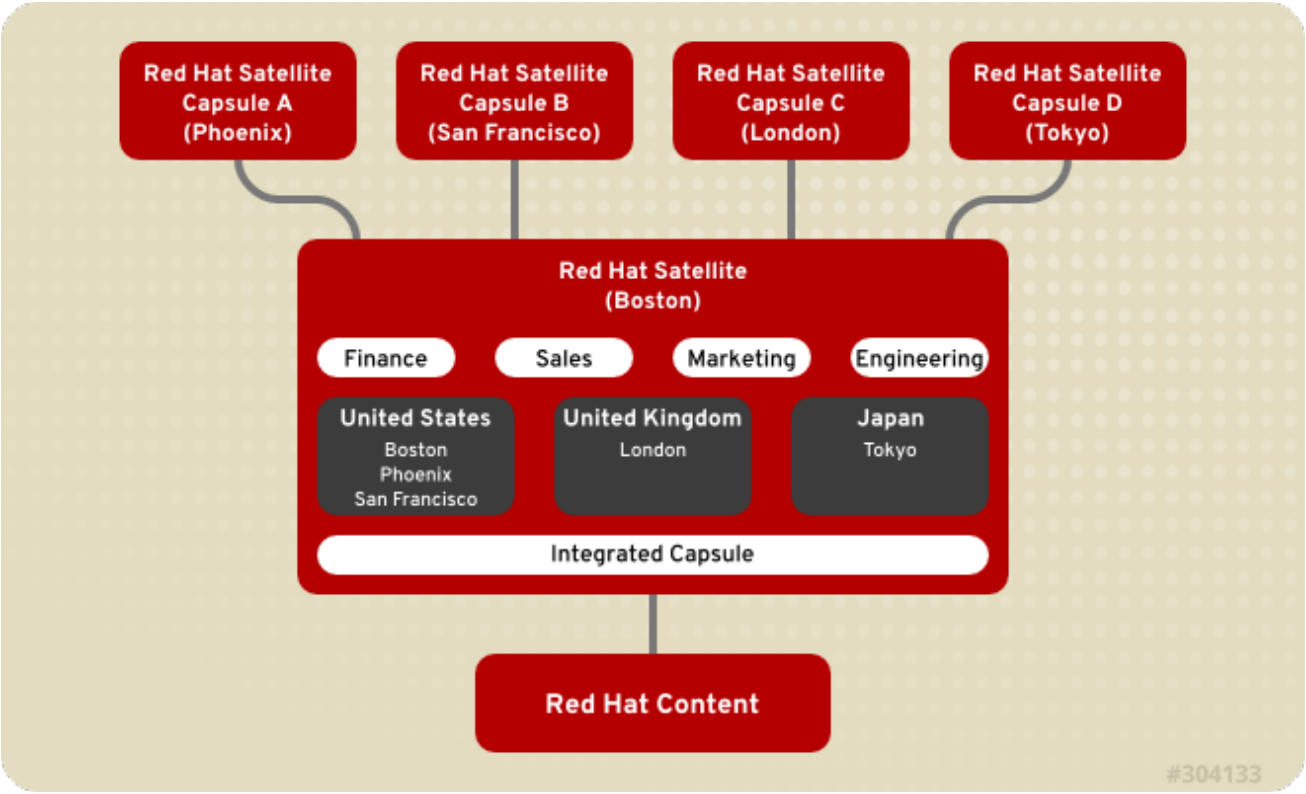


『インストールガイド』の「ポートとファイアウォール要件」セクションには、ホストベースのファイアウォールを設定して必要なポートを開く方法が説明されています。ポートのマトリックス表については、Red Hat ナレッジベースソリューション「Red Hat Satellite 6.3 List of Network Ports」 を参照してください。

第3章 組織、ロケーション、およびライフサイクル環境の設定

Red Hat Satellite 6 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 サーバーに、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、および日本) に 3 つの組織 (「Finance (財務)」、「Marketing (マーケティング)」、および「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。ユーザーは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。

Red Hat Satellite 6 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に基づいて Red Hat Satellite 6 リソースを論理グループに分割します。Red Hat Satellite 6 では複数の組織を作成して管理し、Red Hat 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。

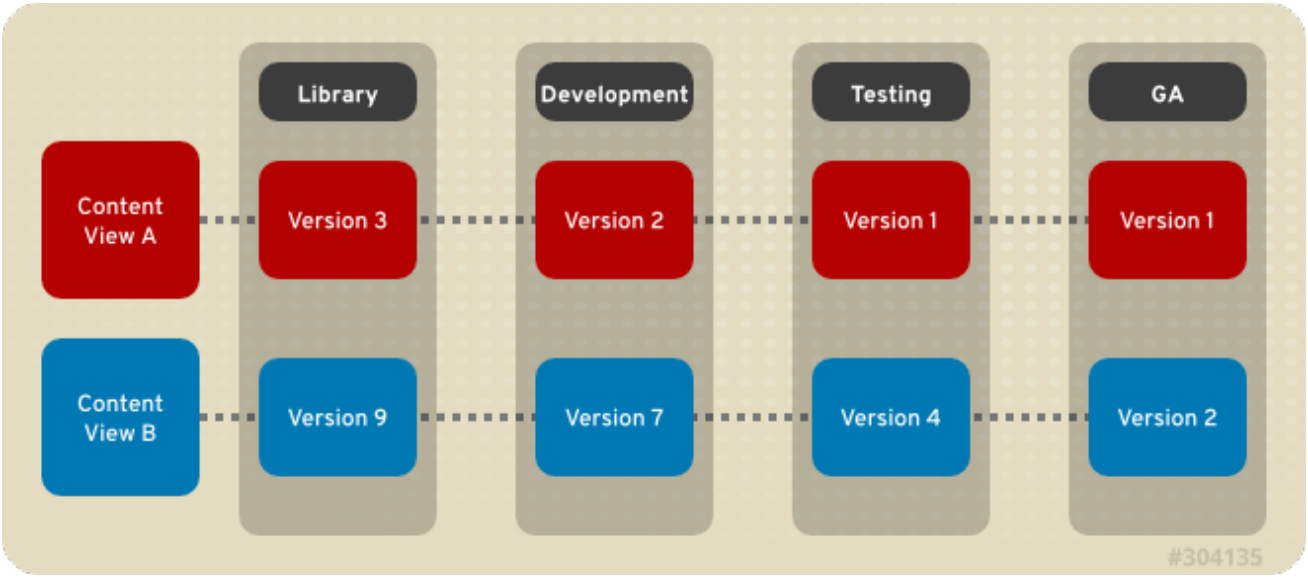
3.2. ロケーション

ロケーションは、地理的な場所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 Red Hat カスタマーポータル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** に分割されます。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。

4 つの環境を含む環境パス



第4章 ホスト分類の概念

Red Hat Satellite は、Capsule Server の物理的なトポロジーのほかにも、ホストを分類するためのいくつかの論理ユニットを提供します。ホストは各グループのメンバーになり、そのグループ設定を継承します。たとえば、プロビジョニング環境を定義する単純なパラメーターは以下のレベルで適用できます (パラメーターの使用については『**Puppet ガイド**』の「パラメーター」を参照してください)。

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

以下は Red Hat Satellite の主な論理グループです。

- **組織:** ホストの高位の論理グループです。組織により、コンテンツと設定が明確に区別されます。各組織には個別のサブスクリプションマニフェストが必要であり、それぞれを個別の Satellite Server 仮想インスタンスと見なすことができます。低位のホストグループが適用可能な場合には組織を使用しないようにしてください。

- **ロケーション:** 物理的な場所に対応するホストのグループ分けです。ロケーションは、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。ホストグループのパラメーターの完全一覧は『**Puppet ガイド**』の「パラメーター」を参照してください。設定の大半は、ホストを直接定義するのではなく、ホストグループレベルで設定することが推奨されます。そのため、「新規ホストを設定すること」は、そのホストを適切なホストグループに追加することと考えることができます。ホストグループはネスト化できるため、要件に最も適した構造を作成できます (「ホストグループの構造」を参照)。

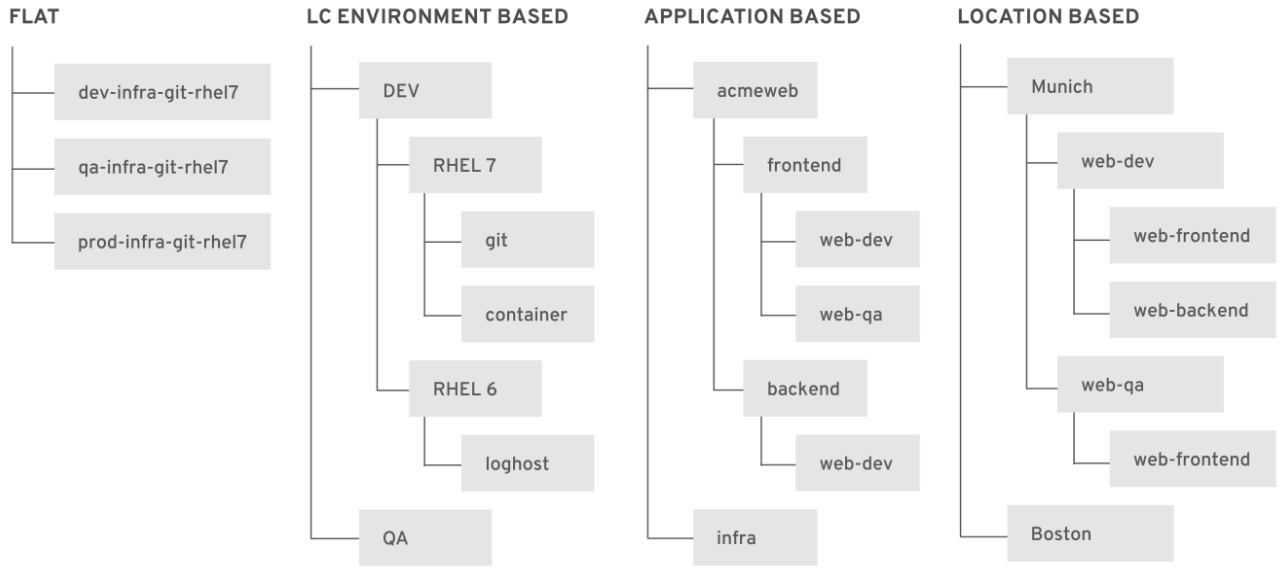
- **ホストコレクション:** サブスクリプションおよびコンテンツの管理目的で Satellite Server に登録されたホストは **コンテンツホスト** と呼ばれます。コンテンツホストはホストコレクションを使用して分類できるため、パッケージ管理やエラータインストールなどの一括処理が可能になります。

ロケーションおよびホストグループにはネストを使用できます。組織およびホストコレクションはフラット (非階層) 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 (本番)**」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。本章は、こ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「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リクエストを送ります。
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`** の場所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『**プロビジョニングガイド**』の「プロビジョニングワークフローの定義」を参照してください。

5.1.2. 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。

仮想マシンの前提条件:

- DHCP サーバーおよび TFTP サーバーにアクセス可能なネットワーク接続を設定しています。

ネットワークの要件:

- UDP ポート 67 および 68 にアクセスして仮想マシンを有効にし、ブートオプションで DHCP を利用できるようにしています。
- UDP ポート 69 にアクセスして、仮想マシンが Capsule で TFTP サーバーにアクセスできるようにしています。
- TCP ポート 80 にアクセスして、仮想マシンが 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しています。

クライアント要件:

- 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います。
- すべてのネットワークベースのファイアウォールで、サブネットのホストが Capsule にアクセスするように設定されています。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。
- ホストが属するサブネットで、DHCP が有効になっています。

Satellite 要件:

- Satellite インストーラーで、適切なサブネットがある DHCP が有効になっています。
- Satellite インストーラーを使用する TFTP を有効にしています。
- DHCP 予約の作成時に使用する Capsule を定義するように、サブネットが作成され、Web UI の DHCP Capsule に関連付けている必要があります。予約は常に Capsule から行われます。

5.2. キックスタート

Red Hat 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。詳細は『**Red Hat Enterprise Linux 5 インストールガイド**』の「キックスタートを使ったインストール」を参照してください。

5.2.1. 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 のインストールの場所を指定します。
2. 事前定義済みのパッケージをインストールします。
3. Red Hat 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` ファイルで Red Hat Satellite または Capsule instance を指定します。
6. Puppet を有効にして実行できるようにし、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サポート構造

Red Hat Content Delivery Network (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。このコンテンツは、Red Hat Subscription Manager を使用して登録したシステムから直接、または Red Hat Satellite 6 Web UI からアクセスできます。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 Content Delivery 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
    │               └── x86_64   5
    │                   └── sat-tools   6
    │                       └── 6   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** (本番環境の場合)、**eus** (延長アップデートサポートの場合) 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、サブスクリプションマニフェストでも使用されます。

パート II. SATELLITE 6 デプロイメントの計画

第7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 6 デプロイメントの計画時に検討される一般的なトピックの概要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。たとえば、お客様のシナリオ例 (Satellite 6.1 に特化) に基づいた実装は、Red Hat ナレッジベースソリューションの「10 Steps to Build an SOE: How Red Hat Satellite 6 Supports Setting up a Standard Operating Environment」を参照してください。

7.1. 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、『インストールガイド』に従って Red Hat Satellite Server のインスタンスを専用 Red Hat Enterprise Linux 7 サーバーにインストールします。ここで説明されている「インストールのための環境準備」および「大規模デプロイメントに関する考慮事項」について検討してください。

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

サブスクリプションマニフェストは、サブスクリプション情報が含まれる暗号化されたファイルのセットです。Satellite Server はこの情報を使用して 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できます。サブスクリプションマニフェストを作成し、インポートする方法は、**『コンテンツ管理ガイド』**の「サブスクリプションの管理」を参照してください。

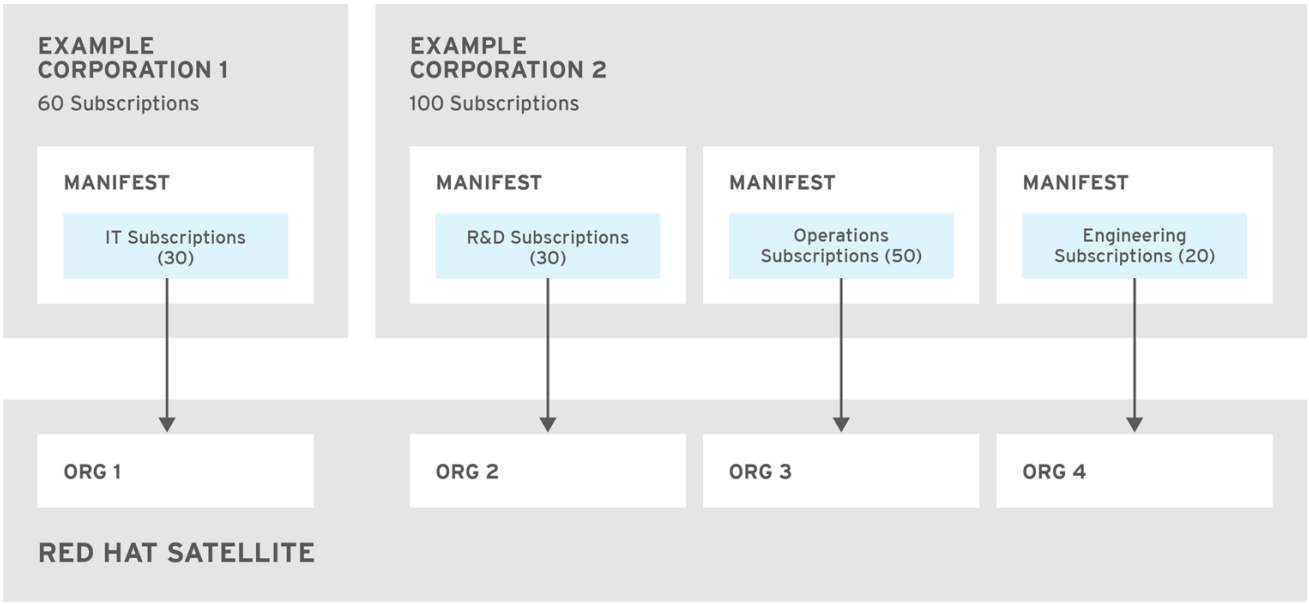
Red Hat Satellite 6 では、Satellite に設定した各組織に対して、マニフェストが 1 つ必要になります。Satellite 6 の組織機能を使用して、Red Hat Network アカウントで、インフラストラクチャーを分けて管理する予定がある場合は、必要に応じて 1 つのアカウントが所有するサブスクリプションを、各組織のマニフェストに割り当てます。

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。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関わらず、RPM のサブスクリプションへのアクセスがない場合は、Red Hat Enterprise Linux Smart Management アドオンを使用する必要があります。詳細は「Smart Management Add-On」を参照してください。

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

複数のマニフェストを含む Satellite Server



サブスクリプションマニフェストの作成時に、以下が行われます。

- 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 Red Hat 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もできます。

Red Hat Satellite 6.3 には、マニフェストに将来の日付が指定されたサブスクリプションを使用する機能が追加されました。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、将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ても、リポジトリーへのアクセスが妨げられることはありません。

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。

7.2. ロケーションおよびトポロジー

このセクションでは、Satellite 6 デプロイメントシナリオを指定するのに役立つ一般的な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については「8章*共通のデプロイメントシナリオ*」にその一覧が記載されています。以下はよくある質問です。

- **必要な Capsule Server の数は?:** 組織が機能する地理的な場所の数が Capsule Server の数であると考えることができます。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 のベースシステムをプロビジョニングするのに使用できます。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 管理者の管理対象外である場合があります。その場合は、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があるか?:** 切断された Satellite は一般的なデプロイメントシナリオで使用されます (「インターネットから切断された Satellite」 を参照)。切断された Satellite で Red Hat コンテンツの更新が頻繁に必要な場合には、Satellite 間の同期を図るための追加の Satellite インスタンスについて計画してください。

- **ホストに必要なコンピュートリソースは?** ベアメタルホストのプロビジョニングのほかに、Satellite 6 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。コンピュートリソースが異なる場合のプロビジョニングの詳細は、『プロビジョニングガイド』を参照してください。

7.3. コンテンツソース

サブスクリプションマニフェストは Satellite Server からアクセスできる Red Hat リポジトリーを決定します。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、Puppet モジュール、Docker イメージ、または OSTree スナップショット) によって異なります。

RPM パッケージのみを含む **yum** リポジトリーとして設定されるリポジトリーは、同期時間と保存スペースを節約するための新規ダウンロードポリシー設定を利用できます。この設定は、**即時**、**オンデマンド**、および**バックグラウンド**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。**バックグラウンド**設定は、初期の同期後にダウンロードを完了することで時間を節約します。コンテンツソースのセットアップに関する詳細な説明は『**コンテンツ管理ガイド**』の「Red Hat 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。

7.4. コンテンツのライフサイクル

Satellite 6 は、コンテンツライフサイクルを詳細に管理する各種機能を提供します。**ライフサイクル環境** は、コンテンツライフサイクルのステージを表し、**コンテンツビュー**は、フィルターが設定されたコンテンツのセットであり、コンテンツの定義済みのサブセットとみなすことができます。コンテンツビューをライフサイクル環境に関連付けると、定義した方法でホストに対してコンテンツを利用可能にできます (プロセスの仮想化については図1.2「Red Hat Satellite 6 におけるコンテンツのライフサイクル」を参照)。コンテンツ管理プロセスの詳細は『**コンテンツ管理ガイド**』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、ホストに利用可能にする前の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、本番ステージに直接プロモートされます。このアプローチでは、複雑さの面で制限がありますが、ホスト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 (本番)**) で構成されており、詳細なテストを可能にしますが、これには追加の作業が必要です。

- **アプリケーション固有のライフサイクル環境パス:** 各アプリケーションには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、Puppet 設定の専用のコンテンツビューを設定することができます。このコンテンツビューを複数のホストタイプの複合コンテンツビューに組み込むと、他のホストコンテンツよりも高い頻度で Puppet 設定を更新できます。

- **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。

7.5. 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。1 つ目は、デフォルトの方法である **Goferd サービスエージェント** です。2 つ目は、Satellite 6.2.11 以降で利用可能な置換で、**リモート実行** を介した管理となります。

- **Goferd サービスエージェント:** このサービスは Satellite サーバーとやり取りし、パッケージのインストール、更新、レポート作成時の主なタスクとなり、**Katello-agent** RPM 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自動的に起動します。
- **リモート実行:** SSH トランスポートを介したリモート実行では、パッケージのインストール、更新、または削除、そして設定管理エージェントのブートストラップと、Puppet 実行のトリガーが可能になります。Satellite Server サーバでは、リモート実行がデフォルトで有効ですが、Capsule Server とコンテンツホストのデフォルトでは無効になっているため、手動で有効にする必要があります。
- **コンテンツのデプロイメント方法を検討する** リモート実行を使用すると、**Goferd** サービスを無効にして、コンテンツホストのメモリーと CPU 負荷を減らすことができます。ただし、リモート実行を全コンテンツホストで手動で設定しないと、**Goferd** を置き換えることができません。この設定プロセスは、デプロイしたコンテンツホストの数が多いシステムでは大掛かりとなります。詳細は『**ホストの管理**』の「Goferd を使用せずにホスト管理の設定」を参照してください。

7.6. プロビジョニング

Satellite 6 は、プロビジョニングテンプレート、Puppet を使用した設定管理、ホストロールの標準化されたプロビジョニングのためのホストグループなど、ホストのプロビジョニングの自動化に役立つ機能を提供します。プロビジョニングのワークフローの詳細は『**プロビジョニングガイド**』の「プロビジョニングワークフローの概要」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。

7.7. 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 6 コンポーネントへのアクセスを制御できます。ロールベースの認証は、ユーザーにとって不要なオブジェクトを非表示にする方法として理解することができます。

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

- **アプリケーションまたはインフラストラクチャーの一部に関連するロール**たとえば、オペレーティングシステムとなる 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、および本番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、または Access Insights 管理者など。

カスタムロールを定義する際に、以下の推奨事項を考慮してください。

- **予想されるタスクおよび責任を定義する:** 所定のロールからアクセスできる Satellite インフラストラクチャーのサブセットと、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 6 は、単独またはロールの組み合わせの一部として使用できる数多くのサンプルロールを提供します。既存のロール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の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』 を参照してください。このガイドでは、外部の認証ソースを設定する方法が記載されています。

7.8. 追加のタスク

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

- **既存ホストのインポート:** Satellite 6 で管理していなかった既存のホストがある場合は、そのホストを Satellite Server にインポートできます。通常、この手順には Red Hat Satellite 5 からの移行で 1 つのステップが必要になります。詳細は『移行ガイド』を参照してください。 移行プロセスの概要については、Red Hat ナレッジベースソリューション「Red Hat Satellite 5 から Satellite 6 への移行」を参照してください。

- **ベアメタルホストの検出:** Satellite 6 検出プラグインは、プロビジョニングネットワーク上にある不明なホストのベアメタルを自動的に検出できます。これらの新規ホストは、シリアル ID、ネットワークインターフェース、メモリー、ディスク情報など、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『ホストの管理』の「Satellite でのベアメタルホストの検出」を参照してください。

- **バックアップ管理:** Satellite Server のバックアップおよび障害回復の手順を参照してください (『Red Hat Satellite の管理』 の「Satellite Server および Capsule Server のバックアップと復

元」を参照)。リモート実行を利用すると、管理対象ホストで繰り返されるバックアップタスクを設定することもできます。リモート実行の詳細は『**ホストの管理**』の「ホストでのリモートジョブの実行」を参照してください。

- **セキュリティー管理:** Satellite 6 は、セキュリティー管理を様々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 OpenSCA 統合、更新およびセキュリティーコンプライアンスのレポート作成、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 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 6 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 6 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 6 は、大半の Web UI の手順と同等の CLI を提供する Hammer というコマンドラインツールを提供します。さらに、Satellite API へのアクセスを使用して、選択したプログラミング言語で自動化スクリプトを作成できます。詳細は『Hammer CLI ガイド』 および 『API ガイド』を参照してください。

第8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。

8.1. 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 つの地理的な場所にある 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。

8.2. サブネットが分類されている単一ロケーション

Red Hat Satellite が 1 つの地理的な場所 (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。この分離は、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行できますが、分離したサブネットを作成するのに使用する Capsule は 1 つだけにすることが推奨されます。次に、この Capsule を使用して、分離されたネットワークでホストおよびコンピュートリソースを管理し、それらが Capsule にアクセスすれば、プロビジョニング、設定、エラータ、およびその他の一般的な管理ができるようにします。サブネット設定の詳細情報は『ホストの管理』を参照してください。

8.3. 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、設定しやすいものになります。このアプローチを示した図については図1.1「Red Hat Satellite 6 システムアーキテクチャー」を参照してください。

8.4. インターネットから切断された 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、パッケージなどのコンテンツをプロビジョニングできます。この場合、Satellite Server にはインターネットへの直接のアクセスはありませんが、他のインフラストラクチャーコンポーネントのレイアウトは影響を受けません。切断された Satellite でインストールまたはアップグレードを行う方法は『インストールガイド』を参照してください。

切断された Satellite Server にコンテンツをインポートする方法は 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 サーバーに展開します。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 (現在は、Red Hat Enterprise Linux と RHEL-OSP7、RHDS、Red Hat Enterprise Linux for Real Time などのレイヤー化された製品のみ)。コンテンツ ISO を切断された Satellite にインポートする方法は**『コンテンツ管理ガイド』**の「非接続の Satellite Server へのコンテンツ ISO のインポート」を参照してください。

- **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 間の同期を設定する方法は『**コンテンツ管理ガイド**』の「Satellite Server 間でのコンテンツ同期」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。

8.5. CAPSULE と外部サービス

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

付録A SATELLITE で提供され、必要になるテクニカルユーザー

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

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

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

表A.1 Satellite で提供され、必要になるテクニカルユーザー

ユーザー名	UID	Flex UID	Home	Shell
qpidd	なし	はい	/var/lib/qpidd	/sbin/nologin
foreman	なし	はい	/usr/share/foreman	/sbin/nologin
unbound	なし	はい	/etc/unbound	/sbin/nologin
foreman-proxy	なし	はい	/usr/share/foreman-proxy	/sbin/nologin
puppet	52	いいえ	/var/lib/puppet	/sbin/nologin
postgres	26	いいえ	/var/lib/pgsql	/bin/bash
mongodb	184	いいえ	/var/lib/mongodb	/sbin/nologin
apache	48	いいえ	/usr/share/httpd	/sbin/nologin
tomcat	91	いいえ	/usr/share/tomcat	/bin/nologin
qdrouterd	なし	はい	なし	/sbin/nologin
saslauth	76	はい	なし	/sbin/nologin

付録B 用語集

この用語集では、Red Hat Satellite 6 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 (Answer ファイル)

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

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 6 では、選択したホストを管理する BMC インターフェースを作成できます。

Boot Disk (ブートディスク)

PXE なしのプロビジョニングに使用される ISO イメージです。この ISO により、ホストは Satellite Server に接続でき、インストールメディアを起動し、オペレーティングシステムをインストールできます。ブートディスクの種類には、**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 ノードとしての機能する) ために、そしてローカライズされた他のサービス (Puppet マスター、**DHCP**、**DNS**、**TFTP** など) を実行するために、Red Hat Satellite 6 デプロイメントで使用できる追加サーバーです。Capsule は、複数の地理的な場所での Satellite のデプロイメントに役立ちます。アップストリームの Foreman 用語では、Capsule は Smart Proxy (スマートプロキシー) 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。これには、管理が必要なすべてのリソースと、それらリソース間の依存関係を一覧表示します。カタログは Puppet マニフェストから Puppet マスターによってコンパイルされ、また Puppet エージェントのデータが使用されます。

Candlepin

サブスクリプションの管理を行う Katello 内のサービスです。

Compliance Policy (コンプライアンスポリシー)

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

Content (コンテンツ)

Satellite がホストに配信するものの総称です。ソフトウェアパッケージ (RPM ファイル)、Puppet モジュール、Docker イメージ、または OSTree スナップショットが含まれます。コンテンツはライブラリーに同期され、コンテンツビューを使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。

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

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

Content Host (コンテンツホスト)

コンテンツとサブスクリプションに関連するタスクを管理するホストの一部です。

Content View (コンテンツビュー)

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

Discovered Host (検出ホスト)

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

Discovery Image (検出イメージ)

プロビジョニングプロセスの開始前に、初期のハードウェア情報を取得し、Satellite Server と通信するためにホスト上で PXE で起動した Red Hat Enterprise Linux ベースの最小オペレーティングシステムを指します。

Discovery Plug-in (検出プラグイン)

プロビジョニングネットワーク上の不明なホストのベアメタルの自動検出を可能にします。このプラグインは、Satellite Server 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構成されます。

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のに使用するマークです。Satellite 6 Web UI の **コンテンツ > Docker タグ** 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。

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

Facter

Facter は一種のプログラムであり、それ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 Red Hat Satellite 6 コンポーネントです。Foreman は Red Hat Satellite 6 に対応する主なアップストリームのコンポーネントです。

Foreman Hook

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

Full Host Image (完全ホストイメージ)

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

Generic Image (汎用イメージ)

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

Hammer

Red Hat Satellite 6 を管理する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。また、 Hammer は対話式シェルも提供します。

Host (ホスト)

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

Host Collection (ホストコレクション)

エラータのインストールなど、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。Satellite 5 システムグループに相当します。

Host Group (ホストグループ)

ホストをビルドするためのテンプレートです。ホストグループは、ホストグループメンバーが継承する、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の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。すべてのジョブはテンプレートで定義されます。Satellite 5 のリモートコマンドに似ています。

Job Template (ジョブテンプレート)

ジョブのプロパティーを定義します。

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

Location (ロケーション)

物理的な場所を表すデフォルト設定のコレクションです。

Library (ライブラリー)

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

Life Cycle Environment (ライフサイクル環境)

コンテンツホストによって使用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。

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

コンテンツビューがプロモートされるライフサイクル環境の順序です。たとえば、プロモートパス (たとえば、開発からテスト、テストから本番へ) で、コンテンツビューをプロモートすることがで

きます。Red Hat Satellite 5 では、チャンネルのクローン作成がこの概念を実行するものとなってきました。

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

Red Hat カスタマーポータルから Red Hat Satellite 6 にサブスクリプションを送信するメカニズムです。これは Red Hat Satellite 5 で使用される証明書と機能的に似ています。
「Puppet マニフェスト」と混同しないようにしてください。

OpenSCAP

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

Organization (組織)

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

OSTree

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

Parameter (パラメーター)

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

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

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

Permission (パーミッション)

Satellite インフラストラクチャーの選択された部分 (リソースタイプ) に関連するアクションを定義します。各リソースタイプはパーミッションのセットに関連付けられます。たとえば、**Architecture** というリソースタイプ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)

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

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

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

Pulp

リポジトリーおよびコンテンツ管理を行う Katello 内のサービスです。

Pulp Node (Pulp ノード)

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

Puppet

Satellite 6 の設定および管理コンポーネントです。

Puppet Agent (Puppet エージェント)

設定変更をホストに適用するホスト上で実行されるサービスです。

Puppet Environment (Puppet 環境)

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

Puppet Manifest (Puppet マニフェスト)

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。 詳細と使用例は『**Puppet ガイド**』の「Puppet モジュールの内容の検証」を参照してください。
「マニフェスト (サブスクリプションマニフェスト)」と混同しないようにしてください。

Puppet Master (Puppet マスター)

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

Puppet Module (Puppet モジュール)

ユーザー、ファイル、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 6 Web UI では、これらのジョブを **モニター > 再帰論理** で確認できます。

Registry (レジストリー)

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

Red Hat Access Insights

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

Repository (リポジトリー)

コンテンツのコレクションにストレージを提供します。

Resource Type (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。

Role (ロール)

ホストなどのリソースのセットに適用されるパーミッションのコレクションを指します。ロールはユーザーおよびユーザーグループに割り当てることができます。Satellite は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ための独自の Answer ファイルがあります。

Smart Proxy (スマートプロキシー)

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

Smart Variable (Smart 変数)

Puppet モジュールのクラスで使用される設定値です。

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

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

Subnet Image (サブネットイメージ)

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

Subscription (サブスクリプション)

Red Hat からコンテンツおよびサービスを受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 6 ライブラリーにミラーリングすることを指します。

Synchronization Plan (同期プラン)

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

User Group (ユーザーグループ)

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

User (ユーザー)

Red Hat Satellite を使用できるように登録されたすべてのユーザーを指します。認証および認可については、外部リソース (LDAP、Identity Management、または Active Directory) または Kerberos を使用し、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 6 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。