Red Hat Satellite 6.3

プロビジョニングガイド

Red Hat Satellite Server で物理ホストと仮想ホストのプロビジョニングを実行するためのガイド

エディッション 1.0

Last Updated: 2018-07-09

Red Hat Satellite 6.3 プロビジョニングガイド

Red Hat Satellite Server で物理ホストと仮想ホストのプロビジョニングを実行するためのガイド
エディッション 1.0

法律上の通知

概要

Red Hat Satellite プロビジョニングガイドは、物理ホストと仮想ホストのプロビジョニングについてのシナリオベースの文書です。本書には、必要なネットワークトポロジーのセットアップや、必要なサービスの設定、およびネットワーク上でのホストのプロビジョニングに必要な他の設定に関する情報すべてが記載されています。本書は、ネットワークの十分な知識とスキルを持つ Satellite 管理者を主な対象としています。

目次

第1章 はじめに

プロビジョニングとは、物理または仮想マシンを完全設定して使用可能なオペレーティングシステムを備えた状態にするプロセスです。Red Hat Satellite は、多数のホストの細かいプロビジョニングを定義して自動化する機能を提供します。プロビジョニングはいくつもの方法で実行できます。たとえば、Satellite Server の統合 Capsule や外部 Capsule Server は、PXE ベースと PXE を使用しない方法の両方を使ってベアメタルシステムをプロビジョニングできます。同様に Satellite Server は、API 経由で特定のプロバイダーからクラウドインスタンスをプロビジョニングできます。これらのプロビジョニング方法は Red Hat Satellite 6 アプリケーションライフサイクルの一部で、これによって新規システムの作成、それらの管理、Red Hat コンテンツでシステムを最新の状態に維持することができます。

1.1. アプリケーションライフサイクルでのシステムのプロビジョニング

アプリケーションライフサイクルは、特定のシステムとそのソフトウェアを特定の段階でプロビジョニングする方法を定義します。たとえば、アプリケーションライフサイクルが単純な場合には、以下に示す 2 つのステージのみが含まれます。

- Development (開発)
- Production (実稼働)

より複雑なアプリケーションライフサイクルの場合には、テストやベータリリースなどのさらに多くのステージが含まれることがあります。その場合、追加のステージがアプリケーションライフサイクルに加わります。

- Development (開発)
- Testing (テスト)
- Beta Release (ベータリリース)
- Production (実稼働)

Satellite Server は、アプリケーションライフサイクルのすべてのステージの新規ホストのプロビジョニングも可能にします。たとえば、製品開発用のホストのセットを作成するには、**`Development (開発)`** 環境内でホストのセットをプロビジョニングします。同様に、製品のテスト用のホストのセットを作成するには、**`Testing (テスト)`** 環境内でホストのセットをプロビジョニングします。

1.2. プロビジョニングタイプの定義

Red Hat Satellite 6 ではホストのプロビジョニングのための各種の方法を提供しています。これには以下が含まれます。

ベアメタルプロビジョニング

Satellite Server は、主としてベアメタルシステムのプロビジョニングを PXE ブートおよび MAC アドレスの特定によって実行します。システム管理者は新規ホストのエントリーを作成し、プロビジョニングされる物理ホストの MAC アドレスを指定します。また、システム管理者は Satellite Server の Discovery サービスを使用するために空のホストを起動し、このサービスによりプロビジョニング可能なホストのプールが作成されます。システムは PXE を使用しない方法で起動したり、プロビジョニングしたりすることもできます。

クラウドプロバイダー

Satellite Server はプライベート (Red Hat OpenStack Platform) およびパブリック (Amazon EC2) クラウドプロバイダーに接続します。これにより、クラウド環境で保存されたイメージから新規インスタンスをプロビジョニングする方法を実行できます。この方法には、使用するハードウェアのプ

ロファイル (またはフレーバー) を定義する機能も含まれます。

仮想化インフラストラクチャー

Satellite Server は、Red Hat Virtualization および VMware などの仮想化インフラストラクチャーサービスに接続します。これにより、仮想イメージテンプレートから、またはベアメタルプロバイダーと同じ PXE ベースのブート方法を使用して、仮想マシンをプロビジョニングする方法を実行できます。

Linux コンテナー

Satellite Server には、Red Hat Enterprise Linux Atomic Server でコンテナーを作成し、管理する機能があります。

1.3. シナリオの定義

本書では、『コンテンツ管理ガイド』のシナリオに従っています。本書では、**ACME** というソフトウェア開発会社が Red Hat Satellite 6 を使用し、各種のプロビジョニングタイプを使用して新規システムをプロビジョニングすることを想定しています。本書では、プロビジョニングを実行するために ACME が使用できる複数のユースケースシナリオを提供します。

この段階では、ACME の Satellite Server は Red Hat のコンテンツ配信ネットワークのコンテンツおよびその他のソースと同期しています。これと同様に、お使いになる Satellite Server にもプロビジョニングの実行前に同期されたコンテンツが含まれている必要があります。そのため、まず『コンテンツ管理ガイド』のシナリオを参照してから、本書を参照することをお勧めします。

注記

『コンテンツ管理ガイド』のステップを事前に実行せずに本書のサンプルを使用する場合は、付録A *プロビジョニングのサンプル用の初期化スクリプト*のスクリプトを使用して、これらのサンプルに必要な Red Hat コンテンツをインポートしてください。

本書では、Red Hat Satellite 6 Web UI または CLI ツール (**hammer**) を使用する手順を説明します。Red Hat Satellite 6 との対話で優先される方法に応じてこれらのいずれかを使用します。CLI を使用する場合で **hammer** コマンドを実行するたびに認証の詳細情報を指定したくない場合は、ローカルユーザーの CLI 設定ファイルを作成します。

```
mkdir ~/.hammer
cat > .hammer/cli_config.yml <<EOF
:foreman:
    :host: 'https://satellite.example.com/'
    :username: 'admin'
    :password: 'p@55w0rd!'

EOF
```

重要

本書の **hammer** コマンドのすべての使用箇所では、この設定ファイルで追加する認証の詳細情報が省略されています。

1.4. 本章のまとめ

本章では、Red Hat Satellite 6 のコンテキストでプロビジョニングの概念を説明し、Red Hat Satellite 6 アプリケーションライフサイクルにおけるプロビジョニングの位置付けについても説明しました。また、Red Hat Satellite 6 で使用される各種のプロビジョニングタイプについても簡単に説明しました。

本書の後の章では、プロビジョニングの複数のシナリオについて説明します。

次章では、組織およびロケーションを使用して定義するプロビジョニングのコンテキストについて扱います。

第2章 プロビジョニングコンテキストの設定

この章では、新規ホストのプロビジョニングを開始する前に Satellite Server に必要な基本的な要素をいくつか定義します。これには、プロビジョニングコンテキストを使用して配置の方法を定義することが含まれます。

2.1. プロビジョニングコンテキストの定義

プロビジョニングコンテキストは、ホストとその関連付けられたリソースに使用する組織およびロケーションを定義します。組織およびロケーションの組み合わせにより、システムの所有者とシステムが置かれる場所が定義されます。

組織は、所有者、目的、コンテンツ、セキュリティーレベルその他の区分に基づいて Red Hat Satellite 6 リソースを論理グループに分類します。Red Hat Satellite 6 では、複数の組織を作成および管理し、リソースをそれぞれの組織に割り当てることができます。これにより、Satellite Server において特定の組織内でホストをプロビジョニングし、その組織に関連付けられたリソースのみを使用できるようになります。組織についての詳細は、**『コンテンツ管理ガイド』**の 「組織の作成」 を参照してください。

ロケーションは、各種リソースを分類し、ホストを割り当てる方法を提供する点で組織に似ています。ただし、物理的または地理的な設定をベースとしている点で組織とは異なっています。さらに、ユーザーはロケーションを階層的にネストできます。例として、以下のロケーションマップを参照してください。

- United States
  - New York
    - Datacenter 1
    - Datacenter 2
    - Datacenter 3
  - San Francisco
    - Datacenter 4
    - Datacenter 5
    - Datacenter 6
- Japan
  - Tokyo
    - Datacenter 7
    - Datacenter 8
    - Datacenter 9

上記の例では、3 都市にある 9 つのデータセンターが使用されています。Satellite Server は各種リソースを管理し、各データセンターでホストをプロビジョニングします。

このシナリオでは、単純なプロビジョニングコンテキストを使用しています。『コンテンツ管理ガイド』 には、本書のシナリオで使用する ACME の組織を作成する方法が説明されています。またロケーションの作成方法についても説明されています。

2.2. 組織の作成

以下では、組織の作成方法を示しています。この手順については、『コンテンツ管理ガイド』にも記載されており、同ガイドの組織の作成シナリオに従う場合は以下の手順に従う必要はありません。組織に変更を一部加える必要があり、その際に組織のプロパティーを編集する必要があることに注意してください。

Web UI を使用する場合

管理 > 組織 に移動します。Satellite Server が現在管理している組織の一覧が表示されます。

新規組織 をクリックします。

以下の 3 つのセクションから構成される作成ウィザードが表示されます。

組織の作成

組織の基本的な詳細情報を指定します。これには以下が含まれます。

- **名前:** 組織のテキスト形式の名前。例:**ACME**
- **ラベル:** 組織の一意の ID。これは、コンテンツ保管用のディレクトリーなどの特定のアセットを作成およびマッピングする場合に使用されます。文字、数字、アンダースコアおよびダッシュを使用し、スペースは使用しないでください。例: **ACME**
- **説明:** 組織のテキスト形式の説明 (オプション)。例:**`Our example organization`**

ホストの選択

すべてのホストには組織が設定される必要があります。ただし、場合によってはホストが孤立することがあります。以下はその例になります。

- 古い組織を削除するとそのホストが孤立する可能性がある。
- ホストがプロビジョニングされていないが Puppet でインポートされている。
- ホストがプロビジョニングされていないが **`subscription-manager`** で登録されている。

上記の場合には、必要に応じて孤立したホストを新たに作成した組織に割り当てることができます。**すべてを割り当て** をクリックしてすべての孤立したホストを割り当てるか、または**手動割り当て** をクリックして割り当てる予定の孤立したホストを選択します。ACME のシナリオでは孤立したホストは存在しないことになっているため、**編集に進む** をクリックして **プロパティーの編集** セクションに移動します。

プロパティーの編集

このセクションでは、組織に特定のインフラストラクチャーリソースを割り当てます。これには、ネットワークリソース、インストールメディア、キックスタートテンプレートその他のパラメーターが含まれます。**管理 > 組織** に移動し、編集する組織を選択すると、この画面にいつでも戻ることができます。

重要

ACME 組織には、シナリオ用に以下のリソースが割り当てられているはずです。

- **Capsule (スマートプロキシー):** Satellite Server と同じホスト名を使用する Satellite Server の統合 Capsule。
- **メディア:** 『コンテンツ管理ガイド』のコンテンツと同期した Red Hat Enterprise Linux 7.2 キックスタートツリー。
- **プロビジョニングテンプレート:** 選択されたすべてのテンプレート。
- **パーティションテーブル:** 選択されたすべてのテーブル。
- **ドメイン:** この組織が使用し、管理するドメイン。
- **環境: `production` (実稼働)** および 『コンテンツ管理ガイド』で作成された環境を含むすべての Puppet 環境。

上記のリソースについて、「プロパティーの編集」セクションでそれぞれのリソースタイプを確認します。

組織の作成後に、**送信** をクリックします。

CLI を使用する場合

```
# hammer organization create --name "ACME" --label "ACME" \
--description "Our example organization"
```

これにより、組織のサンプルが作成されます。

2.3. ロケーションの作成

以下の手順では、ロケーションの作成方法を示します。ロケーションの作成は、新規ホストとそれらのリソースのプロビジョニングコンテキストの定義に役立ちます。

Web UI を使用する場合

管理 > ロケーション に移動します。これにより、Satellite Server が現在管理しているロケーションの一覧が表示されます。

新規ロケーション をクリックします。

以下の 3 つのセクションから構成される作成ウィザードが表示されます。

組織の作成

ロケーションの基本的な詳細情報を指定します。これには、以下が含まれます。

- **親:** このロケーションの親ロケーション。これにより、ロケーションの階層が作成されます。Satellite の場合はロケーションがないため、これを空白のままにして上位のロケーションを作成します。
- **名前:** ロケーションのテキスト形式の名前。例:**`New York`**
- **説明:** 組織のテキスト形式の説明 (オプション)。例:**`Our example location`**

ホストの選択

すべてのホストにはロケーションが設定される必要があります。ただし、場合によってはホストが孤立することがあります。以下はその例になります。

- 古いロケーションを削除するとそのホストが孤立する場合がある。
- ホストがプロビジョニングされていないが Puppet でインポートされている。
- ホストがプロビジョニングされていないが **subscription-manager** で登録されている。

上記の場合には、必要に応じて孤立したホストを新たに作成したロケーションに割り当てることができます。**すべてを割り当て** をクリックしてすべての孤立したホストを割り当てるか、または**手動割り当て** をクリックして割り当てる予定の孤立したホストを選択します。ACME のシナリオでは孤立したホストは存在しないことになっているため、**編集に進む** をクリックして **プロパティーの編集** セクションに移動します。

プロパティーの編集

このセクションでは、ロケーションに特定のインフラストラクチャーリソースを割り当てることができます。これには、ネットワークリソース、インストールメディア、キックスタートテンプレートその他のパラメーターが含まれます。**管理 > ロケーション** に移動し、編集するロケーションを選択すると、この画面にいつでも戻ることができます。

重要

New York のロケーションには、このシナリオ用に以下のリソースが割り当てられているはずです。

- **Capsule (スマートプロキシー):** Satellite Server と同じホスト名を使用する Satellite Server の統合 Capsule。
- **メディア:** 『コンテンツ管理ガイド』のコンテンツと同期した Red Hat Enterprise Linux 7.2 キックスタートツリー。
- **プロビジョニングテンプレート:** 選択されたすべてのテンプレート。
- **パーティションテーブル:** 選択されたすべてのテーブル。
- **ドメイン:** このロケーションが使用し、管理するドメイン。
- **環境: `production` (実稼働)** および 『コンテンツ管理ガイド』で作成された環境を含むすべての Puppet 環境。
- **組織:** ACME の組織。

上記のリソースについて、「プロパティーの編集」セクションでそれぞれのリソースタイプを確認します。

ロケーションの作成後に、**送信** をクリックします。

CLI を使用する場合

```
# hammer location create --name "New York" \
--description "Our example location"
```

これにより、ロケーションのサンプルが作成されます。

2.4. コンテキストの設定

Red Hat Satellite 6 でプロビジョニングを実行する前にコンテキストを設定する必要があります。コンテキストは新規システムのプロビジョニングに使用する組織およびロケーションを定義します。さらに、新規のインフラストラクチャーリソースがこのコンテキストに追加されます。

Web UI を使用する場合

コンテキスト メニューは、画面の左上隅にあります。コンテキストを選択しない場合、メニューには「すべてのコンテキスト」と示されます。このメニューにカーソルを置き、**組織** セレクターで **ACME** を選択します。これにより、コンテキストは ACME 組織に変更されます。次に、コンテキストメニューにカーソルを置き、**ロケーション** セレクターで **New York** を選択します。これにより、コンテキストはサンプルのロケーションに変更されます。

注記

各ユーザーはアカウント設定でデフォルトのコンテキストを設定できます。Web UI の右上のユーザー名に移動し、**マイアカウント** を選択してユーザーアカウントの設定を編集します。

CLI を使用する場合

CLI を使用している場合は、コマンドの末尾にオプションとして **--organization** または **--organization-id**、**--location** または **--location-id** のいずれかが含まれていることを確認します。以下は例になります。

```
# hammer host list --organization "ACME" --location "New York"
```

これにより、CLI を使って対話のコンテキストが設定されます。

2.5. 本章のまとめ

本章では、新しい組織とロケーションを作成し、それらをプロビジョニングのコンテキストとして設定する方法について説明しました。

次章では、Red Hat Satellite 6 プロビジョニングインフラストラクチャーを構成するリソースのいくつかについて説明します。

第3章 プロビジョニングリソースの設定

Red Hat Satellite 6 は、新規ホストの作成に役立つ一連のプロビジョニングリソースを提供します。このセクションでは、これらのリソースのいくつかや、それらのリソースがホストのプロビジョニングにどのように役立つかについて説明します。

サポートされるアーキテクチャー

PXE、Discovery およびブートディスクを使用したプロビジョニングについてサポートされるのは Intel x86_64 アーキテクチャーのみです。詳細は、Red Hat ナレッジベースソリューション Architectures Supported for Satellite 6 Provisioning を参照してください。

3.1. アーキテクチャーの作成

Satellite 内のアーキテクチャーはホストおよびオペレーティングシステムの論理グループを表します。アーキテクチャーは、ホストが Puppet に接続する際に Satellite によって自動的に作成されます。Satellite 6 には、ベーシックな i386 と x86_64 のアーキテクチャーがプリセットされています。

Web UI を使用する場合

1. **ホスト > アーキテクチャー** をクリックしてから、**アーキテクチャーの作成** をクリックします。
2. アーキテクチャーの **名前** を指定します。
3. このアーキテクチャーに含める **オペレーティングシステム** を選択します。利用可能なものがない場合は、作成して **ホスト > オペレーティングシステム** の下に割り当てます。
4. **送信** をクリックします。

CLI を使用する場合

hammer architecture create コマンドを使用して、新規アーキテクチャーを作成します。名前とアーキテクチャーに含めるオペレーティングシステムを指定します。

```
# hammer architecture create --name "architecture_name" \
--operatingsystems "os"
```

3.2. ハードウェアモデルの作成

ハードウェアモデルは、ホストが使用するハードウェアモデルを指定します。

Web UI を使用する場合

1. **ホスト > ハードウェアモデル** に移動します。
2. **モデルの作成** をクリックします。
3. ハードウェアモデルの **名前** を指定します。
4. オプションで、システムに **ハードウェアモデル** および **ベンダークラス** を入力できます。
5. **情報** フィールドに、ハードウェアモデルの詳細を入力します。
6. **送信** をクリックします。

CLI を使用する場合

hammer model create コマンドを使用して、新規のハードウェアモデルを作成します。必須となる唯一のパラメーターは、**--name** です。オプションで、**--hardware-model** パラメーターにハードウェアモデルを、**--vendor-class** パラメーターにベンダークラスを、**--info** パラメーターに詳細を入力します。

```
# hammer model create --name "model_name" --info "description" \
--hardware-model "hardware_model" --vendor-class "vendor_class"
```

3.3. インストールメディアの作成

インストールメディアは、Satellite Server がベースオペレーティングシステムをマシンにインストールするために使用するファイルのソースです。インストールメディアは、オペレーティングシステムのインストールツリーの形式で提供され、インストーラーをホストするマシンから HTTP URL 経由でアクセスできる必要があります。利用可能なインストールメディアは **ホスト > インストールメディア** メニューに表示されます。

ローカルにマウントされた ISO イメージなどの他のインストールメディアの場合、ユーザーは以下の手順を使用して独自のカスタムメディアパスを追加することができます。

Web UI を使用する場合

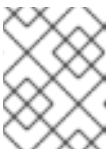
ホスト > インストールメディア に移動して **新規メディア** をクリックします。UI には、インストールメディアの詳細を入力できる一連のフィールドがあります。

- **名前:** ユーザーインターフェースのインストールメディアのエントリーを表す名前。
- **パス:** インストールツリーを含む URL または NFS シェア。複数の異なるシステムアーキテクチャーおよびバージョンを表すために以下の変数をパスで使用できます。
  - **$arch:** システムアーキテクチャー (例: x86_64)
  - **$version:** オペレーティングシステムのバージョン (例: 7.2)
  - **$major:** オペレーティングシステムのメジャーバージョン (例: 7)
  - **$minor:** オペレーティングシステムのマイナーバージョン (例: 2)
    HTTP パスの例:

    ```
    http://download.example.com/rhel/$version/Server/$arch/os/
    ```

    NFS パスの例:

    ```
    nfs://download.example.com:/rhel/$version/Server/$arch/os/
    ```

    注記

    Capsule Server の同期したコンテンツは HTTP パスを常に使用します。Capsule Server で管理されたコンテンツは NFS パスをサポートしません。
- **オペレーティングシステムの種類** メディアのディストリビューションまたはファミリー。たとえば、Red Hat Enterprise Linux、CentOS、および Fedora は、**Red Hat** ファミリーに属します。

Satellite Server はインストールメディアを現在のプロビジョニングコンテキストに追加します。追加のコンテキストは、**組織** および **ロケーション** タブで選択でき、今後のデバッグに役立ちます。

送信 をクリックしてインストールメディアを保存します。

CLI を使用する場合

`hammer medium create` コマンドを使用してインストールメディアを作成します。

```
# hammer medium create --name "CustomOS" --os-family "Redhat" \
--path 'http://download.example.com/rhel/$version/Server/$arch/os/' \
--organizations "ACME" --locations "New York"
```

3.4. パーティションテーブルの作成

パーティションテーブルは、Satellite Server が新規ホストで利用可能なディスクを設定する方法を定義する一連のディレクティブです。Red Hat Satellite 6 には、**`Kickstart default`** などの、デフォルトのパーティションテーブルのセットが含まれます。また、パーティションテーブルのエントリーを編集して、好みのパーティション設定スキームを設定したり、新規パーティションテーブルのエントリーを作成したりでき、そしてそのエントリーを Red Hat Enterprise Linux オペレーティングシステムのエントリーに追加することができます。

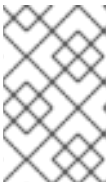
Web UI を使用する場合

ホスト > パーティションテーブル に移動し、**パーティションテーブルの作成** をクリックします。UI にはパーティションテーブルの詳細を入力するフィールドがあります。

- **名前:** パーティションテーブルを表す名前。
- **デフォルト:** テンプレートを新規の組織またはロケーションに自動的に関連付けられるように設定します。
- **スニペット:** テンプレートを他のパーティションテーブルレイアウトの再利用可能なスニペットに設定します。
- **オペレーティングシステムの種類** パーティションレイアウトのディストリビューションまたはファミリー。たとえば、Red Hat Enterprise Linux、CentOS、および Fedora は、**`Red Hat`** ファミリーに属します。
- **テンプレートエディター:** ディスクパーティションのレイアウトを入力するテキスト領域。以下は例になります。

  ```
  zerombr
  clearpart --all --initlabel
  autopart
  ```

  テンプレート ファイルブラウザーを使用してテンプレートファイルをアップロードすることもできます。



注記

レイアウトのフォーマットは、オペレーティングシステムのフォーマットと一致する必要があります。Red Hat Enterprise Linux 7.2 にはキックスタートファイルに一致するレイアウトが必要です。

- **監査コメント:** パーティションレイアウトへの変更の概要フィールド。

Satellite はパーティションテーブルを現在のプロビジョニングコンテキストに追加します。**組織** および **ロケーション** タブから追加のコンテキストを選択できます。

送信 をクリックしてパーティションテーブルを保存します。

CLI を使用する場合

CLI を使用してパーティションテーブルを作成する前に、パーティションレイアウトが含まれるテキスト形式のファイルを作成します。この例では **~/my-partition** ファイルを使用します。**hammer partition-table create** コマンドを使用してインストールメディアを作成します。

```
# hammer partition-table create --name "My Partition" --snippet false \
--os-family Redhat --file ~/my-partition --organizations "ACME" \
--locations "New York"
```

3.5. プロビジョニングテンプレートの作成

プロビジョニングテンプレートは、Satellite Server がホストにオペレーティングシステムをインストールする方法を定義します。プロビジョニングテンプレートには、以下を含む様々なタイプがあります。

- **provision:** プロビジョニングプロセスのテンプレート (例: キックスタートテンプレート)。キックスタートテンプレートの構文についての詳細は、**『Red Hat Enterprise Linux 7 インストールガイド』** の 「キックスタート構文の参考資料」 を参照してください。
- **PXELinux**、**PXEGrub**、**PXEGrub2** - TFTP サーバーにデプロイして、ホストが正しい kernel オプションでインストーラーを使用するための PXE ベースのテンプレート。
- **finish:** 主要なプロビジョニングプロセスの完了後の設定後スクリプト。これは SSH タスクとして実行されます。
- **Bootdisk:** PXE を使用しないブート方法のためのテンプレート。
- **kexec:** PXE を使用しないブート方法のためのカーネル実行テンプレート。
- **user_data:** **cloud-init** スクリプトなどのユーザーデータを受け入れるプロバイダー用の設定後スクリプト。
- **script:** デフォルトで使用されないが、カスタムタスクに役立つ任意のスクリプト。
- **ZTP:** Zero Touch Provisioning テンプレート。
- **POAP:** PowerOn Auto Provisioning テンプレート。
- **iPXE** - PXELinux の代わりに **iPXE** または **gPXE** 環境で使用するテンプレート。

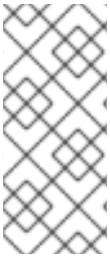
Red Hat Satellite には数多くのテンプレートのサンプルが含まれます。**ホスト > プロビジョニングテンプレート** に移動するとそれらを表示できます。それらのいずれかのクローンを作成したり、調整したり、独自のテンプレートを作成したりできます。テンプレートには Embedded Ruby (ERB) 構文を使用できます。詳細は、**『ホストの管理』** の 「テンプレート作成の参照」 を参照してください。

プロビジョニングテンプレートはダウンロードが可能です。ただし、ダウンロード前にデバッグ証明書を作成する必要があります。**『コンテンツ管理ガイド』** の 「組織のデバッグ証明書の作成」 を参照してください。

注記

テンプレートの変更履歴を表示するには、**ホスト > プロビジョニングテンプレート**に移動してテンプレートを選択し、**履歴** をクリックします。**戻す** をクリックすると、以前のバージョンでコンテンツを上書きできます。**差分の表示** をクリックすると、特定の変更についての情報が確認できます。

1. **テンプレート差分** タブでは、プロビジョニングテンプレートのボディの変更が表示されます。
2. **詳細** タブでは、テンプレートの説明の変更が表示されます。
3. **履歴** タブでは、テンプレートを変更したユーザーと変更日が表示されます。

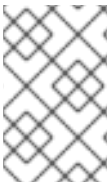


注記

フィニッシュテンプレートは、仮想環境におけるイメージベースのプロビジョニングにのみ使用する設計になっています。イメージと Foreman Discovery ISO を混同しないようにしてください。後者は Foreman Discovery イメージと呼ばれる場合もあります。このコンテキストにおけるイメージとは、デプロイメントを容易にするための仮想環境におけるインストールイメージです。

Web UI を使用する場合

ホスト > プロビジョニングテンプレートに移動し、**テンプレートの作成**をクリックします。UI にはプロビジョニングテンプレートの詳細を入力するフィールドがあります。



注記

ヘルプ タブでは、テンプレート構文についての情報が表示されます。テンプレート内の異なるタイプのオブジェクトで呼び出すことができる関数、変数、およびメソッドについて詳述されています。

別の方法では、テンプレート例からいずれかを選択し、**クローン** をクリックして複製してから、そのプリセットを変更することもできます。

- **テンプレート** タブ:
  - **名前:** プロビジョニングテンプレートのテキスト形式の名前。
  - **デフォルト:** テンプレートを新規の組織またはロケーションに自動的に関連付けられるように設定します。
  - **テンプレートエディター:** プロビジョニングテンプレートの本文を入力するテキスト領域。**テンプレート** ファイルブラウザーを使用してテンプレートファイルをアップロードすることもできます。
  - **監査コメント:** プロビジョニングテンプレートへの変更の概要フィールドです。
- **タイプ** タブ:
  - **スニペット:** プロビジョニングテンプレートをスニペットとして指定します。スニペットはスタンドアロンのプロビジョニングテンプレートではありませんが、他のプロビジョニングテンプレートに挿入できるプロビジョニングテンプレートの一部を構成します。
  - **タイプ: プロビジョニングテンプレート** などのテンプレートのタイプ。

- **関連付け** タブ:
  - **適用可能なオペレーティングシステム**セクションの **すべての項目** の一覧から、オペレーティングシステムエントリーの名前をクリックしてそのオペレーティングシステムエントリーを **選択された項目** の一覧に移動し、プロビジョニングテンプレートをそのオペレーティングシステムエントリーで使用できるようにします。
  - またオプションとして、**組み合わせの追加** をクリックして **ホストグループ**の一覧からホストグループを 1 つ選択するか、または **環境** の一覧から環境を 1 つ選択すると、指定したホストグループと環境の組み合わせにプロビジョニングテンプレートを使用できるようになります。

Satellite はプロビジョニングテンプレートを現在のプロビジョニングコンテキストに追加します。**組織** および **ロケーション** タブから追加のコンテキストを選択できます。

送信 をクリックしてプロビジョニングテンプレートを保存します。

CLI を使用する場合

CLI を使用してテンプレートを作成する前に、テンプレートが含まれるテキスト形式ファイルを作成します。この例では **~/my-template** ファイルを使用します。**hammer partition-table create** コマンドを使用してインストールメディアを作成し、**--type** オプションでタイプを指定します。

```
# hammer template create --name "My Provisioning Template" \
--file ~/my-template --type provision --organizations "ACME" \
--locations "New York"
```

3.5.1. プロビジョニング中の SSH キーのデプロイ

ユーザーに追加した SSH キーは、プロビジョニング中にデプロイできます。ユーザーに SSH キーを追加する方法については、**『RED HAT SATELLITE の管理』** の 「ユーザーへの SSH キーの追加」 を参照してください。

プロビジョニング中の SSH キーのデプロイ

1. **ホスト > プロビジョニングテンプレート** に移動します。
2. プロビジョニングテンプレートを作成、クローン、または既存テンプレートを編集します。詳細は 「プロビジョニングテンプレートの作成」を参照してください。
3. テンプレートで **テンプレート** タブをクリックします。
4. **テンプレートエディター**のフィールドで、**create_users** スニペットを **%post** セクションに追加します。

   ```
   <%= snippet('create_users') %>
   ```

5. **デフォルト** チェックボックスを選択します。
6. **関連付け** タブをクリックします。
7. **適用可能なオペレーティングシステム**リストから適切なオペレーティングシステムを選択します。
8. **送信** をクリックしてプロビジョニングテンプレートを保存します。

9. 新規ホストをプロビジョニングテンプレートに関連付けて作成するか、修正したテンプレートに関連付けた OS を使用しているホストを再ビルドします。詳細は、「Satellite Server でのホストの作成」 を参照してください。
Owned by ユーザーの SSH キーは、プロビジョニングプロセス中に`create_users` スニペットが実行されると、自動的に追加されます。**Owned by** は、個人のユーザーやユーザーグループに設定することができます。**Owned by** をユーザーグループに設定すると、そのユーザー内の全ユーザーの SSH キーが自動的に追加されます。

3.6. オペレーティングシステムの作成

オペレーティングシステムは、Satellite Server がホストにベースオペレーティングシステムをインストールする方法を定義するリソースの集合です。オペレーティングシステムのエントリーは、インストールメディアやパーティションテーブル、プロビジョニングテンプレートなどの事前に定義されたリソースを組み合わせます。

Red Hat の CDN からオペレーティングシステムをインストールすると、**ホスト > オペレーティングシステム** ページで新規エントリーが作成されます。ユーザーは以下の手順でカスタムオペレーティングシステムを追加することもできます。

Web UI を使用する場合

ホスト > オペレーティングシステムに移動し、**新規オペレーティングシステム**をクリックします。UI には、オペレーティングシステムの詳細を入力できる一連のフィールドがあります。

- **オペレーティングシステム** タブ:
  - **名前:** オペレーティングシステムエントリーを表すテキスト形式の名前。
  - **メジャーバージョン:** オペレーティングシステムのメジャーバージョンに対応する番号。
  - **マイナーバージョン:** オペレーティングシステムのマイナーバージョンに対応する番号。
  - **説明:** オペレーティングシステムの説明を入力するテキストフィールド。
  - **ファミリー:** 新規オペレーティングシステムの分類に使用するオペレーティングシステムのファミリー。
  - **root パスワードのハッシュ:** root パスワードのエンコーディング方法。
  - **アーキテクチャー:** オペレーティングシステムが使用するアーキテクチャーを選択します。**ホスト > アーキテクチャー** メニューで追加のアーキテクチャーを作成します。
- **パーティションテーブル** タブ:
  - このオペレーティングシステムに適用できる可能性のあるパーティションテーブルを選択します。
- **インストールメディア** タブ:
  - このオペレーティングシステムに適用するインストールメディアを選択します。詳細は「インストールメディアの作成」を参照してください。
- **テンプレート** タブ:
  - オペレーティングシステムで使用する **PXELinux テンプレート**、**プロビジョニングテンプレート**、および **フィニッシュテンプレート** を選択します。

- たとえば、プロビジョニングに iPXE を使用する場合は、**iPXE テンプレート** などの他のテンプレートを選択することもできます。

送信 をクリックしてプロビジョニングテンプレートを保存します。

CLI を使用する場合

hammer os create コマンドを使ってオペレーティングシステムを作成します。

```
# hammer os create --name "MyOS" \
--description "My custom operating system" \
--major 7 --minor 3 --family "Redhat" --architectures "x86_64" \
--partition-tables "My Partition" --media "Red Hat" \
--provisioning-templates "My Provisioning Template"
```

以下の点に注意してください。

- 先のセクションで作成したリソースを使用します(例: インストールメディア、パーティションテーブル、およびプロビジョニングテンプレート)。
- オペレーティングシステムにはプロビジョニングコンテキストがありません。オペレーティングシステムを構成するリソースのみがプロビジョニングコンテキストを持ちます。

3.7. コンピュートプロファイルの作成

コンピュートプロファイルは、仮想化インフラストラクチャーおよびクラウドプロバイダーなどのコンピュートリソースと併用されます。コンピュートプロファイルにより、ユーザーは CPU、メモリー、およびストレージなどのハードウェアを事前に定義できます。Red Hat Satellite 6 のデフォルトインストールには、以下の 3 つの事前に定義されたプロファイルが含まれます。

- **1-Small**
- **2-Medium**
- **3-Large**

ここでは、**4-Example** という 4 つ目のプロファイルを作成します。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動します。既存のプロファイルの一覧が表示されます。**新規のコンピュートプロファイル** をクリックします。

プロファイルの **名前** (例: **4-Example**) を入力し、**送信** をクリックします。

CLI を使用する場合

Red Hat Satellite 6.3 には、コンピュートプロファイルの CLI コマンドは実装されていません。

3.8. アクティベーションキーの作成

新規ホストを作成する前に、アクティベーションキーを作成することをお勧めします。このアクティベーションキーは、プロビジョニングのシナリオでシステムの登録時に使用されます。本書のシナリオでは、『コンテンツ管理ガイド』のサブスクリプションおよびリポジトリーを割り当てるためにアクティベーションキーのサンプルを作成します。

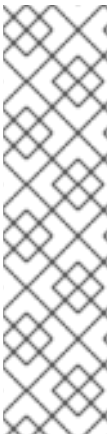
Web UI を使用する場合

コンテンツ > アクティベーションキー に移動し、**アクティベーションキーの作成** をクリックします。アクティベーションキーに以下の情報を入力します。

- **名前:** アクティベーションキーの名前。システムの登録プロセスでこの名前を使用します。**`example`** と入力します。
- **コンテンツホストの制限** このアクティベーションキーで Satellite Server が登録を許可するシステム数。**無制限のコンテンツホスト** を選択します。
- **説明:** アクティベーションキーのテキスト形式の説明。**`Example activation key`** と入力します。
- **環境:** 使用する環境。**`Production`** を選択します。
- **コンテンツビュー:** 環境内で使用するコンテンツビュー (およびリポジトリー) 。**`Base`** を選択します。

保存 をクリックすると、アクティベーションキーの詳細画面が表示されます。

ここで、登録時に割り当てる製品と有効にするリポジトリーを定義する必要があります。**サブスクリプション** タブに移動すると、空のサブスクリプション一覧が表示されます。**追加** をクリックして Red Hat Enterprise Linux サブスクリプションを選択し、**選択を追加** をクリックします。



注記

デフォルトでは、**Auto-Attach** オプションが有効になっています。アクティベーションキーで自動アタッチを有効にし、キーに割り当てたサブスクリプションがある場合は、サブスクリプション管理サービスが、基準に基づいて、最適な関連サブスクリプションを選択して割り当てます。自動アタッチを有効にして、キーにサブスクリプションを関連付けないようにしておくこともできます。このようなキーは、RHEL サブスクリプションを使用し、ハイパーバイザーから RHEL Virtual Data Center (VDC) サブスクリプションを継承する仮想マシンが必要ない場合に、仮想マシンを登録するために一般的に使用されます。自動アタッチを無効にすると、サブスクリプションマネジメントサービスは、ホストの登録中に関連するすべてのサブスクリプションをアタッチしようとします。1 つでもサブスクリプションがアタッチできないと、ホストの登録は失敗します。

製品コンテンツ ページに移動します。アクティベーションキーの製品に関連付けられているリポジトリーがすべて表示されます。デフォルトでは、Satellite Server が有効にしているのは以下のみです。

- システム要件に最も適したリポジトリー。このケースでは Red Hat Enterprise Linux 7 Server RPMs のみになります。
- カスタムコンテンツ

シナリオには以下のデフォルトが設定されているはずです。

Red Hat Enterprise Linux Server:

- Red Hat Enterprise Linux 7 Server (Kickstart) - Enabled: **`No (Default)`**
- Red Hat Satellite Tools 6.3 (for RHEL 7 Server) (RPMs) - Enabled: **`No (Default)`**
- Red Hat Enterprise Linux 7 Server (RPMs) - Enabled: **`Yes (Default)`**

Red Hat Satellite Tools 6.3 リポジトリーには設定ツール (**katello-agent** および **puppet** など) が含まれるため、これを有効にします。以下のように変更します。

- Red Hat Satellite Tools 6.3 (for RHEL 7 Server) (RPMs) - Enabled: **Override to Yes**

保存 をクリックします。

CLI を使用する場合

アクティベーションキーを作成します。

```
# hammer activation-key create --name "example" \
--unlimited-content-hosts true --description "Example activation key" \
--lifecycle-environment "Production" --content-view "Base"  \
--organization "ACME"
```

サブスクリプション ID 一覧を取得します。

```
# hammer subscription list --organization "ACME"
```

Red Hat Enterprise Linux サブスクリプション UUID をアクティベーションキーに割り当てます。

```
# hammer activation-key add-subscription --name "example" \
--subscription-id ff808181533518d50152354246e901aa \
--organization "ACME"
```

アクティベーションキーに関連付けられている製品コンテンツを一覧表示します。

```
# hammer activation-key product-content --name "example" \
--organization "ACME"
```

Red Hat Satellite Tools 6.3 リポジトリーのデフォルトの自動有効化ステータスを上書きします。デフォルトでは無効にされています。以下のコマンドでこれを有効にします。

```
# hammer activation-key content-override --name "example" \
--content-label rhel-7-server-satellite-tools-6.3-rpms \
--value 1 --organization "ACME"
```

アクティベーションキーのサンプルをプロビジョニングされたシステムの登録に使用できます。

3.9. ホストへのデフォルト暗号化 ROOT パスワードの設定

プロビジョニングしたホストにプレーンテキストのデフォルト **root** パスワードを設定したくない場合は、デフォルトの暗号化パスワードを使用することができます。

ホストにデフォルトの暗号化パスワードを設定するには、以下の手順に従います。

1. 暗号化パスワードを生成します。以下のコマンドを使用します。

```
python -c 'import crypt,getpass;pw=getpass.getpass();
print(crypt.crypt(pw)) if (pw==getpass.getpass("Confirm: ")) else
exit()'
```

2. 後で使用するために、パスワードをコピーしておきます。

3. Satellite Web UI で、**管理 > 設定** に移動します。
4. **設定** ページで、**プロビジョニング** タブを選択します。
5. **Name** コラムで **Root パスワード** を探して、**クリックして編集** をクリックします。
6. 生成した暗号化パスワードを貼り付け、**保存** をクリックします。

3.10. 本章のまとめ

本章では、新規ホストをプロビジョニングするためのリソースについて説明しました。これには、インストールメディア、パーティションテーブル、プロビジョニングテンプレート、コンピュートプロファイル、およびアクティベーションキーが含まれます。本書の後のシナリオでは、これらのリソースをホストプロビジョニングプロセスに適用する方法を説明します。

次の章では、プロビジョニング用にネットワークインフラストラクチャーを設定する方法について説明します。

第4章 ネットワークの設定

それぞれのプロビジョニングタイプには、なんらかのネットワーク設定が必要です。新規ホストが Capsule Server にアクセスできるよう確認します。Capsule Server は、Satellite Server の統合 Capsule または外部 Capsule Server のいずれかになります。必要なホストが分離したネットワーク上にあり、Satellite Server に直接接続できない場合、またはコンテンツが Capsule Server と同期している場合、外部 Capsule Server からのホストのプロビジョニングを選択することができます。また、外部 Capsule Server を使用したプロビジョニングは、ネットワークの帯域幅を節約できます。

Capsule Server の設定には、基本的な要件が 2 つあります。

- ネットワークサービスの設定には、以下が含まれます。
  - コンテンツ配信サービス
  - ネットワークサービス (DHCP、DNS、および TFTP)
  - Puppet 設定
- Satellite Server でネットワークリソースデータを定義し、新規ホストでのネットワークインターフェースの設定をサポートします。

本章では、Satellite Server の統合 Capsule でネットワークサービスを設定する方法を重点的に説明します。しかし、これらの方法は、特定のネットワークを管理するスタンドアロン Capsule Server の設定にも同様に当てはまります。Satellite を設定して外部の DHCP、DNS、および TFTP サービスを使用するには、**『Red Hat Satellite インストールガイド』** の 「外部サービスの設定」 を参照してください。

この例では、ACME にはホストをプロビジョニングするためのプライベートネットワークがあります。このプライベートネットワークの詳細は以下のようになります。

サブネット	192.168.140.0/24	
外部ゲートウェイ	192.168.140.1	
Satellite Server	192.168.140.2	
Discovered ホストおよび管理対象外のホストの DHCP 割り当てプール	192.168.140.10 - 192.168.140.110	
ホストプロビジョニングの DHCP 割り当てプール	192.168.140.111 - 192.168.140.250	

Satellite Server では、Discovered システムおよびプロビジョニングシステムの両方に同じ DHCP 範囲を定義することはできますが、それぞれのサービスに同じサブネット内の別々の範囲を使用することが推奨されます。

4.1. イメージベースのプロビジョニングの検討

ブート後の設定方法

finish ブート後設定スクリプトを使用するイメージは、Satellite の 統合 Capsule または外部 Capsule

などの管理された DHCP サーバーが必要です。ホストは DHCP Capsule と関連付けされたサブネットで作成される必要があり、ホストの IP アドレスは、DHCP 範囲の有効な IP アドレスでなければなりません。外部の DHCP サービスを使用することは可能ですが、IP アドレスは手動で入力する必要があります。イメージの設定に対応する SSH 認証情報は、ブート後の設定を実行できるように Satellite に設定しなければなりません。

設定後スクリプトに依存するイメージからブートした仮想マシンのトラブルシューティングの際、以下の項目を確認する必要があります。

- ホストには、Satellite Server に割り当てられたサブネットがあります。
- サブネットには、Satellite Server に割り当てられた DHCP Capsule があります。
- ホストには、Satellite Server に割り当てられた有効な IP アドレスがあります。
- DHCP を使用した仮想マシンが取得した IP アドレスは、Satellite Server に設定されたアドレスと一致します。
- イメージから作成された仮想マシンは、SSH リクエストに応答します。
- イメージから作成された仮想マシンは、SSH を介してユーザーとパスワードを承認します。ユーザーとパスワードは、デプロイされたイメージと関連付けされます。

ブート前の初期化の設定方法

cloud-init スクリプトを使用するイメージは通常、イメージに IP アドレスを含むことを回避するため、DHCP サーバーを必要とします。管理された DHCP Capsule が推奨されます。イメージは、システムがブートされた時に開始し、設定完了時に使用するスクリプトまたは設定データを取得するための **cloud-init** サービスを設定する必要があります。

イメージに含まれる初期スクリプトに依存するイメージからブートした仮想マシンのトラブルシューティングの際、以下の項目を確認する必要があります。

- サブネット上に DHCP サーバーがあります。
- 仮想マシンには、**cloud-init** サービスがインストールされ、有効化されています。

仮想マシンイメージの **finish** および **cloud-init** スクリプトに対する異なるレベルのサポートに関する詳細は、Red Hat カスタマーポータルの Red Hat ナレッジベースソリューション What are the supported compute resources for the finish and cloud-init scripts を参照してください。

4.2. ネットワークサービスの設定

一部のプロビジョニング方法では、Capsule Server サービスを各種の目的で使用します。たとえば、ネットワークが Capsule Server に対して DHCP サーバーとして機能するよう要求する場合があります。また、 オペレーティングシステムを新規ホストにインストールする手段として PXE ブートサービスを必要とするネットワークもあります。この場合には、主な PXE ブートサービスである DHCP、DNS および TFTP を使用するよう Capsule Server を設定する必要があります。これを実現するには、Satellite Server にこれらのサービスを設定するオプションと共に **`satellite-installer`** スクリプトを実行します。外部 Capsule Server にこれらのサービスを設定するには、**`satellite-installer --scenario capsule`** を実行します。

この例では、ACME は Satellite Server の統合 Capsule をプロビジョニングネットワークに接続し、PXE ブートサービスを提供します。Satellite Server は以下の NIC 設定を使用します。

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
```

```
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 52:54:00:33:e3:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.125.35/24 brd 192.168.125.255 scope global dynamic ens3
       valid_lft 3042sec preferred_lft 3042sec
    inet6 fe80::5054:ff:fe33:e31c/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
   link/ether 52:54:00:fd:24:ae brd ff:ff:ff:ff:ff:ff
   inet 192.168.140.2/24 brd 192.168.140.255 scope global ens8
      valid_lft forever preferred_lft forever
   inet6 fe80::5054:ff:fefd:24ae/64 scope link
      valid_lft forever preferred_lft forever
```

Satellite Server は、Red Hat の CDN への接続など、外部との通信に **eth0** を使用します。ACME は **eth1** インターフェースを使用して、**192.168.140.0/24** サブネットを使用するホスト向けにプライベートプロビジョニングネットワークに接続します。ここでの目的は、Satellite Server の統合 Capsule が、このネットワークの新規ホストに対して DHCP、DNS、および TFTP サーバーとして機能することです。

注記

Satellite Server の統合 Capsule は、上記のサービスを提供します。これらのサービスは、他のネットワークの追加の Satellite Capsule にも設定することができます。

satellite-installer スクリプトは、以下のオプションを使用してネットワークサービスを設定します。

DHCP オプション

--foreman-proxy-dhcp

DHCP サービスを有効にします。このオプションを **true** に設定します。

--foreman-proxy-dhcp-managed

DHCP サービスを管理するため Foreman を有効にします。このオプションを **true** に設定します。

--foreman-proxy-dhcp-gateway

この例では、DHCP プールゲートウェイを **192.168.140.1** に設定します。これは、ACME のプライベートネットワーク上のホスト向け外部ゲートウェイのアドレスです。

--foreman-proxy-dhcp-interface

要求をリッスンするために DHCP サービスのインターフェースを設定します。この例では、**eth1** に設定します。

--foreman-proxy-dhcp-nameservers

DHCP を介してクライアントに提供されたネームサーバーのアドレスを設定します。この例では、**eth1** の Satellite Server のアドレス **192.168.140.1** に設定します。

--foreman-proxy-dhcp-range

Discovered サービスおよび管理対象外サービス用のスペースで区切られた DHCP プールの範囲で

す。この例では、**192.168.140.10 192.168.140.110** に設定します。これにより、100 個のアドレスを含むプールが提供されます。

--foreman-proxy-dhcp-server

管理する DHCP サーバーのアドレスを設定します。この例では、**192.168.140.2** になります。

DNS オプション

--foreman-proxy-dns

DNS サービスを有効にします。このオプションを **true** に設定します。

--foreman-proxy-dns-forwarders

DNS フォワーダーを設定します。この例では、これを **8.8.8.8; 4.4.4.4** に設定し、2 つのパブリック DNS サーバーを使用します。個人用としては独自の DNS サーバーを代わりに使用してください。

--foreman-proxy-dns-interface

DNS 要求をリッスンするためのインターフェースを設定します。この場合、**eth1** に設定します。

--foreman-proxy-dns-reverse

DNS 逆引きゾーン名です。この例では **140.168.192.in-addr.arpa** を使用します。

--foreman-proxy-dns-server

管理する DNS サーバーのアドレスを設定します。この例では **192.168.140.2** になります。

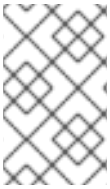
--foreman-proxy-dns-zone

DNS ゾーン名を設定します。この例では **example.com** を使用します。

TFTP オプション

--foreman-proxy-tftp

TFTP サービスを有効にします。このオプションを **true** に設定します。



注記

satellite-installer --scenario capsule --help を実行し、DHCP、DNS、TFTP およびその他の Satellite Capsule サービスに関するさらなるオプションを表示します。

ネットワークサービスの設定

1. **satellite-installer** コマンドを入力し、必要なネットワークサービスを設定します。

```
# satellite-installer --foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed true \
--foreman-proxy-dhcp-gateway "192.168.140.1" \
--foreman-proxy-dhcp-interface "eth1" \
--foreman-proxy-dhcp-nameservers "192.168.140.2" \
--foreman-proxy-dhcp-range "192.168.140.10 192.168.140.110" \
--foreman-proxy-dhcp-server "192.168.140.2" \
--foreman-proxy-dns true \
--foreman-proxy-dns-forwarders "8.8.8.8; 4.4.4.4" \
--foreman-proxy-dns-interface "eth1" \
--foreman-proxy-dns-reverse "140.168.192.in-addr.arpa" \
```

```
--foreman-proxy-dns-server "192.168.140.2" \
--foreman-proxy-dns-zone "example.com" \
--foreman-proxy-tftp true
```

この例では、Satellite Server の統合 Capsule Server を設定します。外部 Capsule Server でこのコマンドを実行する場合は、**satellite-installer --scenario capsule** を使用します。

2. Satellite Server の CLI で、Capsule Server の機能をリフレッシュして変更を表示します。

   a. 設定する Capsule Server を決定します。

   ```
   # hammer proxy list
   ```

   b. Capsule Server の機能をリフレッシュします。

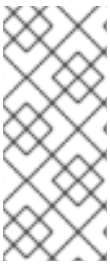
   ```
   # hammer proxy refresh-features --name "satellite.example.com"
   ```

3. Capsule Server に設定されたサービスを確認します。

```
# hammer proxy info --name "satellite.example.com"
```

4.3. SATELLITE SERVER へのドメインの追加

Satellite Server はネットワーク上の各ホストのドメイン名を定義します。これは、Satellite Server がドメインとドメイン名を割り当てる Capsule Server を認識する必要があることを意味します。この例では、ACME の内部ネットワーク用に **example.com** ドメインを作成します。



注記

Satellite Server には、Satellite Server のインストールの一環として関連するドメインがすでに作成されている可能性があります。コンテキストを **任意の組織** および **任意のロケーション** に切り替えてから、ドメインの一覧でこれが存在するかどうかを確認します。存在する場合は、このドメインエントリーを修正して DNS Capsule を定義し、組織を設定してからロケーションを設定します。

Web UI を使用する場合

インフラストラクチャー > ドメイン に移動して、**新規ドメイン** をクリックします。この UI には、ドメインの詳細を入力できるフィールドがあります。

- **ドメイン** タブ:
  - **DNS Domain** - ドメイン名。**example.com** はこの例になります。
  - **Description (フルネーム)**: ドメインのテキスト形式の説明。この例では**ACME's example domain** を使用しています。
  - **DNS Capsule** - DNS の割り当てに使用するカプセル。この例では、Satellite Server の統合 Capsule を使用します。
- **ロケーション** タブ:

- このドメインを使用する場所を選択します。たとえば、**New York** などのロケーションを選択します。

- **組織** タブ:
  - **ACME** など、このドメインを使用する組織を選択します。

CLI を使用する場合

以下のコマンドでドメインを作成します。

```
# hammer domain create --name "example.com" \
--description "ACME's example domain" --dns_id 1 \
--locations "New York" --organizations "ACME"
```

注記

この例では **--dns-id** オプションは 1 を使用しています。1 は、Satellite Server の統合 Capsule の ID です。

4.4. SATELLITE SERVER へのサブネットの追加

Satellite Server は新規ホストのインターフェースを設定します。そのため、Satellite Server はこれらのインターフェースを接続するネットワークを認識する必要があります。これは、各サブネットの情報を Satellite Server に追加する必要があることを意味します。これにはゲートウェイ、DHCP、および DNS などの情報が含まれます。この例では、Satellite Server の統合 Capsule が管理する **'192.168.140.0/24'** ネットワークのサブネットのマッピングを作成します。

Web UI を使用する場合

インフラストラクチャー > サブネット に移動して、**新規サブネット** をクリックします。UI には、サブネットの詳細を入力できる一連のフィールドがあります。

- **サブネット** タブ:
  - **名前:** サブネットのテキスト形式の名前。例:**ACME's Internal Network**
  - **ネットワークアドレス:** サブネットのネットワークアドレス。例:**192.168.140.0**
  - **ネットワークプレフィックス** - サブネット用のネットワークプレフィックスです。例:**24**
  - **ネットワークマスク:** サブネットのネットワークマスク。例:**255.255.255.0**
  - **ゲートウェイアドレス:** サブネットの外部ゲートウェイ。例:**192.168.140.1**
  - **プライマリー DNS サーバー:** サブネットのプライマリー DNS。例:**192.168.140.2**
  - **セカンダリー DNS サーバー:** サブネットのセカンダリー DNS。例:**8.8.8.8**
  - **IPAM:** IP アドレス管理 (IPAM) に使用するメソッド:
    - **DHCP:** サブネットには DHCP サーバーが含まれます。
    - **内部 DB** - サブネットには DHCP サーバーは含まれませんが、Satellite で IP アドレスの割り当てを管理し、内部データベースに IP アドレスを記録することが想定されます。

        - **なし:** IP アドレス管理がありません。
          この例では、Satellite Server は DHCP サーバーとして機能するため、 **DHCP** を使用します。

    - **開始アドレス (Start of IP range):** プロビジョニングサービスの IP 割り当て範囲の開始を定義します。例: **192.168.140.111**

    - **終了アドレス (End of IP range):** プロビジョニングサービスの IP 割り当ての終了を定義します。例: **192.168.140.250**

    - **VLAN ID** - ブロードキャストを分離するためのサブネットの VLAN ID 番号。この例では VLAN を使用しないため、このフィールドを空白にします。

    - **ブートモード** - このネットワークのネットワークインターフェースのデフォルトブートモードを定義します。

        - **静的** ブートモードとは、このサブネットに割り当てられたネットワークインターフェースは、IP アドレスおよびネットワークマスクを設定ファイルに直接設定することを意味し、DHCP を使用してこれらを取得することを回避します。ゲートウェイおよび DNS サーバーは、DHCP から取得できないことに注意してください。したがって、これらを設定したい場合は、**ゲートウェイアドレス** および **プライマリー DNS サーバー** のフィールドに、正しい値を入力します。これらを省略できるのは、ネットワーク外にトラフィックをルーティングせず (インストールメディアはローカル)、DNS 解決なしに IP アドレスを直接使用する場合だけです。

        - **DHCP** ブートモードとは、このサブネットに割り当てられたネットワークインターフェースは、DHCP に設定されることを意味します。

- **リモート実行** タブ:

    - リモート実行を制御する Capsule を選択します。この例では Satellite Server 自体がこれに該当します。

- **ドメイン** タブ:

    - このサブネットに適用されるドメインを選択します。

- **Capsule** タブ:

    - DHCP、TFTP、および逆引き DNS サービスを含む、サブネットの各サービスに適用される Capsule を選択します。この例では、各サービスに Satellite Server の統合 Capsule を使用します。

- **パラメーター** タブでは、任意のサブネットレベルのパラメーターを設定し、このサブネットに割り当てられたホストに適用します。たとえば、テンプレートで使用できるユーザー定義ブール値またはストリングパラメーターなどです。

- **ロケーション** タブ:

    - この capsule を使用するロケーションを選択します。たとえば、**New York** というロケーションを選択します。

- **組織** タブ:

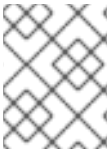
    - この capsule を使用する組織を選択します。たとえば、**ACME** を選択します。

送信 をクリックしてサブネットの情報を保存します。

CLI を使用する場合

以下のコマンドでサブネットを作成します。

```
# hammer subnet create --name "ACME's Internal Network" \
--network "192.168.140.0" --mask "255.255.255.0" \
--gateway "192.168.140.1" --dns-primary "192.168.140.2" \
--dns-secondary "8.8.8.8" --ipam "DHCP" \
--from "192.168.140.111" --to "192.168.140.250" --boot-mode "DHCP" \
--domains "example.com" --dhcp-id 1 --dns-id 1 --tftp-id 1 \
--locations "New York" --organizations "ACME"
```



注記

この例では、**--dhcp-id**、**--dns-id**、および **--tftp-id** オプションは 1 を使用します。1 は Satellite Server の統合 Capsule の ID です。

4.5. プロビジョニング時間を削減するための IPXE の設定

Red Hat Satellite 6.3 では、PXELinux を設定して iPXE をチェーンブートし、HTTP プロトコルを使用してブートすることができます。これは、高レイテンシーのネットワークにおいて TFTP よりも迅速で、信頼性も高まります。

Red Hat Satellite 6.3 で iPXE を使用する方法は 3 つあります。

1. iPXE をプライマリーファームウェアとして使用するハイパーバイザーを使った仮想マシンのチェーンブート
2. TFTP を介して PXELinux を使用し、iPXE を直接ベアメタルホストにチェーンロードします。
3. UNDI を介して PXELinux を使用します。UNDI は、HTTP を使用してカーネルとベアメタルホストの初期 RAM ディスクを移動します。

前提条件

開始する前に、以下の条件を満たしていることを確認してください。

- 使用する Red Hat Satellite にホストが存在します。
- プロビジョニングインターフェースの MAC アドレスが、ホスト設定と一致します。
- ホストのプロビジョニングインターフェースには、有効な DHCP 予約があります。
- NIC は、PXE ブートが可能です。詳細は http://ipxe.org/appnote/hardware_drivers を参照してください。
- NIC は、iPXE と互換性があります。

4.5.1. 仮想マシンのチェーンブート

仮想化ハイパーバイザーの多くは、PXE ブートのプライマリーファームウェアとして iPXE を使用します。このため、TFTP および PXELinux なしでチェーンブートが可能です。

仮想マシンワークフローのチェーンブート

仮想化ハイパーバイザーを使用することで、 TFTP および PXELinux の必要性がなくなります。仮想化ハイパーバイザーのワークフローは、以下の通りです。

1. 仮想マシンが起動します。
2. iPXE が DHCP を使用してネットワークの認証情報を取得します。
3. iPXE が DHCP を使用して HTTP アドレスを取得します。
4. iPXE が Red Hat Satellite から iPXE テンプレートをチェーンロードします。
5. iPXE が、インストーラーのカーネルおよび初期 RAM ディスクをロードします。

使用したいハイパーバイザーが iPXE をサポートしていることを確認します。以下の仮想化ハイパーバイザーは、iPXE をサポートします。

- libvirt
- oVirt
- RHEV

iPXE を使うための Red Hat Satellite Server の設定

デフォルトのテンプレートを使用して、ホストの iPXE ブーティングを設定できます。テンプレートのデフォルトの値を変更したい場合は、テンプレートをクローンし、そのクローンを編集します。

1. Satellite Web UI で、**ホスト > プロビジョニングテンプレート** に移動し、**`Kickstart default iPXE`** を入力後、**検索** をクリックします。
2. オプション: テンプレートを変更したい場合は、**クローン** をクリックして独自の名前を入力し、**送信** をクリックします。
3. 使用したいテンプレートの名前をクリックします。
4. テンプレートをクローンした場合、変更が必要な時は **テンプレート** タブで実施できます。
5. **関連付け** タブをクリックし、ホストが使用するオペレーティングシステムを選択します。
6. **ロケーション** タブをクリックして、ホストの所在を追加します。
7. **組織** タブをクリックして、ホストが属する組織を追加します。
8. **送信** をクリックして変更を保存します。
9. **ホスト > オペレーティングシステム** に移動し、ホストのオペレーティングシステムを選択します。
10. **テンプレート** タブをクリックします。
11. **iPXE テンプレート** リストから、使用したいテンプレートを選択します。
12. **送信** をクリックして変更を保存します。
13. **ホスト > すべてのホスト** に移動します。
14. **ホスト** のページで、使用したいホストを選択します。

15. **テンプレート** タブを選択します。
16. **iPXE テンプレート** リストから、**レビュー** を選択し、**Kickstart default iPXE** テンプレートが正しいテンプレートであることを確認します。
17. iPXE ファームウェアのチェーンブートの無限ループを回避するには、**/etc/dhcp/dhcpd.conf** ファイルを編集し、以下の例に一致させます。分離したネットワークを使用する場合は、Satellite Server の URL ではなく、TCP port 8000 を伴う Capsule Server の URL を使用します。
    a. **/etc/dhcp/dhcpd.conf** ファイルの Bootfile Handoff セクションで、以下の行を見つけます。

```
} else {
  filename "pxelinux.0";
}
```

    b. else ステートメントの前に、以下に示す別の **elsif** ステートメントを追加します。

```
elsif exists user-class and option user-class = "iPXE" {
  filename "http://satellite.example.com/unattended/iPXE";
}
```

    c. **if** セクションが、以下の例と一致することを確認します。

```
if option architecture = 00:06 {
  filename "grub2/shim.efi";
} elsif option architecture = 00:07 {
  filename "grub2/shim.efi";
} elsif option architecture = 00:09 {
  filename "grub2/shim.efi";
} elsif exists user-class and option user-class = "iPXE" {
  filename "http://satellite.example.com/unattended/iPXE";
} else {
  filename "pxelinux.0";
}
```

注記

http://satellite.example.com/unattended/iPXE では、Red Hat Satellite Capsule **http://capsule.example.com:8000/unattended/iPXE** を使用することもできます。アップグレードごとに **/etc/dhcp/dhcpd.conf** ファイルを更新する必要があります。**/etc/dhcp/dhcpd.conf** ファイルのコンテンツは大文字と小文字を区別します。

4.5.2. iPXE ディレクトリーのチェーンブート

この手順を使用すると、ネットワーク通信のビルトインドライバーを使って iPXE を直接チェーンブートできます。Red Hat Satellite Capsule および Server を設定して iPXE を使用するには、別の手順があります。

この手順を使用できるのは、ベアメタルホストのみです。

直接または UNDI ワークフローと共に iPXE をチェーンブート

1. ホストの電源をオンにします。
2. PXE ドライバーは、DHCP を使用してネットワークの認証情報を取得します。
3. PXE ドライバーは、TFTP を使用して PXELinux ファームウェア **pxelinux.0** を取得します。
4. PXELinux は、TFTP サーバーの設定ファイルを検索します。
5. PXELinux は、iPXE **ipxe.lkrn** または **undionly-ipxe.0** をチェーンロードします。
6. iPXE は、再び DHCP を使用してネットワークの認証情報を取得します。
7. iPXE は、DHCP を使用して HTTP アドレスを取得します。
8. iPXE が Red Hat Satellite から iPXE テンプレートをチェーンロードします。
9. iPXE が、インストーラーのカーネルおよび初期 RAM ディスクをロードします。

iPXE を使用するための Red Hat Satellite Capsule の設定

この手順を使用して、iPXE を使用するために Capsule を設定することができます。

この手順は、すべての Capsule で実行する必要があります。

Capsule を設定して iPXE をチェーンブートするには、以下を実行します。

1. **ipxe-bootimgs** RPM パッケージをインストールします。

```
# yum install ipxe-bootimgs
```

2. iPXE ファームウェアを TFTP サーバーのルートディレクトリーにコピーします。

```
# cp /usr/share/ipxe/ipxe.lkrn /var/lib/tftpboot/
```

TFTP は **chroot** 環境で実行するので、シンボリックリンクは使用しません。

3. ファイルのコンテンツを修正します。

```
# restorecon -RvF /var/lib/tftpboot/
```

iPXE を使うための Red Hat Satellite Server の設定

デフォルトのテンプレートを使用して、ホストの iPXE ブーティングを設定できます。テンプレートのデフォルトの値を変更したい場合は、テンプレートをクローンし、そのクローンを編集します。

1. Satellite Web UI で、**ホスト > プロビジョニングテンプレート** に移動し、**PXELinux chain iPXE** を入力後、**検索** をクリックします。
2. オプション: テンプレートを変更したい場合は、**クローン** をクリックして独自の名前を入力し、**送信** をクリックします。
3. 使用したいテンプレートの名前をクリックします。
4. テンプレートをクローンした場合、変更が必要な時は **テンプレート** タブで実施できます。

5. **関連付け** タブをクリックし、ホストが使用するオペレーティングシステムを選択します。
6. **ロケーション** タブをクリックして、ホストの所在を追加します。
7. **組織** タブをクリックして、ホストが属する組織を追加します。
8. **送信** をクリックして変更を保存します。
9. **プロビジョニングテンプレート** ページの検索フィールドに **`Kickstart default iPXE`** を入力し、**検索** をクリックします。
10. オプション: テンプレートを変更したい場合は、**クローン** をクリックして独自の名前を入力し、**送信** をクリックします。
11. 使用したいテンプレートの名前をクリックします。
12. テンプレートをクローンした場合、変更が必要な時は **テンプレート** タブで実施できます。
13. **関連付け** タブをクリックし、テンプレートをホストが使用するオペレーティングシステムに関連付けします。
14. **ロケーション** タブをクリックして、ホストの所在を追加します。
15. **組織** タブをクリックして、ホストが属する組織を追加します。
16. **送信** をクリックして変更を保存します。
17. **ホスト > オペレーティングシステム** に移動し、ホストのオペレーティングシステムを選択します。
18. **テンプレート** タブをクリックします。
19. **PXELinux テンプレート** リストから、使用したいテンプレートを選択します。
20. **iPXE テンプレート** リストから、使用したいテンプレートを選択します。
21. **送信** をクリックして変更を保存します。
22. **ホスト > すべてのホスト** に移動し、使用したいホストを選択します。
23. **テンプレート** タブを選択し、**PXELinux テンプレート** リストから **レビュー** を選択して、そのテンプレートが正しいテンプレートであることを確認します。
24. **iPXE テンプレート** リストから **レビュー** を選択し、そのテンプレートが正しいテンプレートであることを確認します。

注記

PXELinux のエントリーがない場合、または新しいテンプレートが見つからない場合は、**ホスト > すべてのホスト** に移動し、ホスト上で **編集** をクリックします。**オペレーティングシステム** タブをクリックし、続いてプロビジョニングテンプレート **解決** ボタンをクリックして、テンプレートのリストをリフレッシュします。

4.5.3. UNDI を使用した iPXE のチェーンブート

この手順を使用すると、UNDI を使って iPXE をチェーンブートできます。Red Hat Satellite Capsule および Server を設定して iPXE を使用するには、別の手順があります。

この手順を使用できるのは、ベアメタルホストのみです。

直接または UNDI ワークフローと共に iPXE をチェーンブート

1. ホストの電源をオンにします。
2. PXE ドライバーは、DHCP を使用してネットワークの認証情報を取得します。
3. PXE ドライバーは、TFTP を使用して PXELinux ファームウェア **pxelinux.0** を取得します。
4. PXELinux は、TFTP サーバーの設定ファイルを検索します。
5. PXELinux は、iPXE **ipxe.lkrn** または **undionly-ipxe.0** をチェーンロードします。
6. iPXE は、再び DHCP を使用してネットワークの認証情報を取得します。
7. iPXE は、DHCP を使用して HTTP アドレスを取得します。
8. iPXE が Red Hat Satellite から iPXE テンプレートをチェーンロードします。
9. iPXE が、インストーラーのカーネルおよび初期 RAM ディスクをロードします。

iPXE を使用するための Red Hat Satellite Capsule の設定

この手順を使用して、iPXE を使用するために Capsule を設定することができます。

この手順は、すべての Capsule で実行する必要があります。

Capsule を設定して iPXE をチェーンブートするには、以下を実行します。

1. **ipxe-bootimgs** RPM パッケージをインストールします。

```
# yum install ipxe-bootimgs
```

2. iPXE ファームウェアを TFTP サーバーのルートディレクトリーにコピーし、ファイル名を変更します。

```
# cp /usr/share/ipxe/undionly.kpxe /var/lib/tftpboot/undionly-ipxe.0
```

TFTP は **chroot** 環境で実行するので、シンボリックリンクは使用しません。

3. ファイルのコンテンツを修正します。

```
# restorecon -RvF /var/lib/tftpboot/
```

iPXE を使うための Red Hat Satellite Server の設定

デフォルトのテンプレートを使用して、ホストの iPXE ブーティングを設定できます。テンプレートのデフォルトの値を変更したい場合は、テンプレートをクローンし、そのクローンを編集します。

1. Satellite Web UI で、**ホスト > プロビジョニングテンプレート** に移動し、**PXELinux chain iPXE UNDI** を入力後、**検索** をクリックします。

2. オプション: テンプレートを変更したい場合は、**クローン** をクリックして独自の名前を入力し、**送信** をクリックします。
3. 使用したいテンプレートの名前をクリックします。
4. テンプレートをクローンした場合、変更が必要な時は **テンプレート** タブで実施できます。
5. **関連付け** タブをクリックし、ホストが使用するオペレーティングシステムを選択します。
6. **ロケーション** タブをクリックして、ホストの所在を追加します。
7. **組織** タブをクリックして、ホストが属する組織を追加します。
8. **送信** をクリックして変更を保存します。
9. **プロビジョニングテンプレート** ページの検索フィールドに **`Kickstart default iPXE`** を入力し、**検索** をクリックします。
10. オプション: テンプレートを変更したい場合は、**クローン** をクリックして独自の名前を入力し、**送信** をクリックします。
11. 使用したいテンプレートの名前をクリックします。
12. テンプレートをクローンした場合、変更が必要な時は **テンプレート** タブで実施できます。
13. **関連付け** タブをクリックし、テンプレートをホストが使用するオペレーティングシステムに関連付けします。
14. **ロケーション** タブをクリックして、ホストの所在を追加します。
15. **組織** タブをクリックして、ホストが属する組織を追加します。
16. **送信** をクリックして変更を保存します。
17. **ホスト > オペレーティングシステム** に移動し、ホストのオペレーティングシステムを選択します。
18. **テンプレート** タブをクリックします。
19. **PXELinux テンプレート** リストから、使用したいテンプレートを選択します。
20. **iPXE テンプレート** リストから、使用したいテンプレートを選択します。
21. **送信** をクリックして変更を保存します。
22. **ホスト > すべてのホスト** に移動し、使用したいホストを選択します。
23. **テンプレート** タブを選択し、**PXELinux テンプレート** リストから **レビュー** を選択して、そのテンプレートが正しいテンプレートであることを確認します。
24. **iPXE テンプレート** リストから **レビュー** を選択し、そのテンプレートが正しいテンプレートであることを確認します。

注記

PXELinux のエントリーがない場合、または新しいテンプレートが見つからない場合は、**ホスト > すべてのホスト** に移動し、ホスト上で **編集** をクリックします。**オペレーティングシステム** タブをクリックし、続いてプロビジョニングテンプレート **解決** ボタンをクリックして、テンプレートのリストをリフレッシュします。

25. iPXE ファームウェアのチェーンブートの無限ループを回避するには、**/etc/dhcp/dhcpd.conf** ファイルを編集し、以下の例に一致させます。分離したネットワークを使用する場合は、Satellite Server の URL ではなく、TCP port 8000 を伴う Capsule Server の URL を使用します。

    a. **/etc/dhcp/dhcpd.conf** ファイルの Bootfile Handoff セクションで、以下の行を見つけます。

    ```
    } else {
      filename "pxelinux.0";
    }
    ```

    b. else ステートメントの前に、以下に示す別の **elsif** ステートメントを追加します。

    ```
    elsif exists user-class and option user-class = "iPXE" {
      filename "http://satellite.example.com/unattended/iPXE";
    }
    ```

    c. **if** セクションが、以下の例と一致することを確認します。

    ```
    if option architecture = 00:06 {
      filename "grub2/shim.efi";
    } elsif option architecture = 00:07 {
      filename "grub2/shim.efi";
    } elsif option architecture = 00:09 {
      filename "grub2/shim.efi";
    } elsif exists user-class and option user-class = "iPXE" {
      filename "http://satellite.example.com/unattended/iPXE";
    } else {
      filename "pxelinux.0";
    }
    ```

注記

http://satellite.example.com/unattended/iPXE では、Red Hat Satellite Capsule **http://capsule.example.comf:8000/unattended/iPXE** も使用できます。アップグレードごとに **/etc/dhcp/dhcpd.conf** ファイルを更新する必要があります。**/etc/dhcp/dhcpd.conf** ファイルのコンテンツは大文字と小文字を区別します。

4.6. 本章のまとめ

本章では、Satellite Server の統合 Capsule で特定のネットワークサービスを設定する方法や、Satellite Server が制御するドメインおよびサブネットの詳細をマップする方法を説明しました。これで、新規

ホストのネットワークが提供され、また PXE ブートやネットワーク設定などの主なサービスがホストに提供されます。

次章では、新規ホストおよびホストグループを作成する方法を含む、基本的なプロビジョニングのワークフローについて扱います。

第5章 プロビジョニングワークフローの概要

本章では、Red Hat Satellite 6 におけるプロビジョニングの基本的なワークフローについて説明します。この章の内容は、後の章で特定のプロビジョニング方法を使用する際のベースとなります。

5.1. プロビジョニングワークフローの定義

プロビジョニングプロセスは、以下に概説される基本的なワークフローに従います。

1. Web UI の **ホスト > ホストの作成** の **ホストの作成** ページか、または Hammer CLI のいずれかで新規ホストを作成します。また、Satellite Server は、サブネットに関連付けられた DHCP Capsule Server から未使用の IP アドレスを要求します。**ホストの作成** ページは、この IP アドレスを IP アドレスフィールドに使用します。新規ホストのすべてのオプションの実行後には、新規ホストの要求を送信します。
2. サブネットに関連付けられた DHCP Capsule Server がホストのエントリーを予約します。
3. Satellite Server は DNS レコードを設定します。
    - 正引き DNS レコードがドメインに関連付けられた Capsule Server に作成されます。
    - 逆引き DNS レコードがサブネットに関連付けられた DNS Capsule Server に作成されます。
4. ホストの PXELinux メニューがサブネットに関連付けられた TFTP Capsule Server に作成されます。
5. 新規ホストは DHCP サーバーから DHCP リースを要求します。
6. DHCP サーバーはリース要求に応答し、TFTP オプション (**`next-server`**、**`filename`**) を返します。
7. ホストは TFTP サーバーからブートローダーおよびメニューを要求します。
8. ホストの PXELinux メニューおよび OS インストーラーは TFTP 経由で返されます。
9. インストーラーは Satellite Server から選択された **`provision`** テンプレートまたはスクリプトを要求します。
10. Satellite Server はテンプレートをレンダリングし、結果として生成されるキックスタートをホストに返します。
11. ホストは、オペレーティングシステムをインストールするビルドシステムに入り、ホストを Satellite Server に登録して管理ツール (**`katello-agent`**, **`puppet`**) をインストールします。
12. インストーラーは Satellite に対し、**`postinstall`** スクリプトで正常なビルドについて通知します。
13. PXELinux メニューはローカルの起動テンプレートに戻ります。
14. ホストはオペレーティングシステムを起動します。ホストで Puppet クラスを使用するように設定している場合は、ホストは Satellite Server に保存されるモジュールを使用して設定を行います。

このワークフローは、特定のオプションに応じて異なります。詳細については、後の章で説明します。以下はオプションの例になります。

- **Discovery** – Discovery サービスを使用している場合、Satellite Server は新規ホストの MAC アドレスを自動的に検出し、要求の送信後にホストを再起動します。Satellite でホストを再起動できるよう、ホストが割り当てられる Capusule から TCP ポート 8443 にアクセスできなければならないことに注意してください。

- **PXE を使用しないプロビジョニング** – 新規ホストの要求の送信後に、Satellite Server からダウンロードしたブートディスクで特定のホストを起動する必要があります。

- **コンピュートリソース** – コンピュートリソースは新規ホストの仮想マシンを作成し、MAC アドレスを Satellite Server に返します。また、イメージベースのプロビジョニングを使用している場合、ホストは標準の PXE ブートおよびオペレーティングシステムのインストール方法に従いません。その代わりに、コンピュートリソースは新規ホストが使用する選択したイメージのコピーを作成します。

- **コンテナー** – コンテナーのプロビジョニングプロセスは、このワークフロープロセスに従いません。

5.2. SATELLITE SERVER でのホストの作成

ホストのプロビジョニングを設定するには、Satellite Server でホストのエントリーを作成することからスタートします。Web UI または Hammer CLI のいずれかを使用できます。これは、Config Group を含むホストのプロビジョニングの基本を提供します。これらは、後の章で扱われる特定のプロビジョニング方法に関連した参照情報になります。Config Group に関する詳細は、**『Puppet Guide』** の「Using Config Groups to Manage Puppet Classes」 を参照してください。

Web UI を使用する場合

ホスト > ホストの作成 に移動します。UI には、ホストの詳細を入力できる一連のフィールドがあります。

ホスト タブで、ホストとその配置についての主な詳細を定義します。Capsule Server は、セットアップに応じて Satellite Server の統合 Capsule または外部 Capsule Server になります。

- **名前** – ホストの名前。

- **組織** – このホストを所有する組織。

- **ロケーション** – このホストのロケーション。

- **ホストグループ** – このホストのテンプレートとして使用するホストグループ。詳細については、**『アーキテクチャーガイド』** の 「ホスト分類の概念」 を参照してください。

- **デプロイ先** – ホストのデプロイメントのタイプで、ベアメタルのホスト上か、コンピュートリソースを介するか、のいずれかになります。

- **ライフサイクル環境** – ホストのアプリケーションライフサイクルにおけるステージ。

- **コンテンツビュー** – リポジトリーに使用するコンテンツビュー。

- **コンテンツソース** – コンテンツビューからコンテンツを提供するために使用する Capsule Server。外部 Capsule Server を使用する場合、コンテンツがすでに Capsule Server と同期していることを確認します。詳細については、**『インストールガイド』** の 「Capsule Server へのライフサイクル環境の追加」 を参照してください。

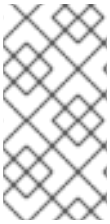
- **Puppet 環境** – ホストを含む Puppet 環境。通常、これは事前に選択されたコンテンツビューおよびライフサイクル環境を使用して定義されます。

- **Puppet マスター** – エージェント通信のマスターサーバーとして使用する Capsule Server。
- **Puppet CA** – エージェントの認定に使用する Capsule Server。
- **OpenSCAP Capsule** – OpenSCAP プロキシーとして使用する Capsule Server。

Puppet クラス タブで、どの Puppet クラスおよび Config Group をプロビジョニング後にホストに適用するか選択します。これらのクラスは、**ホスト** タブで選択されたコンテンツビューおよび Puppet 環境から取得します。**組み込み済みのクラス** セクションにはホストに適用されるクラスが表示され、**利用可能なクラス** セクションにはホストに追加できるクラスが表示されます。

インターフェース タブでは、ホストのネットワークインターフェースの設定を定義します。**インターフェースの追加** をクリックして新規インターフェースを作成するか、または**編集** をクリックして特定のインターフェースを編集します。新規または変更されたインターフェースは以下のフィールドを含むフォームを使用します。

- **Type (タイプ)** – 使用するインターフェースのタイプ。これには、基本的なイーサネット接続 **(インターフェース)** だけでなく、ベースボード管理コントローラー **(BMC)**、ボンド **(Bond)**、およびブリッジ **(Bridge)** が含まれます。これにより、ホストの複雑なネットワーク設定を作成することができます。
- **MAC アドレス** – インターフェースの MAC アドレス。これを使用して、ネットワークの詳細を特定のインターフェースにマッピングできます。さらに、プロビジョニングインターフェースの MAC アドレスは、PXE ブート時にベアメタルホストを特定するために使用されます。
- **デバイス ID** – **`eth0`**、**`ens8`**、**`bond0`**、および **`br0`** などのインターフェース ID。
- **DNS 名** – ホストのドメイン名。これには通常、**ホスト** タブのホスト名が自動的に設定されます。
- **ドメイン** – ホストのプロビジョニングに使用するドメイン。これと**DNS 名** の組み合わせにより、ホストの完全修飾ドメイン名 (FQDN) が作成されます。
- **サブネット** – このインターフェースに接続するネットワーク。
- **IP アドレス** – このインタフェースの IP アドレス。選択される **サブネット** とそのオプションにより、このフィールドにはデータが自動的に設定されます。
- インターフェースタイプのセレクションには以下が含まれます。
  - **管理** – このインターフェースはプロビジョニングの間、DHCP、DNS、および TFTP を提供します。さらに、インターフェースの設定を使用して、ホストのインターフェース設定ファイルが生成されます。個別のサービスを無効にするには、**インフラストラクチャー> サブネット** そして **インフラストラクチャー> ドメイン** へ行き、該当する Capsule 設定を **なし** に設定します。
  - **プライマリー** – メインのインターフェースで、インターフェースの詳細からホストの FQDN を構築します。
  - **プロビジョニング** – PXE ブートサービスのインターフェースです。イメージをベースにしたプロビジョニングの場合、このインターフェースの IP アドレスがクライアントへの SSH 接続に使用されます。
  - **リモート実行** – このインターフェースはリモート実行機能に使用されます。



注記

このフォームには、選択されたネットワークインターフェースの **タイプ** に関連した追加のフィールドも表示されます。たとえば、**Bond** を選択すると、ボンディングモードを設定するためのオプション、ボンディングオプション、およびボンドに割り当てるデバイスを選択するオプションが提供されます。

オペレーティングシステム タブで、ホストにインストールするオペレーティングシステムおよび関連分野を定義します。ホストの **アーキテクチャー** を選択してから、そのアーキテクチャー に関連する **オペレーティングシステム** を選択します。選択したオペレーティングシステムに応じて、様々なオプションが利用できます。

- **ビルドモード** – ホストのプロビジョニングとオペレーティングシステムのインストールを可能にします。このオプションは、すべてのプロビジョニングタスクで必要です。すでに存在するプロビジョニングされたホストのエントリーを作成する場合にのみ、このオプションを無効にする必要があります。

- **メディアセレクション** – 同期したキックスタートのリポジトリーから、またはすべてのリポジトリーからの選択を可能にします。このホストのプロビジョニングに使用するインストールメディアのタイプを選択します。同期したキックスタートリポジトリーの場合は **同期したコンテンツ** を選択します。または、**すべてのメディア** を選択し、他のインストールメディアから選択します。通常、これは **ホスト > インストールメディア > 新規メディア** で手動で追加されたものになります。

- **メディア** – オペレーティングシステムのインストールメディア。通常、これはキックスタートツリーですが、ローカルにマウントされた ISO イメージの場合もあります。

- **パーティションテーブル** – ルートディスクのレイアウトに使用するパーティションテーブルのテンプレート。このフォームで **カスタムパーティションテーブル** を直接定義することもできます。

- **PXE ローダー** - Red Hat Satellite 6.3 は、BIOS および UEFI の両システムのブートをサポートします。ホストが PXE プロビジョニングを使用する場合、正しい DHCP ファイルを選択してロードする必要があります。ホストが iPXE などの PXE を使用しないプロビジョニングを使用する場合、**なし** を選択します。

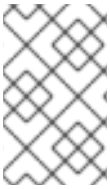
- **ルートパスワード** – オペレーティングシステムにおけるルートユーザーのパスワード。

- **プロビジョニングテンプレート** – ホストをプロビジョニングするために選択されたテンプレートが表示されます。**解決** をクリックすると、Satellite Server がテンプレートをプロビジョニングプロセスの特定の機能に割り当てる方法を確認できます (PXE、プロビジョニング、ユーザーデータなど)。

注記

オペレーティングシステム タブには、**ホスト** タブの **デプロイ先** でコンピュートリソースを選択している場合には追加のオプションが表示されます。これらのオプションについては後の章で説明します。

パラメーター タブでは、プロビジョニングプロセスと Puppet 設定の両方に変数データを設定します。**Puppet クラスパラメーター** セクションでは、Puppet のパラメーターに送信されたデータを変更することができます。**グローバルパラメーター** および **ホストパラメーター** では、プロビジョニングテンプレートなどの Satellite Server 内で使用できるカスタムパラメーターを定義できます。



注記

アクティベーションキーをホストに割り当てる場合、**名前** を **kt_activation_keys** に設定し、**値** をアクティベーションキーの名前に設定して新規のホストパラメーターを追加します。

追加情報 タブでは、ホストの所有者、レポートに組み込むかどうか、ハードウェアモデルおよび追加のコメントなど、ホストについての各種データを定義します。

ホストのエントリーを保存するには、**送信** をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成します。以下は例になります。

```
# hammer host create --name "testhost" --organization "ACME" \
--location "New York" --environment "Test" --architecture "x86_64" \
--build true --domain "example.com" --enabled true \
--mac "aa:aa:aa:aa:aa:aa" --subnet "ACME's Internal Network" \
--managed true --medium "Red Hat Kickstart Tree" \
--operatingsystem "RedHat 7.2" --owner admin \
--partition-table "Kickstart Default" \
--puppet-proxy "satellite.example.com" \
--puppet-ca-proxy "satellite.example.com" --root-password "p@55w0rd!"
```

--interface オプションを使用して、特定のインターフェース設定を行います。詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。**hammer host interface create** コマンドで特定のネットワークインターフェース設定を定義することもできます。**--host** または **--host-id** オプションを使用して、インターフェースを受信するホストを特定します。以下に例を示します。

```
# hammer host interface create --host "testhost" --type interface \
--mac "aa:aa:aa:aa:aa:aa" --identifier "eth0" --name "testhost" \
--domain "example.com" --subnet "ACME's Internal Network" \
--managed true --primary true --provision true
```

この手順は、大半のプロビジョニング方法のベースとなる手順です。ただし、各ホストのすべての情報を定義するプロセスには時間がかかります。そのため、ホストグループを作成してすべてのホストに共通する設定を定義することをお勧めします。

5.3. SATELLITE SERVER でのホストグループの作成

数多くのホストをプロビジョニングする場合、ホストのすべての詳細を毎回入力すると多大な時間がかかります。ただし、Red Hat Satellite 6 では **ホストグループ** の概念が導入されているため、ホストのプロビジョニングに要する時間を短縮できます。

ホストグループは、共通するホスト設定のテンプレートとして機能します。これには、ホストに提供する詳細情報として同じものが多数含まれます。新規ホストをホストグループを使ってプロビジョニングする場合、ホストはホストグループから定義された設定を継承します。その後に、ホストを個別化するために必要に応じて追加の詳細情報を指定できます。

さらに、ホストグループの階層を作成することができます。通常は、組織内のすべてのホストを代表するベースレベルのホストグループを 1 つ設定し、一般的な設定を提供します。続いて、特定の設定を提供するためにネスト化したグループを設定します。たとえば、オペレーティングシステムを定義する

ベースレベル (親) のホストグループと、ベースレベルのホストグループを継承する 2 つのネスト化した (子) ホストグループを設定することができます。

- **Hostgroup: Base** (Red Hat Enterprise Linux 7.2)
  - **Hostgroup: `Webserver`** (**`httpd`** Puppet クラスを適用)
    - **Host: `webserver1.example.com`** (Web サーバー)
    - **Host: `webserver2.example.com`** (Web サーバー)
  - **Hostgroup: `Storage`** (**`nfs`** Puppet クラスを適用)
    - **Host: `storage1.example.com`** (ストレージサーバー)
    - **Host: `storage2.example.com`** (ストレージサーバー)
  - **Host: `custom.example.com`** (カスタムホスト)

この例では、プロビジョニングされたすべてのホストは **Base** ホストグループの継承により、Red Hat Enterprise Linux 7.2 をオペレーティングシステムとして使用します。2 つの Web サーバーホストは **`Webserver`** ホストグループからの設定を継承します。これには、**`httpd`** Puppet クラスおよび **Base** ホストグループの設定が含まれます。同様に、2 つのストレージサーバーは **`Storage`** ホストからの設定を継承します。これには、**`nfs`** Puppet クラスおよび **Base** ホストグループの設定が含まれます。カスタムホストは **Base** ホストグループからの設定のみを継承します。

このシナリオでは、ACME のホストグループを作成する方法を示します。本書の後の章では、このホストグループをプロビジョニングプロセスで使用します。

Web UI を使用する場合

設定 > ホストグループ に移動し、**ホストグループの作成** をクリックします。UI には、ホストの作成フォームと同様のフィールドを含むフォームが表示されます。以下の詳細を入力します。

- **ホストグループ** タブでは、以下を入力します。
  - **親** – 基本的な設定を継承する親のホストグループ。最初のグループを作成する際は適用されないので、このシナリオでは空白のままにします。
  - **名前** – ホストグループの名前。このシナリオの場合、**Base** と入力します。
  - **ライフサイクル環境**– アプリケーションライフサイクルにおけるホストのステージ。『**コンテンツ管理ガイド**』 の 「アプリケーションライフサイクルの作成」で作成された **`Production`** **(本番)** 環境を選択します。
  - **コンテンツビュー** – リポジトリーに使用するコンテンツビュー。『**コンテンツ管理ガイド**』 の 「コンテンツビューの管理」 で作成された **ベース** ビューを選択します。
  - **コンテンツソース**– コンテンツビューからコンテンツを提供するために使用する Capsule Server。Satellite Server の統合 Capsule を選択します。
  - **Puppet 環境** – ホストを含む Puppet 環境。通常、これは事前に選択されたコンテンツビューおよびライフサイクル環境を使用して定義されます。この例では、Puppet モジュールを含まない **`Production`** **(本番)** 環境を選択します。

注記

Puppet は、**Production** 環境内に作成した Puppet 環境に関連付けられているホストグループを持つホストの登録時に、Puppet CA 証明書を取得することができません。ホストグループに関連付けられる適切な Puppet 環境を作成するには、以下の手順を実行します。

1. 手動でディレクトリーを作成して、所有者を変更します。

```
# mkdir
/etc/puppet/environments/example_environment
# chown apache
/etc/puppet/environments/example_environment
```

2. **設定** → **環境** へと移動し、**環境をインポート** をクリックします。ボタン名には、内部または外部の Capsule の FQDN が含まれます。

3. 作成したディレクトリーを選択し、**更新** をクリックします。

- **Puppet マスター** – エージェント通信のマスターサーバーとして使用する Capsule Server。Satellite Server の統合 Capsule を選択します。
- **Puppet CA** – エージェントの認定に使用する Capsule Server。Satellite Server の統合 Capsule を選択します。
- **OpenSCAP Capsule** – SCAP コンテンツを取得し、ARF レポートをアップロードするために使用する OpenSCAP Capsule。この例では、空白のままにします。

- **Puppet クラス** タブでは、プロビジョニング後にホストに適用する Puppet クラスを選択します。このシナリオでは Puppet クラスを使用しないため、このタブは省略します。
- **ネットワーク** タブ:
  - **ドメイン** – ホストのプロビジョニングに使用するドメイン。これと **DNS 名** との組み合わせにより、ホストの完全修飾ドメイン名 (FQDN) が作成されます。ACME の **example.com** ドメインを選択します。
  - **IPv4 サブネット** – このインターフェースに接続するネットワーク。**ACME の内部ネットワーク** を選択します。
  - **IPv6 サブネット** – このインターフェースに接続するネットワーク。この例では、空白のままにします。
  - **レルム** – ホストの認証レルム。このシナリオではレルムを使用しないので、このフィールドを空白のままにします。
- **オペレーティングシステム** タブ:
  - **アーキテクチャー** – ホストのアーキテクチャー。**x86_64** を選択します。
  - **オペレーティングシステム** – インストールするベースオペレーティングシステム。Red Hat Enterprise Linux 7.2 のエントリーは、同期の手順の実行後に表示されるはずです。手順の説明は、『**コンテンツ管理ガイド**』の 「Red Hat リポジトリーの同期」 を参照してください。エントリーを選択します。
  - **メディアセレクション** – 同期したキックスタートのリポジトリーから、またはすべてのリ

ポジトリーからの選択を可能にします。このホストグループのプロビジョニングに使用するインストールメディアのタイプを選択します。同期したキックスタートリポジトリーの場合は **同期したコンテンツ** を選択します。他のインストールメディアの場合は**すべてのメディア** を選択します。通常、これは **ホスト > インストールメディア > 新規メディア** で手動で追加されたものになります。

- **メディア** – オペレーティングシステムのインストールメディア。Red Hat Enterprise Linux 7.2 からキックスタートツリーを選択します。これは、同期の手順の実行後に表示されるはずです。手順の説明は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- **パーティションテーブル** – ルートディスクのレイアウトに使用するパーティションテーブルのテンプレート。**Default Kickstart** を選択します。
- **ルートパスワード** – オペレーティングシステムにおけるルートユーザーのパスワード。ルートパスワードを入力します。

- **パラメーター** タブでは、プロビジョニングプロセスと Puppet 設定の両方に変数データを設定します。このセクションは空白にします。
- **ロケーション** タブで、ホストグループのロケーションを設定します。この例では、**New York** を選択します。
- **組織** タブで、ホストグループを使用できる組織を設定します。この例では、**ACME** を選択します。
- **アクティベーションキー** タブでは、**サンプル**のアクティベーションキーを選択します。これにより、登録に使用するアクティベーションキーを定義する新規のパラメーター (**kt_activation_keys**) が各ホストに追加されます。

送信 をクリックしてホストグループを保存します。

CLI を使用する場合

hammer hostgroup create コマンドでホストグループを作成します。以下は例になります。

```
# hammer hostgroup create --name "Base" \
--lifecycle-environment "Production" --content-view "Base" \
--environment "production" --content-source-id 1 \
--puppet-ca-proxy-id 1 --puppet-proxy-id 1 --domain "example.com" \
--subnet `ACME's Internal Network` --architecture "x86_64" \
--operatingsystem "RedHat 7.2" --medium-id 9 \
--partition-table "Kickstart default" --root-pass "p@55w0rd!" \
--locations "New York" --organizations "ACME"
```

サーバーはホストグループのエントリーを作成します。このシナリオでは、プロビジョニングのサンプル用にこのホストグループを使用します。

5.4. 本章のまとめ

本章では、Red Hat Satellite 6 で新規ホストのエントリーを作成するための基本的なワークフローを説明しました。また、新規ホストの作成時にホストグループを作成して特定のパラメーターを事前に定義する方法についても説明しました。

次章では、ベアメタルホストのプロビジョニング方法について説明します。次章では、**Base** ホストグループを使用してホストの設定を事前に定義します。

第6章 ベアメタルホストのプロビジョニング

本章では、Red Hat Satellite 6 でベアメタルインスタンスをプロビジョニングする主な方法について説明します。これらには以下が含まれます。

- **無人プロビジョニング (Unattended Provisioning):** MAC アドレスでホストを特定し、Satellite Server は PXE ブートプロセスでプロビジョニングを実行します。
- **Discovery を使用した無人プロビジョニング (Unattended Provisioning with Discovery):** 新規ホストは PXE ブートを使用して Satellite Discovery サービスをロードします。このサービスはホストのハードウェア情報を特定し、ホストをプロビジョニング可能なホストとして一覧表示します。
- **PXE を使用しないプロビジョニング (PXE-less Provisioning):** ブートディスクまたは Satellite Server が生成する PXE なしの Discovery イメージを使用して新規ホストをプロビジョニングする機能です。
- **Discovery を使用した PXEを使用しないプロビジョニング (PXE-less Provisioning with Discovery):** 新規ホストは Satellite Discovery サービスをロードする ISO ブートディスクを使用します。このサービスはホストのハードウェア情報を特定し、ホストをプロビジョニング可能なホストとして一覧表示します。

注記

Red Hat Satellite の以前のバージョンでは、プロビジョニング時にホストグループベースのテンプレートのレンダリング機能を使用できました。この機能により、ユーザーは単一ホストではなくホストグループのテンプレートをレンダリングできました。しかしながら、この機能は、ホストのレコードまたは監査証跡がないなどの多少の制限により Red Hat Satellite 6.3 ではサポートされていません。そのため、同様の機能を提供する Discovery 機能を使用することをお勧めします。

6.1. ベアメタルプロビジョニングの要件定義

ベアメタルプロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- ベアメタルホストのネットワークを管理する Capsule Server。無人プロビジョニングおよび Discovery ベースのプロビジョニングの場合、Satellite Server には PXE サーバーの設定が必要です。詳細は、*4章ネットワークの設定* を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。
- テスト目的で使用する空のベアメタルホスト。

6.2. 無人プロビジョニングによる新規ホストの作成

無人プロビジョニングは、ホストのプロビジョニングの最も単純な形態です。この方法では、ホストの詳細を Satellite Server に入力し、ホストを起動する必要があります。Satellite Server は PXE 設定の管理や、ネットワークサービスの整理、およびホストのオペレーティングシステムと設定の提供を自動的に実行します。このプロビジョニングの方法では、プロセス時に最小限の対話を使用します。

このシナリオでは、ホストを ACME のプライベートネットワークでプロビジョニングする方法を示します。この例では、ベアメタルホストが ACME のプライベートネットワーク (192.168.140.0/24) に接続し、MAC アドレスとしての **aa:aa:aa:aa:aa:aa** を持つインターフェースを使用します。

Web UI を使用する場合

ホスト > ホストの作成 に移動します。UI には、ホストの詳細を入力できる一連のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**baremetal-test1** と入力します。
  - プロビジョニングコンテキスト (**組織** および **ロケーション**) は **ACME** および **New York** に自動的に設定されます。
  - **ホストグループ**フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - ホストの MAC アドレスを入力します。この例では、MAC アドレスは **aa:aa:aa:aa:aa:aa** です。これは、PXE ブートのプロセス時のホストの識別に必要となるため重要です。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニングテンプレート**の **解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。これには、以下が含まれます。
    - **PXELinux テンプレート: Kickstart default PXELinux**
    - **provision テンプレート: Satellite Kickstart Default**
      プロビジョニングテンプレートの関連付けについての詳細は、「プロビジョニングテンプレートの作成」 を参照してください。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成します。以下は例になります。

```
# hammer host create --name "baremetal-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" --mac "aa:aa:aa:aa:aa:aa" \
--build true --enabled true --managed true
```

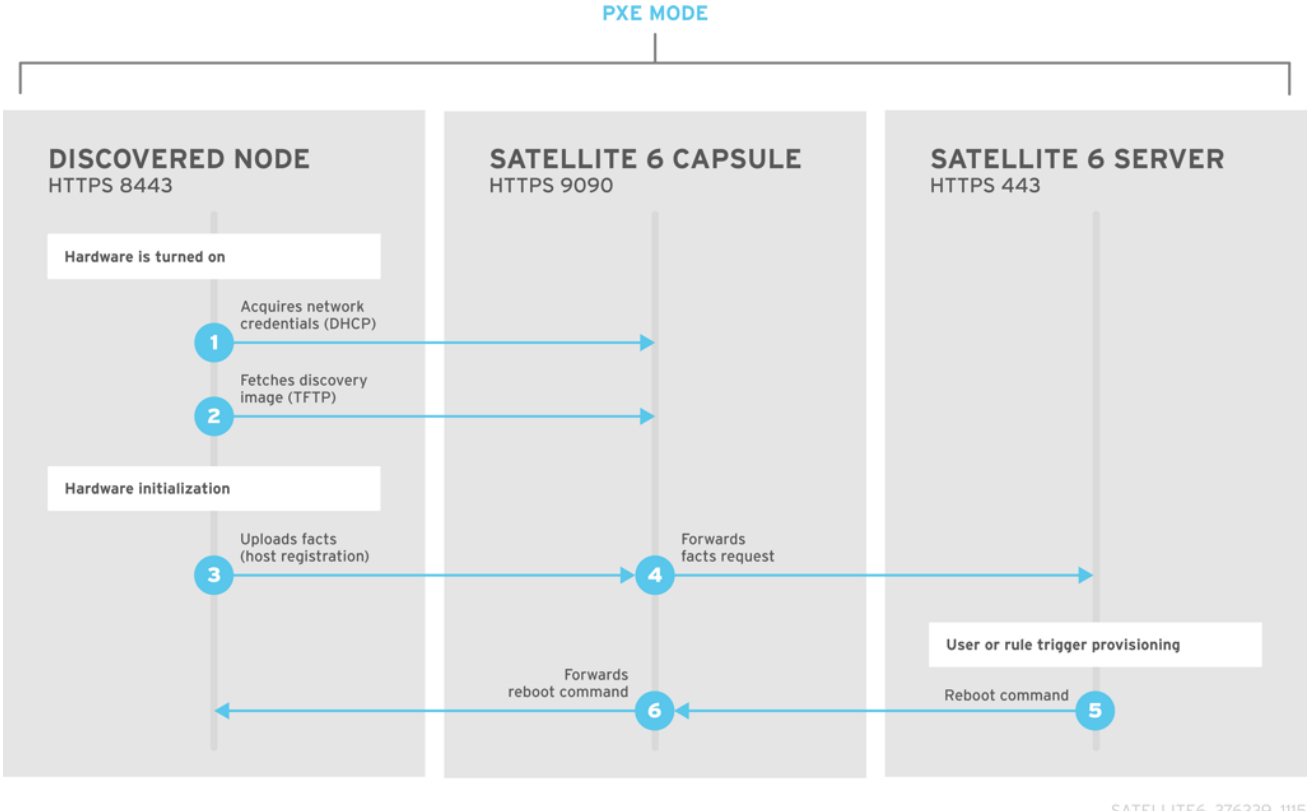
hammer host interface update コマンドを使用し、ネットワークインターフェースのオプションが設定されていることを確認します。以下は例になります。

```
# hammer host interface update --host "test1" --managed true \
--primary true --provision true
```

これで、ホストのエントリーおよび関連するプロビジョニングの設定が作成されます。これには、PXE を使用したベアメタルホストの起動に必要なディレクトリーおよびファイルを作成することも含まれます。物理ホストの電源をオンにして、ブートモードを PXE に設定すると、ホストは Satellite Server の統合 Capsule の DHCP サービスを検出し、キックスタートツリーから Red Hat Enterprise Linux 7.2 のインストールを開始します。インストールが完了すると、ホストは **サンプル** アクティベーションキーを使用して Satellite Server にも登録し、Red Hat Satellite Tools リポジトリーから必要な設定および管理ツールをインストールします。

6.3. RED HAT SATELLITE の DISCOVERY サービスの設定

Red Hat Satellite は Discovery 機能を提供します。これにより、ネットワーク上の空のホストを自動的に検出できるようになります。これらのホストは、ハードウェアの検出を実行し、この情報を Satellite Server に送り返す特殊なイメージを起動します。これにより、各ホストの MAC アドレスを入力せずに Satellite Server でプロビジョニング可能なホストのプールの作成を実行できます。



インストール

Discovery サービスを使用する前に、Satellite Server に Discovery イメージをインストールし、Discovery プラグインを有効にする必要があります。

satellite-installer コマンドで **--enable-foreman-plugin-discovery** オプションを使用してプラグインを有効にします。

```
# satellite-installer --enable-foreman-plugin-discovery
```

これにより、Satellite Server に Discovery サービスプラグインがインストールされ、有効にされます。インストールの完了後に以下のパッケージをインストールします。

```
# yum install foreman-discovery-image rubygem-smart_proxy_discovery
```

- **foreman-discovery-image** パッケージは、Discovery ISO を **/usr/share/foreman-discovery-image/** ディレクトリーにインストールし、**livecd-iso-to-pxeboot** ツールを使用してこの ISO から PXE ブートイメージも作成します。ツールはこの PXE ブートイメージを **/var/lib/tftpboot/boot** ディレクトリーに保存します。
- **rubygem-smart_proxy_discovery** パッケージは、Capsule Server (Satellite Server の統合 Capsule など) が Discovery サービスのプロキシーとして機能するように設定します。

インストールの完了後に、新規メニューオプションが Satellite Server の Web UI の **ホスト > Discovered ホスト** の下に表示されます。

Capsule Server での Discovery サービスの有効化

以下の手順を実行し、Capsule Server で Discovery サービスを有効にします。

1. 必要な Capsule Server で以下のコマンドを順番に実行します。

```
# yum install foreman-discovery-image rubygem-smart_proxy_discovery
-y
```

```
# katello-service restart
```

2. Satellite Web UI にログインし、**インフラストラクチャー> Capsule (スマートプロキシー)**に移動します。
3. Capsule Server をクリックし、リフレッシュボタンをクリックします。Capsule Server は実行済みのコマンドに対応します。設定されたサービスを見ると、**Discovery** が一覧表示されているのがわかります。これは、Discovery サービスが実行されていることを意味します。

プロビジョニングテンプレート

ホスト > プロビジョニングテンプレート セクションの **PXELinux global default** テンプレートには、Discovery サービスのエントリーが含まれています。

```
LABEL discovery
   MENU LABEL (discovery)
   KERNEL boot/fdi-image-rhel_7-vmlinuz
   APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force rd.luks=0
```

```
rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset
proxy.url=https://SATELLITE_CAPSULE_URL:9090 proxy.type=proxy
   IPAPPEND 2
```

KERNEL と **APPEND** オプションは、Discovery イメージおよび ramdisk を起動します。また、**APPEND** オプションには、プロビジョニングに使用する Capsule Server の URL を指定する **proxy.url** パラメーターが含まれることに注意してください。**SATELLITE_CAPSULE_URL** を必要なプロビジョニング Capsule の名前に編集します。このシナリオでは、これは Satellite Server の統合 Capsule になります。

```
proxy.url=https://satellite.example.com:9090
```

注記

Discovery サービスを空のホストに対して起動するデフォルトサービスに変更することができます。**PXELinux global default** の ONTIMEOUT 値を以下のように編集します。

```
ONTIMEOUT Discovery
```

ここでは、**PXELinux global default** テンプレートから Satellite Server のデフォルト PXE テンプレートへの変更をプッシュする必要があります。**ホスト > プロビジョニングテンプレート** に移動し、**PXE デフォルトのビルド** をクリックします。これにより、Satellite Server でデフォルトの PXE テンプレートがリフレッシュされます。

サブネット

Discovery 可能なホストを含むすべてのサブネットでは、Discovery サービスを提供するために適切な Capsule Server が選択されている必要があります。これを実行するには、**インフラストラクチャー > Capsule** に移動し、使用したい Capsule Server が Discovery 機能を一覧表示するか確認します。一覧表示がない場合は、**Refresh features (機能のリフレッシュ)** をクリックすると、それがすぐに表示されます。

インフラストラクチャー > サブネット に移動してサブネットを選択します。次に Capsule タブをクリックし、使用したい **Discovery プロキシー** を選択します。該当するそれぞれのサブネットにこれを実行します。

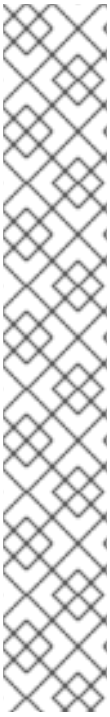
Testing (テスト)

192.168.140.0/24 ネットワークで Discovery サービスをテストし、空のベアメタルホストを起動します。ブートメニューが表示され、2 つのオプションが表示されます。

- **(local)**: ハードディスクから起動します。
- **(discovery)**: Discovery サービスで起動します。

(discovery) を選択して Discovery イメージを起動します。数分後に Discovery イメージの起動を完了すると、ステータス画面が表示されます。

ホスト > Discovered ホスト に移動します。一覧には新たに Discovered ホストが含まれます。Discovered ホストは MAC アドレスに基づいてホスト名を自動的に定義します。たとえば、Satellite は、MAC アドレスの ab:cd:ef:12:34:56 を持つ Discovered ホストのホスト名が **macabcdef123456** となるように設定します。このホスト名は、ホストのプロビジョニング時に変更することができます。



注記

Satellite Server は以下のルールに従って、組織とロケーションを Discovered ホストに割り当てます (上から下の順)。

- **Discovery 組織** または **Discovery ロケーション** の設定 (ある場合)。これらは **管理 > 設定 > Discovered** で設定できます。
- ホストの **foreman_organization** または **foreman_location** ファクトの設定。検索されるファクト名は **管理 > 設定 > Puppet** セクションで、**デフォルト組織** および **デフォルト ロケーション** のファクト設定として設定できます。
- Discovered ホストが Satellite で定義されるサブネットを使用する場合、サブネットに関連付けられた最初の組織およびロケーションを使用します。
- 名前順で最初にリストされた組織とロケーションを選択します。

組織またはロケーションは、**Discovered ホスト** ページの一括処理メニューを使用して変更できます。Discovered ホストを選択し、**アクションの選択** メニューから **組織の割り当て** または **ロケーションの割り当て** を選択します。

6.4. DISCOVERED ホストからの新規ホストの作成

Discovered ホストのプロビジョニングは、PXE のプロビジョニングと同様のプロビジョニングプロセスを踏みます。主な違いは、ホストの MAC アドレスを手動で入力する代わりに、Discovered ホストの一覧からプロビジョニングするホストを選択できる点です。

Web UI を使用する場合

ホスト > Discovered ホスト に移動します。これにより、ACME の Discovered ホストの一覧が表示されます。いずれかのホストを選択し、一覧の右側にある **Provision (プロビジョニング)** をクリックします。UI には、ホストの詳細を入力できる一連のフィールドがあります。

- **ホスト** タブ:
  - ホストの新規の **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**baremetal-test2** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は **ACME** および **New York** に自動的に設定されます。
  - **ホストグループ**フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
    - Satellite Server は Discovery の結果より MAC アドレスを自動的に設定します。

- 詳細を確認します。
- Satellite Server は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。

- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニングテンプレート** の **解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。これには、以下が含まれます。
    - **PXELinux テンプレート: `Kickstart default PXELinux`**
    - **provision テンプレート: `Satellite Kickstart Default`**

  プロビジョニングテンプレートの関連付けについての詳細は、「プロビジョニングテンプレートの作成」 を参照してください。

- **パラメーター** タブ:
  - **`kt_activation_keys`** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。
  - **送信** をクリックします。

CLI を使用する場合

1. プロビジョニングに使用する Discovered ホストを特定します。

```
# hammer discovery list
```

2. ホストを選択し、これをプロビジョニングするために **Base** ホストグループを使用します。**--new-name** オプションを使用して新規のホスト名を設定します。

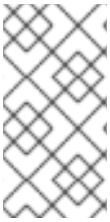
```
# hammer discovery provision --name "macaaaaaaaaaaaa" \
--new-name "baremetal-test2" \
--hostgroup "Base" --build true \
--enabled true --managed true
```

これにより、Discovered ホストの一覧からホストが削除され、関連するプロビジョニング設定でホストのエントリーが作成されます。Discovery イメージはホストを自動的にリセットし、ホストが PXE で起動できるようにします。ホストは Satellite Server の統合 Capsule で DHCP サービスを検出し、キックスタートツリーから Red Hat Enterprise Linux 7.2 のインストールを開始します。インストールが完了すると、ホストは **サンプル** アクティベーションキーを使用して Satellite Server にも登録し、Red Hat Satellite Tools リポジトリーから必要な設定および管理ツールをインストールします。

6.5. DISCOVERY ルールの作成

Discovered ホストのプロビジョニングプロセスの自動化方法として、Red Hat Satellite 6 は Discovery ルールを作成する機能を提供します。これらのルールは、Discovered ホストが、割り当てられたホス

トグループをベースに自らを自動的にプロビジョニングする方法を定義します。たとえば、CPU 数の多いホストをハイパーバイザーとして自動的にプロビジョニングすることができます。同様に、ハードディスクが大容量のホストは、ストレージサーバーとしてプロビジョニングすることもできます。



注記

現在、自動プロビジョニングでは NIC の設定を許可していません。すべてのシステムは、Discovery の際に検出された NIC 設定でプロビジョニングされています。しかし、NIC は Anaconda キックスタート、スクリプトレット、または設定管理を使用して後で設定することができます。

Web UI を使用する場合

ルールを作成するには、**設定 > Discovery ルール** に移動します。これにより、既存ルールの一覧が表示されます。**新規ルール** を選択すると、UI にルールの詳細を入力する一連のフィールドがあります。

- **名前:** ルールを表すテキスト形式の名前。例:**Hypervisor**
- **検索 -** ホストをプロビジョニングするかどうかを決定するためのルール。このフィールドには、入力する値についての推奨案が提供され、複数のルールに演算子を使用できます。例: **cpu_count > 8**
- **ホストグループ -** このホストのテンプレートとして使用するホストグループ。
- **ホスト名 -** 複数ホストのホスト名を決定するパターンです。これはプロビジョニングテンプレートと同じ ERB 構文を使用します。ホスト名には、ホスト固有の値に **@host** 属性を使用したり、乱数に **rand** 関数を使用したりできます。以下に例を示します。
  - **myhost-<%= rand(99999) %>**
  - **abc-<%= @host.facts['bios_vendor'] + '-' + rand(99999).to_s %>**
  - **xyz-<%= @host.hostgroup.name %>**
  - **srv-<%= @host.discovery_rule.name %>**
  - **server-<%= @host.ip.gsub('.','-') + '-' + @host.hostgroup.subnet.name %>**

  rand() 関数は、ストリングと連結できない整数を返します。したがって、この例では**to_s** 関数の呼び出しが必要な点に注意してください。ホスト名のパターンを作成する際に、作成されるホスト名が固有の名前であることを確認してください。ホスト名は数字で始めることができません。また、アンダースコアあるいはドットを含むこともできません。適切な方法は、Facter が提供する固有の情報 (MAC アドレス、BIOS、またはシリアル ID など) を使用することです。
- **ホストの制限:** ルールを使ってプロビジョニングできるホストの最大数。無制限に設定するには 0 を使用します。
- **優先度 -** ルール間の優先度です。値が低いルールほど優先度が高くなります。
- **有効化 -** ルールの有効化。

Satellite Server はルールの現在のプロビジョニングコンテキストを使用します。追加のコンテキストは、**組織** および **ロケーション** タブで選択できます。

送信 をクリックしてルールを保存します。

ホスト > Discovered ホスト に移動し、以下のいずれかを選択します。

- **Auto-Provision (自動プロビジョニング):** 右側の Discovered ホストのメニューにあります。これは単一ホストを自動的にプロビジョニングします。
- **Auto-Provision All (すべてを自動プロビジョニング):** 表の右上にあります。これはすべてのホストを自動的にプロビジョニングします。

CLI を使用する場合

hammer discovery_rule create コマンドを使用してルールを作成します。

```
# hammer discovery_rule create --name "Hypervisor" \
--search "cpu_count  > 8" --hostgroup "Base" \
--hostname "hypervisor-<%= rand(99999) %>" \
--hosts-limit 5 --priority 5 --enabled true
```

hammer discovery auto-provision コマンドを使用してホストを自動的にプロビジョニングします。

```
# hammer discovery auto-provision --name "macabcdef123456"
```

6.6. PXE を使用しないプロビジョニングによる新規ホストの作成

一部のハードウェアは PXE ブートインターフェースを提供しません。Red Hat Satellite 6 は、PXE を使用しない Discovery サービスを提供します。これは、PXE ベースサービス (DHCP と TFTP) を必要とせずに機能するものです。PXE ブートなしに新規ホストをプロビジョニングすることができます。これは、PXE を使用しないプロビジョニングとしても知られ、ホストが使用できるブート ISO の生成に関係しています。この ISO により、ホストは Satellite Server に接続して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 ISO には 4 つのタイプがあります。

ホストイメージ - 特定ホストのブート ISO。このイメージには、Satellite Server で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。ユーザーが Satellite のサブネットデータを定義し、静的ネットワークでイメージが作成されます。

完全ホストイメージ - 特定ホストのカーネルおよび初期 RAM ディスクイメージを含むブート ISO。このイメージは、ホストが正しくチェーンロードできない場合に役立ちます。プロビジョニングのテンプレートは、現在も Satellite Server からダウンロードされます。

汎用イメージ - 特定ホストに関連付けられていないブート ISO。ISO はホストの MAC アドレスを Satellite Server に送信します。ここでは、ホストのエントリーに対してマッチングが行われます。イメージは、IP アドレスの詳細を保存しません。また、ブートストラップするためにネットワークの DHCP サーバーへのアクセスを必要とします。このイメージは、Satellite Server の **/bootdisk/disks/generic** URL からも利用できます。例: **https://satellite.example.com/bootdisk/disks/generic**

サブネットイメージ - 汎用イメージと類似するが、Capsule Server のアドレスで設定されるブート ISO。このイメージは、同じサブネットのプロビジョニングした NIC を伴うすべてのホストに対して汎用性があります。

Web UI を使用する場合

ホスト > ホストの作成 に移動します。UI には、ホストの詳細を入力できる一連のフィールドがあります。

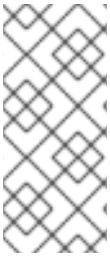
- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされるシステムのホスト名になります。この例では、**`baremetal-test3`** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は **`ACME`** および **`New York`** に自動的に設定されます。
  - **ホストグループ** フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - ホストの MAC アドレスを入力します。この例では、MAC アドレスは **`aa:aa:aa:aa:aa:aa`** です。
  - Satellite Server は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニングテンプレート** の **解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。これには、以下が含まれます。
    - **bootdisk テンプレート: `Boot disk iPXE - host`**
    - **kexec テンプレート: `Discovery Red Hat kexec`**
    - **provision テンプレート: `Satellite Kickstart Default`**

    プロビジョニングテンプレートの関連付けについての詳細は、「プロビジョニングテンプレートの作成」 を参照してください。
- **パラメーター** タブ:
  - **`kt_activation_keys`** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

これにより、ホストのエントリーが作成され、ホストの詳細ページが表示されます。ページ右上のオプ

ションには **ブートディスク** メニューが表示され、ダウンロード用として以下から 1 つのイメージが提供されます。イメージとは、**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** の 4 つです。



注記

完全ホストイメージは SYSLINUX をベースとし、ほとんどのハードウェアと機能します。iPXE ベースのブートディスク (**ホストイメージ**、**汎用イメージ**、または **サブネットイメージ**) を使用する際は、http://ipxe.org/appnote/hardware_drivers のページで、PXE ベースのブートディスクと機能することが見込まれるハードウェアドライブの一覧を参照してください。

CLI を使用する場合

hammer host create コマンドでホストを作成します。以下は例になります。

```
# hammer host create --name "baremetal-test3" --organization "ACME" \
--location "New York" --hostgroup "Base" --mac "aa:aa:aa:aa:aa:aa" \
--build true --enabled true --managed true
```

hammer host interface update コマンドを使用し、ネットワークインターフェースのオプションが設定されていることを確認します。以下は例になります。

```
# hammer host interface update --host "test3" --managed true \
--primary true --provision true
```

hammer bootdisk host コマンドで Satellite Server からブートディスクをダウンロードします。

- **ホストイメージ**向け。

  ```
  # hammer bootdisk host --host test3.example.com
  ```

- **完全ホストイメージ**向け。

  ```
  # hammer bootdisk host --host test3.example.com --full true
  ```

- **汎用イメージ** 向け。

  ```
  # hammer bootdisk generic
  ```

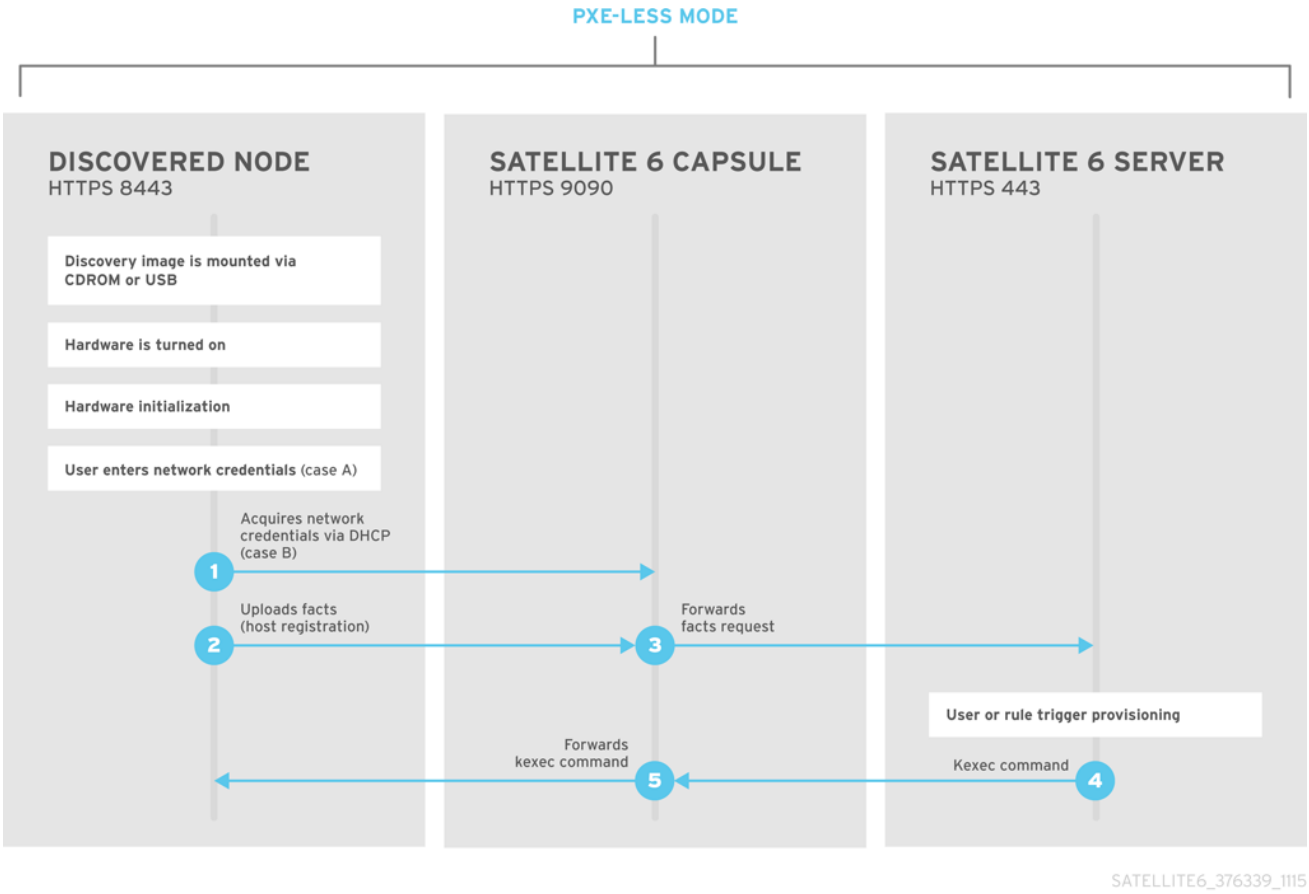
- **サブネットイメージ**向け。

  ```
  # hammer bootdisk subnet --subnet subnetName
  ```

これによりホストで使用するブート ISO が作成されます。必要に応じて、**dd** ユーティリティーまたは **livecd-tools** を使用して ISO を USB ストレージデバイスに書き込みます。 物理ホストの電源をオンにして ISO または USB ストレージデバイスから起動する場合、ホストは Satellite Server に接続し、キックスタートツリーからの Red Hat Enterprise Linux 7.2 のインストールを開始します。インストールが完了すると、ホストは **サンプル** アクティベーションキーを使用して Satellite Server にも登録し、Red Hat Satellite Tools リポジトリーから必要な設定および管理ツールをインストールします。

6.7. PXE を使用しない DISCOVERY の実装

Red Hat Satellite 6 は、PXE ベースのサービス (DHCP および TFTP) を必要とせずに機能する PXE を使用しない Discovery サービスを提供します。これは、Satellite Server の Discovery イメージを使用して実行できます。



Discovery サービスまたはイメージをインストールしていない場合は、「Red Hat Satellite の Discovery サービスの設定」の「**インストール**」セクションに従います。

Discovery サービスの ISO は **/usr/share/foreman-discovery-image/** にあり、**foreman-discover-image** パッケージを使用してインストールされます。

手動による使用

この ISO は起動可能なメディアとして機能します。このメディアを CD、DVD、または USB スティックのいずれかにコピーします。たとえば、**/dev/sdb** の USB スティックにコピーするには、以下を実行します。

```
# dd bs=4M \
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.1.1-10.iso
\
of=/dev/sdb
```

Discovery ブートメディアをベアメタルホストに挿入してホストを開始し、メディアから起動します。Discovery イメージには、**Manual network setup** または **Discovery with DHCP** のいずれかのオプションが表示されます。

- **Manual network setup** を選択する場合、Discovery イメージはネットワークオプションのセットを要求します。これには、Satellite Server に接続されるプライマリーネットワークインターフェースが含まれます。この Discovery イメージは、**IPv4 アドレス**、**IPv4 ゲートウェイ** および **IPv4 DNS** サーバーなどのネットワークインターフェースの設定オプションも要求します。一例として、ACME は以下の詳細を使用する可能性があります。

- **IPv4 Address:** 192.168.140.20
- **IPv4 Gateway:** 192.168.140.1
- **IPv4 DNS:** 192.168.140.2 (Satellite Server の 統合 Capsule)
  これらの詳細を入力した後に、**次へ** を選択します。

- **Discovery with DHCP** を選択する場合、Discovery イメージは Satellite Server に接続されるプライマリーネットワークインターフェースのみを要求します。このサービスは、Capsule Server が提供するサーバーなどの DHCP サーバーを使用してネットワークインターフェースを自動的に設定しようとします。

プライマリーインターフェースの設定後に、Discovery イメージは、Discovery サービスを提供する Satellite Server または Capsule Server の URL である **サーバー URL** を要求します。たとえば、ACME の Satellite Server で統合 Capsule を使用するには、以下の URL を使用します。

```
https://satellite.example.com:9090
```

さらに **Connection type** を **Proxy** に設定します。設定後は **次へ** を選択します。

Discovery イメージは、Facter ツールが Satellite Server に送り戻す **Custom facts (カスタムファクト)** を入力するための一連のフィールドも提供します。これらは **名前-値** の形式で入力されます。要求したカスタムファクトを指定し、**確認** を選択して継続します。

Satellite は Satellite Server の Discovery サービスとの正常な通信を報告します。**ホスト > Discovered ホスト** に移動します。一覧には、新規に Discovered ホストが含まれます。

Discovered ホストをプロビジョニングするには、「Discovered ホストからの新規ホストの作成」を参照してください。

無人での使用およびカスタマイズ

起動後のイメージの設定プロセスを自動化するカスタマイズされた Discovery ISO を作成することができます。Discovery イメージはオペレーティングシステムの Linux カーネルを使用します。これは、イメージのオペレーティングシステムを設定するためにカーネルパラメーターを渡すことを意味します。これらのカーネルパラメーターには以下が含まれます。

proxy.url

Discovery サービスを提供する Capsule Server の URL。

proxy.type

プロキシーのタイプ。通常、これは Capsule Server に接続するために **proxy** に設定されます。このパラメーターはレガシーの **Foreman** オプションもサポートします。この場合、Capsule Server ではなく Satellite Server との通信が直接行われます。

fdi.pxmac

プライマリーインターフェースの MAC アドレス (**AA:BB:CC:DD:EE:FF** 形式)。これは Capsule Server との通信に使用するインターフェースです。自動化モードでは、リンクを含む最初の NIC (ネットワーク ID をアルファベット順に使用) が使用されます。準自動化モードでは、画面が表示され、正しいインターフェースを選択するよう求められます。

fdi.pxip, fdi.pxgw, fdi.pxdns

プライマリーネットワークインターフェースの IP アドレス (**fdi.pxip**)、ゲートウェイ (**fdi.pxgw**)、および DNS (**fdi.pxdns**) を手動で設定します。これらのパラメーターを省略する場合、イメージは DHCP を使用してネットワークインターフェースを設定します。

fdi.pxfactname1, fdi.pxfactname2 ... fdi.pxfactnameN

カスタムファクト名を指定できます。

fdi.pxfactvalue1, fdi.pxfactvalue2 ... fdi.pxfactvalueN

各カスタムファクトの値。それぞれの値はファクト名に対応しています。たとえば、**fdi.pxfactvalue1** は、**fdi.pxfactname1** の名前が付けられたファクトの値を設定します。

fdi.pxauto

自動化モードまたは準自動化モードを設定します。0 に設定される場合、イメージは準自動化モードを使用します。これにより、一連のダイアログオプションによって選択肢を確認することができます。1 に設定される場合、イメージは自動化モードを使用し、確認なしに次に進みます。

Satellite Server は、**foreman-discovery-image** パッケージでツール (**discovery-remaster**) も提供します。このツールは、カーネルパラメーターを含めるようにイメージのマスターを新たに作成します。イメージのマスターを新たに作成するには、**discovery-remaster** ツールを実行します。以下は例になります。

```
# discovery-remaster ~/iso/foreman-discovery-image-3.1.1-10.iso \
"fdi.pxip=192.168.140.20/24 fdi.pxgw=192.168.140.1 \
fdi.pxdns=192.168.140.2 proxy.url=https://satellite.example.com:9090 \
proxy.type=proxy fdi.pxfactname1=customhostname \
fdi.pxfactvalue1=myhost fdi.pxmac=52:54:00:be:8e:8c fdi.pxauto=1"
```

ツールは、元の Discovery イメージと同じディレクトリーに新規の ISO ファイルを作成します。このシナリオでは、 **/usr/share/foreman-discovery-image/** の下に保存されます。

このメディアを CD、DVD、または USB スティックのいずれかにコピーします。たとえば、**/dev/sdb** の USB スティックにコピーするには、以下を実行します。

```
# dd bs=4M \
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.1.1-10.iso
\
of=/dev/sdb
```

Discovery ブートメディアをベアメタルホストに挿入してホストを開始し、メディアから起動します。

Discovered ホストをプロビジョニングするには、「Discovered ホストからの新規ホストの作成」を参照してください。

最終メモ

ホストは以下のプロビジョニングテンプレートに対して解決される必要があります。

- **kexec テンプレート: Discovery Red Hat kexec**
- **provision テンプレート: Satellite Kickstart Default**

プロビジョニングテンプレートの関連付けについての詳細は、「プロビジョニングテンプレートの作成」を参照してください。

6.8. 本章のまとめ

本章では、ベアメタルホストのプロビジョニングについて説明しました。たとえば、無人プロビジョニング、Discovery ベースのプロビジョニング、および PXE を使用しないプロビジョニングなど、複数の異なる方法を紹介しました。カーネルベースの仮想マシン (KVM) サーバー、Red Hat Virtualization、および VMware vSphere などの仮想化インフラストラクチャーからホストをプロビジョニングする際には、これらの方法の一部を使用することができます。

次章では、**libvirt** 仮想化を使用した KVM サーバーからのプロビジョニング方法について説明します。

第7章 KVM サーバー (LIBVIRT) での仮想マシンのプロビジョニング

カーネルベースの仮想マシン (KVM) はオープンソースの仮想化デーモンおよび Red Hat Enterprise Linux で実行される **libvirt** という API を使用します。Red Hat Satellite 6 は KVM サーバーで **libvirt** API に接続でき、ハイパーバイザーにホストをプロビジョニングし、特定の仮想化機能を制御することができます。本章では、ACME の KVM サーバーに接続を追加し、仮想マシンのプロビジョニングを実行します。

7.1. KVM プロビジョニングの要件定義

KVM プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- KVM サーバーでネットワークを管理する Capsule Server。Capsule Server との競合を避けるためにその他の DHCP サービスがこのネットワーク上で実行されていないことを確認します。Capsule Server のネットワークサービス設定の詳細は、*4章ネットワークの設定* を参照します。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。
- KVM 仮想化ツールを実行する Red Hat Enterprise Linux サーバー。詳細は、『Red Hat Enterprise Linux 7 仮想化スタートガイド』 を参照してください。
- 既存の仮想マシンイメージ (イメージベースのプロビジョニングを使用する場合)。このイメージが KVM ホストのストレージプールにあることを確認します。**デフォルトの** ストレージプールは通常 **/var/lib/libvirt/images** にあります。

7.2. SATELLITE SERVER での KVM 接続の設定

KVM 接続を追加する前に、Satellite Server には安全な接続を確認するための設定が必要になります。つまり、接続を実行するユーザーである **Foreman** ユーザー用に SSH キーペアを作成する必要があります。

Satellite Server で **Foreman** ユーザーに切り替えます。

```
# su foreman -s /bin/bash
```

キーペアを生成します。

```
$ ssh-keygen
```

公開キーを KVM サーバーにコピーします。以下は例になります。

```
$ ssh-copy-id root@kvm.example.com
```

以下のコマンドを使用して、KVM サーバーへの接続をテストします。

```
$ virsh -c qemu+ssh://root@kvm.example.com/system list
```

Satellite Server で KVM 接続を追加する場合、**qemu+ssh** プロトコルおよびサーバーのアドレスを使用します。たとえば、**qemu+ssh://root@kvm.example.com/system** のようになります。

7.3. SATELLITE SERVER への KVM 接続の追加

このプロセスでは、Satellite Server のコンピュートリソースで KVM 接続を追加します。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、**新規のコンピュートリソース** をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前:** リソースのテキスト形式の名前。例:**ACME's KVM Server**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**Libvirt** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**KVM server at kvm.example.com**
- **URL:** KVM サーバーへの **libvirt** 接続 URL。例: **qemu+ssh://root@kvm.example.com/system**
- **ディスプレイタイプ:** 使用するリモートアクセスプロトコルを選択します (**VNC** または **Spice** のいずれか)。
- **コンソールのパスワード:** ランダムに生成されたパスワードで新規ホストのコンソールアクセスのセキュリティーを保護します。

テスト接続 をクリックして Satellite Server が KVM サーバーに問題なく接続できることを確認します。

ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして KVM 接続を保存します。

CLI を使用する場合

hammer compute-resource create コマンドで接続を作成します。

```
# hammer compute-resource create --name "ACME's KVM Server" \
--provider "Libvirt" --description "KVM server at kvm.example.com" \
--url "qemu+ssh://root@kvm.example.com/system" --locations "New York" \
--organizations "ACME"
```

7.4. SATELLITE SERVER での KVM イメージの追加

イメージベースのプロビジョニングを使用して新規ホストを作成する場合、イメージの詳細を Satellite Server に追加する必要があります。これには、アクセスの詳細およびイメージの場所が含まれます。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、KVM 接続の名前をクリックします。UI には、**イメージ** タブを含む接続についての情報が表示されます。このタブには、新規プロバイダーのイメージは含まれませんが、新規イメージを追加することができます。**新規イメージ** をクリックすると、UI に KVM イメージの一連のフィールドが表示されます。

- **名前:** イメージのテキスト形式の名前。例:**`Test KVM Image`**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **`RedHat 7.2`**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: **`x86_64`**
- **ユーザー名:** イメージにアクセスするための SSH ユーザー名。通常、これは**`root`** ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **ユーザーデータ** - イメージが **`cloud-init`** データなどのユーザーデータ入力をサポートするかどうかを設定します。
- **イメージパス:** KVM サーバーのイメージを指す完全パス。例: **`/var/lib/KVM/images/TestImage.qcow2`**

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

`hammer compute-resource image create` コマンドでイメージを作成します。**`--uuid`** フィールドを使用して KVM サーバー上のイメージの場所の完全パスを保存します。

```
# hammer compute-resource image create --name "Test KVM Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" --username root \
--user-data false --uuid "/var/lib/libvirt/images/TestImage.qcow2" \
--compute-resource "ACME's KVM Server"
```

7.5. KVM の詳細をコンピュートプロファイルへ追加

KVM ベースの仮想マシンの特定のハードウェア設定を事前に定義することができます。これは、これらのハードウェア設定をコンピュートプロファイルに追加して実行できます。この例では、「コンピュートプロファイルの作成」 で作成した **`4-Example`** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動し、プロファイルの名前をクリックします。たとえば、「コンピュートプロファイルの作成」 で事前に作成した **`4-Example`** プロファイルを使用します。UI には、コンピュートリソースの一覧が表示されます。KVM 接続をクリックします。

UI には、プロファイルに KVM 固有の詳細を入力できる一連のフィールドがあります。これには以下が含まれます。

- **CPU:** 新規ホストに割り当てる CPU の数。
- **メモリー:** 新規ホストに割り当てるメモリーの容量。
- **イメージ:** イメージベースのプロビジョニングを実行する際に使用するイメージ。この例では、**`Test KVM Image`** を使用します。
- **ネットワークインターフェース** - ホストのネットワークインターフェースの KVM ネットワークパラメーター。複数のネットワークインターフェースを作成することができます。ただし、

少なくとも 1 つのインターフェースが Capsule で管理されるネットワークをポイントしている必要があります。ネットワークインターフェースのオプションには、以下が含まれます。

- **ネットワークタイプ** - ホストおよび NIC のネットワークのタイプ。これは、**物理 (ブリッジ)** または **仮想 (Nat)** のいずれかになります。
- **ネットワーク: ネットワークタイプ** の選択内容により、使用するホストの物理インターフェースか、または KVM サーバー上の仮想ネットワークのいずれかになります。
- **NIC タイプ** - **virtio** などの仮想 NIC タイプ。

- **ストレージ** - ホストのボリューム。ホストに複数のボリュームを作成することができます。ストレージのオプションには以下が含まれます。
  - **ストレージプール** - ボリュームを含む KVM サーバーのプール。
  - **サイズ** - ボリュームのサイズ (GB)。
  - **割り当て** - ボリュームに事前に割り当てられた物理スペースのサイズ。
  - **タイプ** - ボリュームのタイプ。**RAW** または **QCOW2** のいずれかになります。

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.3 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

7.6. KVM サーバーでのネットワークベースのホストの作成

KVM プロビジョニングプロセスは、ネットワーク接続を介して新規ホストを作成するオプションを提供します。この場合、新規ホストは Satellite Server の統合 Capsule または KVM 仮想ネットワークの外部 Capsule Server のいずれかにアクセスする必要があります。これは、ホストが PXE プロビジョニングサービスにアクセスできるようにするためです。

重要

プロビジョニングのために仮想ネットワークを KVM サーバーで使用している場合、DHCP 割り当てを提供しないネットワークを選択します。そうしないと、新規ホストの起動時に Satellite Server と DHCP との競合が生じるからです。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これは、プロビジョニングされたシステムのホスト名になります。この例では、**kvm-test1** になります。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** および **New York**
  - **ホストグループ** フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。

- **デプロイ先** で、KVM 接続を選択します。この例では、**ACME's KVM Server** になります。仮想マシンの新規タブが表示されます。
- **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **4-Example**

- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。KVM サーバーは MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
  - インターフェース画面には、コンピュートプロファイルの設定が入力された KVM 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニング方法** が **ネットワークベース** に設定されていることを確認します。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

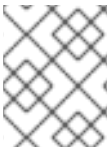
送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method build** を組み込んでネットワークベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "kvm-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's KVM Server" --provision-method build \
--build true --enabled true --managed true \
```

```
--interface
"managed=true,primary=true,provision=true,compute_type=network,compute_net
work=acmenetwork" \
--compute-attributes="cpus=1,memory=1073741824" \
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```



注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

新規ホストのエントリーは、KVM サーバーが仮想マシンを作成し、これを開始するようにトリガーします。仮想マシンが仮想ネットワーク経由で定義済みの Capsule Server を検出する場合、仮想マシンは PXE で起動し、選択したオペレーティングシステムのインストールを開始します。

7.7. KVM サーバーでのイメージベースのホストの作成

KVM プロビジョニングプロセスは、KVM サーバーで既存イメージから新規ホストを作成するオプションも提供します。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**kvm-test2** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は **ACME** および **New York** に自動的に設定されます。
  - **ホストグループ**フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
  - **デプロイ先**で、KVM 接続を選択します。この例では、**ACME's KVM Server** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル**で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **4-Example**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。KVM サーバーは MAC アドレスをホストに割り当てます。

- Satellite Server は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- インターフェース画面には、コンピュートプロファイルの設定が入力された KVM 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。

- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニング方法** が **イメージベース** に設定されていることを確認します。新規の**イメージ** フィールドが表示されます。このフィールドから、新規ホストの root ボリュームのベースとしてイメージを選択できます。これは、**ホスト** タブで選択されたコンピュートプロファイルから自動的に設定されます。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method image** を組み込んでイメージベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "kvm-test2" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's KVM Server" --provision-method image \
--image "Test KVM Image" --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=network,compute_net
work=acmenetwork" \
--compute-attributes="cpus=1,memory=1073741824" \
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

新規ホストのエントリーは、KVM サーバーが新規ボリュームのベースとして既存イメージを使用し、仮想マシンを作成するようトリガーします。

7.8. NOVNC コンソールの設定

以下の手順に従い、KVM サーバーおよびブラウザーを設定して NoVNC コンソールを使用できるようにします。

作業を開始する前の注意事項

- **Foreman** ユーザー用に SSH キーを設定する必要があります。詳細は、Satellite Server での KVM 接続の設定 を参照してください。
- 既存の Libvirt ゲストに関しては、**コンピュートリソース** 設定の **ディスプレイタイプ** が、**VNC** である点を確認します。詳細は、Satellite Server への KVM 接続の追加 を参照してください。

noVNC コンソールを設定するには、以下の手順を実行します。

1. KVM ホストシステムで、VNC サービスをポート **5900** から **5930** まで許可するようにファイアウォールを設定します。
   - Red Hat Enterprise Linux 6 の場合:

     ```
     # iptables -A INPUT -p tcp --dport 5900:5930 -j ACCEPT
     # service iptables save
     ```

   - Red Hat Enterprise Linux 7 の場合:

     ```
     # firewall-cmd --add-port=5900-5930/tcp
     # firewall-cmd --add-port=5900-5930/tcp --permanent
     ```

2. Firefox ブラウザーで、Satellite Server のパブリックダウンロードページ (例: https://satellite.example.com/pub/) に移動し、証明書ファイル **katello-server-ca.crt** をクリックします。
   a. **表示** をクリックして CA 証明書を開きます。
   b. **発行済み** 一覧で、**コモンネーム (CN)** が Satellite Server の FQDN であることを確認し、**閉じる** をクリックします。
   c. **この CA を信頼して Website を特定** を選択し、**OK** をクリックします。
3. Firefox ブラウザーで、HTTP Strict Transport Security (HSTS) を無効にします。たとえば、ブラウザーのアドレスバーに **About:Config** と入力し、以下のブール値を **True** に設定します。

   ```
   network.websocket.allowInsecureFromHTTPS
   ```

   HSTS に関する詳細情報は、HTTP Strict Transport Security (HSTS) を参照してください。
4. Satellite Web UI で、**インフラストラクチャー> コンピュートリソース** に移動し、Libvirt リソースの名前を選択します。
5. **仮想マシン** タブで、Libvirt ゲストの名前を選択します。マシンの電源がオンになっていることを確認してから、**コンソール** を選択します。

コンソールウィンドウは noVNC ハンドシェイクの完了後に表示されます。

7.9. 本章のまとめ

本章では、Red Hat Satellite 6 を設定して KVM サーバーを使用する方法と、KVM サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。ここでは、ネットワークベースのホストおよびイメージベースのホストの両方について紹介しました。

Red Hat Satellite 6 で設定が必要なコンピュートリソースが他にない場合は、13章 *プロビジョニングの最終設定* のプロビジョニングについての最終メモを参照してください。

次章では、Red Hat Virtualization 環境からのプロビジョニング方法について説明します。

第8章 RED HAT VIRTUALIZATION での仮想マシンのプロビジョニング

Red Hat Virtualization (バージョン 4.0 以降) または Red Hat Enterprise Virtualization (バージョン 3.6 以前) は、Red Hat Enterprise Linux 上に構築されたエンタープライズクラスのサーバーおよびデスクトップの仮想化プラットフォームです。Red Hat Satellite 6 は Red Hat Virtualization の REST API バージョン 3 で仮想化機能を管理できます。REST API バージョン 4 は、Satellite 6 ではまだサポートされていません。これには、新規仮想マシンの作成やそれらの電源状態の制御が含まれます。本章の目的は、ACME の Red Hat Virtualization 環境に接続を追加し、仮想マシンをプロビジョニングすることです。

8.1. RED HAT VIRTUALIZATION のプロビジョニング要件の定義

Red Hat Virtualization のプロビジョニング要件には、以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- Red Hat Virtualization 環境で、論理ネットワークを管理する Capsule Server。Capsule Server との競合を避けるために他の DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、*4章ネットワークの設定* を参照してください。
- イメージベースのプロビジョニングを使用する場合は、**空の** テンプレート以外の既存のテンプレート。仮想マシンのテンプレートを作成する方法についての詳細は、**『仮想マシン管理ガイド』** の 「テンプレート」 を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。

8.2. RED HAT VIRTUALIZATION ユーザーの作成

Red Hat Virtualization サーバーには、Satellite Server の通信用に管理者に相当するユーザーが必要です。セキュリティー上の理由により、Red Hat ではこのような通信に **`admin@internal`** ユーザーを使用しないことを推奨しています。代わりに、以下の権限を持つ新規の Red Hat Virtualization ユーザーを作成してください。

- システム
  - システムの設定
    - ログインパーミッション
- ネットワーク
  - vNIC プロファイルの設定
    - チームの作成
    - プロパティーの編集
    - 削除
    - vNIC プロファイルの VM への割り当て
    - vNIC プロファイルのテンプレートへの割り当て

- テンプレート
  - プロビジョニング操作
    - インポート/エクスポート
- 仮想マシン
  - プロビジョニング操作
    - チームの作成
    - 削除
    - インポート/エクスポート
    - ストレージの編集
- ディスク
  - プロビジョニング操作
    - チームの作成
  - ディスクプロファイル
    - ディスクプロファイルの割り当て

Red Hat Virtualization で新規ユーザーを作成し、パーミッションを追加する方法の詳細は、『**Red Hat Virtualization 管理ガイド**』の「管理ポータルからのユーザータスクの管理」を参照してください。

8.3. SATELLITE SERVER への RED HAT VIRTUALIZATION 接続の追加

このプロセスでは、Satellite Server のコンピュートリソースで Red Hat Virtualization 接続を追加します。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソースに移動し、**コンピュートリソースの作成**をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前** - リソースのプレーンテキスト形式の名前。例:**`ACME's RHV`**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**RHEV** を選択すると、新規のフィールドのセットが表示されます。
- **説明** - リソースのプレーンテキスト形式の説明。例:**`RHV-M server at rhvm.example.com`**.
- **URL** - Red Hat Virtualization Manager の API への接続 URL。たとえば、RHEV 3.6 以前のバージョンでは、この URL の形式は **`https://rhvm.example.com/api`** になります。RHV 4.0 以降のバージョンでは、この URL の形式は **`https://rhvm.example.com/ovirt-engine/api/v3`** になります。
- **ユーザー名** - Red Hat Virtualization Manager のリソースにアクセスするパーミッションを持つユーザー。例: **`satellite@internal`**
- **パスワード:** 選択されたユーザーのパスワード。

- **データセンター- URL**、**ユーザー名**、および **パスワード** を入力したら、**データセンターのロード** をクリックし、Red Hat Virtualization 環境からデータセンターの一覧を入力します。この一覧から管理する特定のデータセンターを選択します。
- **クォータ ID:** Satellite Server で利用可能なリソースを制限するためにクォータを選択します。
- **X509 証明機関:** SSL/TLS アクセスの証明機関。

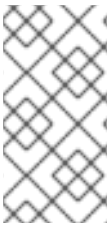
ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして接続を保存します。

CLI を使用する場合

`hammer compute-resource create` コマンドで接続を作成します。**`--provider`** で **`Ovirt`** を選択し、**`--uuid`** で使用するデータセンターの UUID を設定します。

```
# hammer compute-resource create --name "ACME's RHV" \
--provider "Ovirt" --description "RHV-M server at rhvm.example.com" \
--url "https://rhvm.example.com/api" --user "satellite@internal" \
--password "p@55w0rd!" --locations "New York" --organizations "ACME" \
--uuid 72cb9454-81cd-4231-a863-d9baf0f399f8
```



注記

RHV の 4.0 未満のバージョンでは、この URL の形式は **`https://rhvm.example.com/api`** になります。RHV 4.0 以降のバージョンでは、この URL の形式は **`https://rhvm.example.com/ovirt-engine/api/v3`** になります。

8.4. SATELLITE SERVER での RED HAT VIRTUALIZATION イメージの追加

Red Hat Virtualization は、新規仮想マシンを作成するためのイメージとしてテンプレートを使用します。イメージベースのプロビジョニングを使用して新規ホストを作成する場合、Red Hat Virtualization テンプレートの詳細を Satellite Server に追加する必要があります。これには、アクセスの詳細およびテンプレート名が含まれます。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、Red Hat Virtualization 接続の名前をクリックします。UI には、**イメージ** タブを含む接続についての情報が表示されます。このタブには、新規プロバイダーのイメージは含まれませんが、新規イメージを追加することができます。**新規イメージ** をクリックすると、UI にRed Hat Virtualization テンプレートの一連のフィールドが表示されます。

- **名前** - イメージのプレーンテキスト形式の名前。例:**`Test RHV Image`**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **`RedHat 7.2`**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: **`x86_64`**

- **ユーザー名:** イメージにアクセスするための SSH ユーザー名。通常、これは**root** ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ** - Red Hat Virtualization 上のイメージの名前。一覧からイメージ名を選択します。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

hammer compute-resource image create コマンドでイメージを作成します。**--uuid** フィールドを使用して、Red Hat Virtualization サーバー上のテンプレート UUID を保存します。

```
# hammer compute-resource image create --name "Test RHV Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" --username root \
--uuid "9788910c-4030-4ae0-bad7-603375dd72b1" \
--compute-resource "ACME's RHV"
```

8.5. RED HAT VIRTUALIZATION の詳細をコンピュートプロファイルへ追加

Red Hat Virtualization の仮想マシンの特定のハードウェア設定を事前に定義することができます。これは、これらのハードウェア設定をコンピュートプロファイルに追加することで実行できます。この例では、**4-Example** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動し、プロファイルの名前をクリックします。たとえば、事前に作成した **4-Example** プロファイルを使用します。UI には、コンピュートリソースの一覧が表示されます。Red Hat Virtualization 接続をクリックします。

UI には、プロファイルに Red Hat Virtualization 固有の詳細を入力できる一連のフィールドがあります。これには以下が含まれます。

- **クラスター** - Red Hat Virtualization 環境のターゲットホストクラスター。
- **テンプレート** - **コア数** および **メモリー** 設定に使用する RHV テンプレート。
- **コア数:** 新規ホストに割り当てる CPU コア数の数。
- **メモリー:** 新規ホストに割り当てるメモリーの容量。
- **イメージ** - イメージベースのプロビジョニングを実行する場合に使用するイメージ。この例では、**Test RHV Image** を使用します。
- **ネットワークインターフェース** - ホストのネットワークインターフェースのネットワークパラメーター。複数のネットワークインターフェースを作成することができます。ただし、少なくとも 1 つのインターフェースが Capsule で管理されるネットワークをポイントしている必要があります。ネットワークインターフェースのオプションには、以下が含まれます。
  - **名前** - ネットワークインターフェースの名前。
  - **ネットワーク** - 使用する論理ネットワーク。

- **ストレージ** - ホストのボリューム。ホストに複数のボリュームを作成することができます。ストレージのオプションには以下が含まれます。
  - **サイズ (GB)** - ボリュームのサイズ (GB)。
  - **ストレージドメイン** - ボリュームのストレージドメイン。
  - **ディスクの事前割り当て** - シンプロビジョニングまたはフルディスクの事前割り当てを設定します。
  - **ブート可能:** ブートボリュームを定義します。

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.3 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

8.6. RED HAT VIRTUALIZATION サーバーでのネットワークベースのホストの作成

Red Hat Virtualization プロビジョニングプロセスは、ネットワーク接続を介して新規ホストを作成するオプションを提供します。この場合、新規ホストは Satellite Server の統合 Capsule または Red Hat Virtualization 仮想ネットワークの外部 Capsule Server のいずれかにアクセスする必要があります。これは、ホストが PXE プロビジョニングサービスにアクセスできるようにするためです。

重要

プロビジョニング目的で Red Hat Virtualization サーバーで仮想ネットワークを使用する場合は、DHCP 割り当てを提供しない仮想ネットワークを必ず選択します。DHCP 割り当ては、新規ホストの起動時に Satellite Server との競合を引き起こします。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**`rhv-test1`** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** および **`New York`**
  - **ホストグループ** フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
  - **デプロイ先** で、Red Hat Virtualization 接続を選択します。この例では、**`ACME's RHV`** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **`4-Example`**
- **インターフェース** タブ:

- ホストのインターフェースの **編集** をクリックします。
- ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
  - **ホスト** タブの **名前** は **DNS 名** になります。
  - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
- **MAC アドレス** を空白にします。サーバーは MAC アドレスをホストに割り当てます。
- Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- インターフェース画面には、コンピュートプロファイルの設定が入力された Red Hat Virtualization 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。

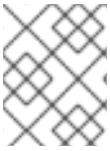
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニング方法** が **ネットワークベース** に設定されていることを確認します。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method build** を組み込んでネットワークベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "rhv-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's RHV" --provision-method build \
--build true --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_name=eth0,compute_networ
k=satnetwork" \
--compute-
attributes="cluster=Default,cores=1,memory=1073741824,start=true" \
--volume="size_gb=20G,storage_domain=Data,bootable=true"
```



注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

この新規ホストのエントリーは、Red Hat Virtualization サーバーが仮想マシンを作成するようにトリガーします。仮想マシンが仮想ネットワーク経由で定義済みの Capsule Server を検出する場合、仮想マシンは PXE で起動し、選択したオペレーティングシステムのインストールを開始します。

8.7. RED HAT VIRTUALIZATION サーバーでのイメージベースのホストの作成

Red Hat Virtualization プロビジョニングプロセスは、Red Hat Virtualization サーバーで既存イメージから新規ホストを作成するオプションも提供します。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**`rhv-test2`** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **`ACME`** および **`New York`**
  - **ホストグループ** フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
  - **デプロイ先** で、Red Hat Virtualization 接続を選択します。この例では、**`ACME's RHV`** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **`4-Example`**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白のままにします。 The Red Hat Virtualization サーバーが MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。

- インターフェース画面には、コンピュートプロファイルの設定が入力された Red Hat Virtualization 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。

- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニング方法** が **イメージベース** に設定されていることを確認します。新規の**イメージ** フィールドが表示されます。このフィールドから、新規ホストの root ボリュームのベースとしてイメージを選択できます。これは、**ホスト** タブで選択されたコンピュートプロファイルから自動的に設定されます。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。

- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。

- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method image** を組み込んでイメージベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "rhv-test2" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's RHV" --provision-method image \
--image "Test RHV Image" --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_name=eth0,compute_networ
k=satnetwork" \
--compute-
attributes="cluster=Default,cores=1,memory=1073741824,start=true" \
--volume="size_gb=20G,storage_domain=Data,bootable=true"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

この新規ホストのエントリーは、Red Hat Virtualization サーバーが新規ボリュームのベースとして既存イメージを使用し、仮想マシンを作成するようトリガーします。

8.8. 本章のまとめ

本章では、Red Hat Satellite 6 を設定して Red Hat Virtualization サーバーを使用する方法と、Red Hat

Virtualization サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。ここでは、ネットワークベースのホストおよびイメージベースのホストの両方について紹介しました。

Red Hat Satellite 6 で設定が必要なコンピュートリソースが他にない場合は、13章*プロビジョニングの最終設定*のプロビジョニングについての最終メモを参照してください。

次章では、VMware vSphere プラットフォームからのプロビジョニング方法について説明します。

第9章 VMWARE VSPHERE での仮想マシンのプロビジョニング

VMware vSphere は VMware のエンタープライズクラスの仮想化プラットフォームです。Red Hat Satellite 6 は、新規仮想マシンの作成やそれらの電源状態の制御を含む、vSphere プラットフォームとの対話を実行することができます。本章では、ACME の vSphere 環境に接続を追加し、仮想マシンをプロビジョニングします。

9.1. VMWARE VSPHERE プロビジョニングの要件定義

VMware vSphere プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- vSphere 環境でネットワークを管理する Capsule Server。Capsule Server との競合を避けるために DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、*4 章ネットワークの設定*を参照してください。
- 既存の VMware テンプレート (イメージベースのプロビジョニングを使用する場合)。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。

9.2. VMWARE VSPHERE ユーザーの作成

VMware vSphere サーバーには、Satellite Server の通信用に管理者に相当するユーザーが必要になります。セキュリティー上の理由により、この通信に **`administrator`** ユーザーを使用しないことをお勧めします。その代わりに、以下の権限を持つ新規のユーザーを作成してください。

- All Privileges (すべての権限) → Datastore (データストア) → Allocate Space (スペースの割り当て)
- All Privileges (すべての権限)→ Network (ネットワーク) → Assign Network (ネットワークの割り当て)
- All Privileges (すべての権限) → Resource (リソース) → Assign virtual machine to resource pool (仮想マシンのリソースプールへの割り当て)
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン) → Configuration (All) (設定 (すべて))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Interaction (対話)
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン) → Inventory (インベントリー)
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Provisioning (プロビジョニング)

9.3. SATELLITE SERVER への VMWARE VSPHERE 接続の追加

このプロセスは、Satellite Server のコンピュートリソースで VMware vSphere 接続を追加します。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、**新規のコンピュートリソース** をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前:** リソースのテキスト形式の名前。例:**ACME's vSphere**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**VMware** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**VMware vSphere at vsphere.example.com**
- **VCenter/Server (サーバー):** vCenter サーバーの IP アドレスまたはホスト名。例: **vsphere.example.com**
- **ユーザー名:** vCenter のリソースにアクセスするパーミッションを持つユーザー。例: **SatelliteUser**
- **パスワード:** 選択されたユーザーのパスワード。
- **データセンター: URL**、**ユーザー名** および **パスワード** を入力したら **データセンターのロード** をクリックし、VMware vSphere 環境からデータセンターの一覧を入力します。この一覧から管理する特定のデータセンターを選択します。
- **フィンガープリント:** vSphere 環境にアクセスするための証明書フィンガープリント。通常、このフィールドには選択したデータセンターのフィンガープリントが設定されます。
- **コンソールのパスワード:** ランダムに生成されたパスワードで新規ホストのコンソールアクセスのセキュリティーを保護します。
- **キャッシングの有効化 -** コンピュートリソースのキャッシングを有効化します。詳細は、「コンピュートリソースのキャッシング」 を参照してください。

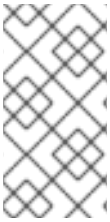
ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして接続を保存します。

CLI を使用する場合

hammer compute-resource create コマンドで接続を作成します。**--provider** で **Vmware** を選択し、**--uuid** でデータセンターのインスタンス UUID を設定します。

```
# hammer compute-resource create --name "ACME's vSphere" \
--provider "Vmware" \
--description "vSphere server at vsphere.example.com" \
--server "vsphere.example.com" --user "SatelliteUser" \
--password "p@55w0rd!" --locations "New York" --organizations "ACME" \
--uuid 72cb9454-81cd-4231-a863-d9baf0f399f8
```



注記

Satellite が TCP ポート 443 を介して vCenter と通信できるよう、ホストおよびネットワークベースのファイアウォールが設定されていることを確認します。Satellite が vCenter のホスト名を解決でき、vCenter が Satellite Server のホスト名を解決できることを確認します。

9.4. SATELLITE SERVER での VMWARE VSPHERE イメージの追加

VMware vSphere は、新規仮想マシンを作成するためのイメージとしてテンプレートを使用します。イメージベースのプロビジョニングを使用して新規ホストを作成する場合、VMware vSphere テンプレートの詳細を Satellite Server に追加する必要があります。これには、アクセスの詳細およびテンプレート名が含まれます。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、VMware vSphere 接続の名前をクリックします。UI には、**イメージ** タブを含む接続についての情報が表示されます。このタブには、新規プロバイダーのイメージは含まれませんが、新規イメージを追加することができます。**新規イメージ** をクリックすると、UI に VMware vSphere テンプレートの一連のフィールドが表示されます。

- **名前:** イメージのテキスト形式の名前。例:**`Test vSphere Image`**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **`RedHat 7.2`**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: **`x86_64`**
- **ユーザー名:** イメージにアクセスするための SSH ユーザー名。通常、これは**`root`** ユーザーになります。
- **ユーザーデータ** - イメージが **`cloud-init`** データなどのユーザーデータ入力をサポートするかどうかを設定します。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ:** vSphere 環境でのテンプレートの相対パスおよび名前 (例:**`Templates/RHEL72`**)。相対パスにデータセンターを組み込まないでください。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

`hammer compute-resource image create` コマンドでイメージを作成します。**`--uuid`** フィールドを使用して vSphere 環境の相対テンプレートパスを保存します。

```
# hammer compute-resource image create --name "Test vSphere Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" \
--username root --uuid "Templates/RHEL72" \
--compute-resource "ACME's vSphere"
```

9.5. VMWARE VSPHERE の詳細をコンピュートプロファイルへ追加

VMware vSphere の仮想マシンの特定のハードウェア設定を事前に定義することができます。これは、これらのハードウェア設定をコンピュートプロファイルに追加して実行できます。この例では、**`4-Example`** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動し、プロファイルの名前をクリックします。たとえば、事前に作成した **`4-Example`** プロファイルを使用します。UI には、コンピュートリソースの一覧が表示されます。vSphere 接続をクリックします。

UI には、プロファイルに VMware 固有の詳細を入力できる一連のフィールドがあります。これには以下が含まれます。

- **CPU:** 新規ホストに割り当てる CPU の数。
- **1 ソケットあたりのコア数:** 各 CPU に割り当てるコアの数。
- **メモリー:** 新規ホストに割り当てるメモリーの容量。
- **クラスター:** VMware 環境のターゲットホストクラスター。
- **リソースプール:** ホストの利用可能なリソース割り当てを含むリソースプール。
- **フォルダー:** ホストを整理するためのフォルダー。
- **ゲスト OS:** VMware vSphere で基礎となるオペレーティングシステムを定義します。
- **SCSI コントローラー:** ディスクアクセス方法を定義します。
- **仮想ハードウェアのバージョン:** 仮想マシンに使用する基礎となる VMware ハードウェアの抽象化を定義します。
- **メモリーホット追加** または **CPU ホット追加:** 仮想マシンの電源をオンにしたままリソースを追加できるかどうかを定義します。
- **イメージ:** イメージベースのプロビジョニングを実行している場合に使用するイメージ。この例では、**`Test VMware Image`** を使用します。
- **ネットワークインターフェース:** ホストのネットワークインターフェースのネットワークパラメーターを定義します。複数のネットワークインターフェースを作成することができます。ただし、少なくとも 1 つ以上のインターフェースが Capsule で管理されるネットワークをポイントしている必要があります。ネットワークインターフェースのオプションには、以下が含まれます。
  - **NIC タイプ:** VMware ネットワークインターフェースのタイプを定義します。
  - **ネットワーク:** 使用する仮想ネットワークを定義します。
- **ストレージ:** ホストのボリュームを定義します。ホストに複数のボリュームを作成することができます。ストレージのオプションには以下が含まれます。
  - **データストア:** ボリュームの保管場所を定義します。
  - **サイズ (GB):** ボリュームのサイズ (GB) を定義します。
  - **シンプロビジョニング:** シンプロビジョニングまたはフルディスクの事前割り当てを使用するかどうかを定義します。
  - **Eager Zero:** Eager Zero シックプロビジョニングを使用するかどうかを定義します。これにチェックを付けない場合、ディスクは Lazy Zero シックプロビジョニングを使用します。

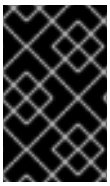
送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.3 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

9.6. VMWARE VSPHERE サーバーでのネットワークベースのホストの作成

VMware vSphere プロビジョニングプロセスは、ネットワーク接続を介して新規ホストを作成するオプションを提供します。この場合、新規ホストは Satellite Server の統合 Capsule または VMware vSphere 仮想ネットワークの外部 Capsule Server のいずれかにアクセスする必要があります。これは、ホストが PXE プロビジョニングサービスにアクセスできるようにするためです。



重要

プロビジョニングのために VMware vSphere サーバーで仮想ネットワークを使用している場合、DHCP 割り当てを提供しないネットワークを選択します。そうしないと、新規ホストの起動時に Satellite Server と DHCP との競合が生じるからです。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これは、プロビジョニングされたシステムのホスト名になります。この例では、**`vmware-test1`** になります。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **`ACME`** および **`New York`**
  - **ホストグループ** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。例: **`Base`**
  - **デプロイ先** で、VMware vSphere 接続を選択します。この例では、**`ACME's vSphere`** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **`4-Example`**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。VMware vSphere サーバーは MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
  - インターフェース画面には、コンピュートプロファイルの設定が入力された VMware vSphere 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- **オペレーティングシステム** タブ:

- すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
- **プロビジョニング方法** が **ネットワークベース** に設定されていることを確認します。
- **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。

- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method build** を組み込んでネットワークベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "vmware-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's vSphere" --provision-method build \
--build true --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=mynetwork" \
--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true" \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

新規ホストのエントリーは、VMware vSphere サーバーが仮想マシンを作成するようにトリガーします。仮想マシンが仮想ネットワーク経由で定義済みの Capsule Server を検出する場合、仮想マシンは PXE で起動し、選択したオペレーティングシステムのインストールを開始します。

9.7. VMWARE VSPHERE サーバーでのイメージベースのホストの作成

VMware vSphere プロビジョニングプロセスは、VMware vSphere サーバーで既存イメージから新規ホストを作成するオプションも提供します。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これは、プロビジョニングされたシステムのホスト名になります。この例では、**`vmware-test1`** になります。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** および **`New York`**
  - **ホストグループ** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。例: **`Base`**
  - **デプロイ先** で、VMware vSphere 接続を選択します。この例では、**`ACME's vSphere`** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **`4-Example`**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。VMware vSphere サーバーは MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
  - インターフェース画面には、コンピュートプロファイルの設定が入力された VMware vSphere 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **プロビジョニング方法** が **イメージベース** に設定されていることを確認します。新規の **イメージ** フィールドが表示されます。このフィールドから、新規ホストの root ボリュームのベースとしてイメージを選択できます。これは、**ホスト** タブで選択されたコンピュートプロファイルから自動的に設定されます。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:

- **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method image** を組み込んでイメージベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "vmware-test2" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's RHEV" --provision-method image \
--image "Test RHEV Image" --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=mynetwork" \
--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true" \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

この新規ホストのエントリーは、VMware vSphere サーバーが新規ボリュームのベースとして既存イメージを使用し、仮想マシンを作成するようトリガーします。

9.8. コンピュートリソースのキャッシング

コンピュートリソースのキャッシングは、VMware 情報のレンダリングを迅速化します。

9.8.1. コンピュートリソースのキャッシングの有効化

コンピュートリソースのキャッシングの有効化または無効化

1. **インフラストラクチャー> コンピュートリソース** に移動します。
2. 更新したい VMware サーバーの右側にある **編集** ボタンをクリックします。
3. **キャッシングの有効化**のチェックボックスを選択します。

9.8.2. コンピュートリソースのキャッシュのリフレッシュ

コンピュートリソースのキャッシュをリフレッシュして、コンピュートリソース情報を更新するには、以下を実行します。

Web UI を使用する場合

1. **インフラストラクチャー> コンピュートリソース** に移動します。

2. コンピュートリソースのキャッシュをリフレッシュしたい VMware サーバーを選択し、**キャッシュのリフレッシュ** ボタンをクリックします。

CLI を使用する場合

この API の呼び出しを使用して、コンピュートリソースのキャッシュをリフレッシュします。

```
# curl -H "Accept:application/json,version=2" \
-H "Content-Type:application/json" -X PUT \
-u username:password -k \
https://satellite.example.com/api/compute_resources/compute_resource_id/refresh_cache
```

コンピュートリソースのキャッシュをリフレッシュしたい VMware サーバーの id を決定します。

9.9. 本章のまとめ

本章では、Red Hat Satellite 6 を設定して VMware vSphere サーバーを使用する方法と、VMware vSphere サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。ここでは、ネットワークベースのホストおよびイメージベースのホストの両方について紹介しました。

Red Hat Satellite 6 で設定が必要なコンピュートリソースが他にない場合は、13章 *プロビジョニングの最終設定* のプロビジョニングについての最終メモを参照してください。

次章では、Red Hat OpenStack Platform 環境からのプロビジョニング方法について説明します。

第10章 RED HAT OPENSTACK PLATFORM でのクラウドインスタンスのプロビジョニング

Red Hat OpenStack Platform は、Red Hat Enterprise Linux をベースとして、プライベートまたはパブリックの Infrastructure-as-a-Service (IaaS) クラウドを構築するための基盤を提供します。これにより、スケーラビリティーが極めて高く、耐障害性に優れたプラットフォームをクラウド対応のワークロード開発にご利用いただくことができます。Red Hat Satellite 6 は、Red Hat OpenStack Platforms REST API と対話し、新規クラウドインスタンスを作成できます。また、それらの電源管理の状態を制御することができます。本章では、接続を ACME の Red Hat OpenStack Platform 環境に追加し、クラウドインスタンスをプロビジョニングします。

10.1. RED HAT OPENSTACK PLATFORM のプロビジョニングの要件定義

Red Hat OpenStack Platform のプロビジョニングの要件には、以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- OpenStack 環境でネットワークを管理する Capsule Server。詳細は、*4章ネットワークの設定* を参照してください。
- イメージベースのプロビジョニング用に OpenStack Image Storage (glance) サービスに追加されたイメージ。詳細は、『Red Hat OpenStack Platform インスタンスおよびイメージガイド』 を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。

10.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM 接続の追加

このプロセスでは、Satellite Server のコンピュートリソースで Red Hat OpenStack Platform 接続を追加します。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、**新規のコンピュートリソース** をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前:** リソースのテキスト形式の名前。例:**`ACME's OpenStack`**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**`RHEL OpenStack Platform`** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**`ACME's OpenStack environment at openstack.example.com`**
- **URL:** **`tokens`** リソースの OpenStack 認証 (keystone) サービスの API をポイントする URL。例: **`http://openstack.example.com:5000/v2.0/tokens`**
- **ユーザー名** および **パスワード**：Satellite から環境にアクセスするために使用する認証ユーザーおよびパスワード。
- **テナント** - Satellite Server が管理するテナントまたはプロジェクト。

- **外部ネットワークを主要ネットワークとして許可 (Allow external network as main network):** これを選択して外部ネットワークをホストのプライマリーネットワークとして使用できるようにします。

ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして Red Hat OpenStack Platform の接続を保存します。

CLI を使用する場合

hammer compute-resource create コマンドで接続を作成します。

```
# hammer compute-resource create --name "ACME's OpenStack" \
--provider "OpenStack" \
--description "ACME's OpenStack environment at openstack.example.com" \
--url "http://openstack.example.com:5000/v2.0/tokens" --user "admin" \
--password "p@55w0rd!" --tenant "openstack" --locations "New York" \
--organizations "ACME"
```

10.3. SATELLITE SERVER での RED HAT OPENSTACK PLATFORM イメージの追加

Red Hat OpenStack Platform はイメージベースのプロビジョニングを使用して新規ホストを作成します。つまり、イメージの詳細を Satellite Server に追加する必要があり、これにはアクセスの詳細およびイメージの場所が含まれます。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、 Red Hat OpenStack Platform 接続の名前をクリックします。UI には、**イメージ** タブを含む接続についての情報が表示されます。このタブには、新規プロバイダーのイメージは含まれませんが、新規イメージを追加することができます。**新規イメージ** をクリックすると、UI に Red Hat OpenStack Platform イメージの一連のフィールドが表示されます。

- **名前:** イメージのテキスト形式の名前。例:**Test OpenStack Image**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **RedHat 7.2**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: **x86_64**
- **ユーザー名:** イメージにアクセスするための SSH ユーザー名。通常、これは**root** ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ:** OpenStack Image Storage のイメージ。
- **ユーザーデータ** - イメージが **cloud-init** データなどのユーザーデータ入力をサポートするかどうかを設定します。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

hammer compute-resource image create コマンドでイメージを作成します。**--uuid** フィールドを使用して Red Hat OpenStack Platform サーバーのイメージの場所の完全パスを保存します。

```
# hammer compute-resource image create --name "Test OpenStack Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" \
--username root --user-data true \
--compute-resource "ACME's OpenStack Platform"
```

10.4. RED HAT OPENSTACK PLATFORM の詳細をコンピュートプロファイルへ追加

Red Hat OpenStack Platform のインスタンスの特定のハードウェア設定を事前に定義することができます。これらのハードウェア設定をコンピュートプロファイルに追加して実行できます。この例では、**4-Example** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動し、プロファイルの名前をクリックします。たとえば、事前に作成した **4-Example** プロファイルを使用します。UI には、コンピュートリソースの一覧が表示されます。OpenStack Platform 接続をクリックします。

UI には、プロファイルに OpenStack 固有の詳細を入力できる一連のフィールドがあります。これには以下が含まれます。

- **フレーバー:** ホストに使用する OpenStack Platform のハードウェアのプロファイル。
- **利用可能ゾーン (Availability zone):** OpenStack Platform 環境内で使用するターゲットクラスター。
- **イメージ:** イメージベースのプロビジョニングに使用するイメージ。この例では、**Test OpenStack Image** を使用します。
- **テナント:** OpenStack Platform インスタンスのテナントまたはプロジェクト。
- **セキュリティーグループ** - ポートおよび IP アドレスのクラウドベースのアクセスルール。
- **内部ネットワーク** - ホストが加わるプライベートネットワーク。
- **Floating IP ネットワーク** - ホストが加わる外部ネットワークで、Floating IP アドレスを割り当てます。
- **ボリュームからの起動** - イメージからボリュームが作成されるかを設定します。これが選択されていない場合、インスタンスはイメージを直接起動します。
- **新規起動ボリュームサイズ (GB)** - 新規起動ボリュームのサイズ (GB)。

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.3 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

10.5. RED HAT OPENSTACK PLATFORM でのイメージベースのホストの作成

Red Hat OpenStack Platform プロビジョニングプロセスでは、Red Hat OpenStack Platform サーバーで既存イメージから新規ホストを作成します。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**`openstack-test1`** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** および **`New York`**
  - **ホストグループ** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。例: **`Base`**
  - **デプロイ先** で、OpenStack Platform 接続を選択します。この例では、**`ACME's OpenStack Platform`** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、クラウドインスタンスベースの設定を自動的に行うために使用するプロファイルを選択します。例: **`4-Example`**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。Red Hat OpenStack Platform サーバーは MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **イメージ** フィールドには、コンピュートプロファイルから選択されたイメージが含まれます。このフィールドから、新規ホストの root ボリュームのベースとなる別のイメージを選択することもできます。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。

- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method image** を組み込んでイメージベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "openstack-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's OpenStack Platform" --provision-method image \
--image "Test OpenStack Image" --enabled true --managed true \
--interface "managed=true,primary=true,provision=true" \
--compute-
attributes="flavor_ref=m1.small,tenant_id=openstack,security_groups=defaul
t,network=mynetwork"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

この新規ホストのエントリーは、Red Hat OpenStack Platform サーバーが新規ボリュームのベースとして既存イメージを使用し、インスタンスを作成するようトリガーします。

10.6. 本章のまとめ

本章では、Red Hat Satellite 6 を設定して Red Hat OpenStack Platform サーバーを使用する方法と、Red Hat OpenStack Platform サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。ここでは、ネットワークベースのホストおよびイメージベースのホストの両方について紹介しました。

Red Hat Satellite 6 で設定が必要なコンピュートリソースが他にない場合は、13章 *プロビジョニングの最終設定* のプロビジョニングについての最終メモを参照してください。

次章では、 Amazon の EC2 パブリッククラウドサービスからのプロビジョニング方法について説明します。

第11章 AMAZON EC2 でのクラウドインスタンスのプロビジョニング

Amazon Elastic Compute Cloud (Amazon EC2) は、パブリッククラウドコンピュートリソースを提供する Web サービスです。Red Hat Satellite 6 は、Amazon EC2 のパブリック API を使用して新規クラウドインスタンスを作成し、それらの電源管理の状態を制御することができます。本章では、接続を ACME の Amazon EC2 アカウントに追加し、クラウドインスタンスをプロビジョニングします。

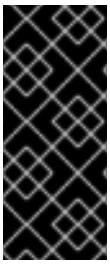
11.1. AMAZON EC2 プロビジョニングの要件定義

Amazon EC2 プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリー。詳細は、**『コンテンツ管理ガイド』** の 「Red Hat リポジトリーの同期」 を参照してください。
- EC2 環境でネットワークを管理する Capsule Server。ホストと Capsule Server 間のネットワークのセキュリティーを確保するために、Virtual Private Cloud (VPC) を使用するのが望ましいと言えます。
- イメージベースのプロビジョニングに選択された Amazon Machine Image (AMI)
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「アクティベーションキーの作成」 を参照してください。

11.2. SATELLITE SERVER への AMAZON EC2 接続の追加

このプロセスでは、Satellite Server のコンピュートリソースで Amazon EC2 接続を追加します。



重要

Amazon Web Services は、認証プロセスの一環として時間設定を使用します。これは、Satellite Server の時間が正常に同期される必要があることを意味します。**ntpd** または **chronyd** などの NTP サービスが Satellite Server で適切に実行されていることを確認します。Amazon Web Services に正確な時間を指定できないと、認証が失敗する可能性があります。詳細は、**『インストールガイド』** の 「時間の同期」 を参照してください。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、**新規のコンピュートリソース** をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前:** リソースのテキスト形式の名前。例:**`ACME's EC2`**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**EC2** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**`Amazon EC2 Public Cloud`**
- **アクセスキー** および **シークレットキー** - Amazon EC2 アカウントのアクセスキー。これらのキーは、Amazon EC2 管理コンソールの **セキュリティー認証情報** で生成します。詳細は、Amazon ドキュメントの Web サイト Managing Access Keys for your AWS Account を参照してください。

- **リージョン** - 使用する Amazon EC2 リージョン/データセンター。アクセスキーを入力したら、**リージョンのロード** をクリックして利用可能なリージョンを表示します。

ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして Amazon EC2 接続を保存します。

CLI を使用する場合

`hammer compute-resource create` コマンドで接続を作成します。**`--user`** および **`--password`** フィールドはそれぞれアクセスキーおよびシークレットキーとして機能します。以下は例になります。

```
# hammer compute-resource create --name "ACME's EC2" --provider "EC2" \
--description "Amazon EC2 Public Cloud` --user "ABCDEFGHIJ1234567" \
--password "*********" --region "us-east-1" --locations "New York" \
--organizations "ACME"
```

11.3. SATELLITE SERVER での AMAZON EC2 イメージの追加

Amazon EC2 はイメージベースのプロビジョニングを使用して新規ホストを作成します。つまり、イメージの詳細を Satellite Server に追加する必要があることを意味します。これにはアクセスの詳細およびイメージの場所が含まれます。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、Amazon EC2 接続の名前をクリックします。UI には、**イメージ** タブを含む接続についての情報が表示されます。このタブには、新規プロバイダーのイメージは含まれませんが、新規イメージを追加することができます。**新規イメージ** をクリックすると、UI に Amazon EC2 イメージの一連のフィールドが表示されます。

- **名前:** イメージのテキスト形式の名前。例:**`Test Amazon EC2 Image`**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **`RedHat 7.2`**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: **`x86_64`**
- **ユーザー名:** イメージにアクセスするための SSH ユーザー名。通常、これは**`root`** ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ ID:** イメージの Amazon Machine Image (AMI) ID。通常、この形式は **`ami-xxxxxxxx`** になります。例: **`ami-b32c14ad`**
- **ユーザーデータ** - イメージが **`cloud-init`** データなどのユーザーデータ入力をサポートするかどうかを設定します。ユーザーデータを有効にすると、フィニッシュスクリプトが無効になります。ユーザーデータを有効にすると、フィニッシュスクリプトは自動的に無効になります。これは、逆の場合にも当てはまります。フィニッシュスクリプトを有効にすると、ユーザーデータが無効になります。
- **IAM ロール:** イメージを作成するために使用される Amazon のセキュリティーロール。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

hammer compute-resource image create コマンドでイメージを作成します。**--uuid** フィールドを使用して Amazon EC2 サーバーのイメージの場所の完全パスを保存します。

```
# hammer compute-resource image create --name "Test Amazon EC2 Image" \
--operatingsystem "RedHat 7.2" --architecture "x86_64" --username root \
--user-data true --uuid "ami-b32c14ad" --compute-resource "ACME's EC2"
```

11.4. AMAZON EC2 の詳細をコンピュートプロファイルへ追加

Amazon EC2 のインスタンスの特定のハードウェア設定を事前に定義することができます。これは、これらのハードウェア設定をコンピュートプロファイルに追加して実行できます。この例では、**4-Example** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

インフラストラクチャー > コンピュートプロファイル に移動し、プロファイルの名前をクリックします。たとえば、事前に作成した **4-Example** プロファイルを使用します。UI には、コンピュートリソースの一覧が表示されます。EC2 接続をクリックします。

UI には、プロファイルに Amazon 固有の詳細を入力できる一連のフィールドがあります。これには以下が含まれます。

- **フレーバー:** ホストに使用する EC2 のハードウェアのプロファイル。
- **イメージ:** イメージベースのプロビジョニングに使用するイメージ。この例では、**Test EC2 Image** を使用します。
- **利用可能ゾーン (Availability zone):** EC2 リージョン内で使用するターゲットクラスター。
- **サブネット:** EC2 インスタンスのサブネット。新規ホストのプロビジョニング用の VPC がある場合は、そのサブネットを使用します。
- **セキュリティーグループ** - ポートおよび IP アドレスのクラウドベースのアクセスルール。ホストに適用するグループを選択します。
- **管理 IP** - IP アドレスの割り当てのタイプ。これは、**パブリック** IP または **プライベート** IP のいずれかになります。

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.3 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

11.5. AMAZON EC2 でのイメージベースのホストの作成

Amazon EC2 プロビジョニングプロセスでは、Amazon EC2 サーバーで既存イメージから新規ホストを作成します。

Web UI を使用する場合

ホスト > 新規ホスト に移動します。UI には、ホストの詳細を入力できる一覧のフィールドがあります。

- **ホスト** タブ:
  - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**ec2-test1** と入力します。
  - プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** および **New York**
  - **ホストグループ** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。例: **Base**
  - **デプロイ先** で、EC2 接続を選択します。この例では、**ACME's EC2** になります。仮想マシンの新規タブが表示されます。
  - **コンピュートプロファイル** で、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **4-Example**
- **インターフェース** タブ:
  - ホストのインターフェースの **編集** をクリックします。
  - ほとんどのフィールドには、値が自動的に含まれるはずです。特に以下の点に注意してください。
    - **ホスト** タブの **名前** は **DNS 名** になります。
    - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** を空白にします。Amazon EC2 サーバーは MAC アドレスをホストに割り当てます。
  - Satellite Server は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずです。選択されていない場合は、それらを選択してください。
- **オペレーティングシステム** タブ:
  - すべてのフィールドには値が自動的に含まれるはずです。オペレーティングシステムの各要素を確認してください。
  - **イメージ** フィールドには、コンピュートプロファイルから選択されたイメージが含まれます。このフィールドから、新規ホストの root ボリュームのベースとなる別のイメージを選択することもできます。
  - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシン** タブ:
  - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずです。必要に応じてこれらの設定を変更します。
- **パラメーター** タブ:
  - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

hammer host create コマンドでホストを作成し、**--provision-method image** を組み込んでイメージベースのプロビジョニングを使用します。以下は例になります。

```
# hammer host create --name "ec2-test1" --organization "ACME" \
--location "New York" --hostgroup "Base" \
--compute-resource "ACME's EC2" --provision-method image \
--image "Test Amazon EC2 Image" --enabled true --managed true \
--interface "managed=true,primary=true,provision=true,subnet_id=EC2" \
--compute-
attributes="flavor_id=m1.small,image_id=TestImage,availability_zones=us-
east-1a,security_group_ids=Default,managed_ip=Public"
```

注記

このコンピュートリソースの追加のホスト作成パラメーターについての詳細は、付録B *Hammer CLI の追加のホストパラメーター*を参照してください。

この新規ホストのエントリーは、Amazon EC2 サーバーが新規ボリュームのベースとして既存イメージを使用し、インスタンスを作成するようトリガーします。

11.6. SSH を使って AMAZON EC2 インスタンスに接続

SSH を使用して、Satellite Server から Amazon EC2 インスタンスにリモートで接続することができます。しかし、Red Hat Satellite を介してプロビジョニングするあらゆる Amazon Web Services EC2 インスタンスへ接続するには、Foreman データベースのコンピュートリソースに関連するプライベートキーに最初にアクセスし、このキーを使って認証する必要があります。

プライベートキーの場所を確認し、SSH を使用して Amazon EC2 サーバーへ接続するには、以下の手順を実行します。

1. Satellite Server ベースシステムで、コンピュートリソースリストの場所を確認するには、以下のコマンドを入力し、使用したいコンピュートリソースの ID を確認します。

```
# hammer compute-resource list
```

2. ユーザーを **postgres** ユーザーに切り替えます。

```
# su - postgres
```

3. **postgres** シェルを開始します。

```
$ psql
```

4. **postgres** ユーザーとして、Foreman データベースに接続します。

```
# postgres=# \c foreman
```

5. **compute_resource_id = 3** である **key_pairs** から、シークレットを選択します。

```
# select secret from key_pairs where compute_resource_id = 3; secret
```

6. **-----BEGIN RSA PRIVATE KEY-----** 以降、**-----END RSA PRIVATE KEY-----** までキーをコピーします。

7. **.pem** ファイルを作成し、ファイルにキーを貼り付けます。

```
# vim Keyname.pem
```

8. **.pem** ファイルへのアクセスを制限するよう確認します。

```
# chmod 600 Keyname.pem
```

9. Amazon EC2 インスタンスへ接続するには、以下のコマンドを入力します。

```
ssh -i Keyname.pem   ec2-user@example.aws.com
```

11.7. AMAZON WEB SERVICE EC2 環境向けフィニッシュテンプレートの設定

Amazon EC2 環境で Red Hat Enterprise Linux インスタンスをプロビジョニングする間、Red Hat Satellite のフィニッシュテンプレートを使用できます。

Amazon EC2 向けのフィニッシュテンプレートを設定するには、以下の手順を実行します。

1. Red Hat Satellite 6 Web UI で、**ホスト > プロビジョニングテンプレート** に移動します。

2. **プロビジョニングテンプレート** ページの検索フィールドに **Kickstart default finish** を入力し、**検索** をクリックします。

3. **Kickstart default finish** テンプレートで、**クローン** を選択します。

4. **名前** フィールドに、テンプレート向けに独自の名前を入力します。

5. テンプレートで、**subscription-manager register** コマンドおよび **yum** コマンド以外の root 権限が必要な各コマンドを **sudo** で指定します。または、以下の行を追加してテンプレート全体を sudo ユーザーとして実行します。

```
sudo -s << EOS
_Template_ _Body_
EOS
```

6. **関連付け** タブをクリックし、使用したい Red Hat Enterprise Linux オペレーティングシステムとテンプレートを関連付けします。

7. **ロケーション** タブをクリックして、ホストがある場所を追加します。

8. **組織** タブをクリックして、ホストが属する組織を追加します。

9. 必要なカスタマイズまたは変更を追加したら、**送信** をクリックしてテンプレート保存します。

10. **ホスト > オペレーティングシステム** に移動し、ホスト用に必要なオペレーティングシステムを選択します。

11. **テンプレート** タブをクリックし、**フィニッシュテンプレート** リストから、フィニッシュテンプレートを選択します。

12. **ホスト > ホストの作成** に移動し、作成したいホストの情報を入力します。
13. **パラメーター** タブをクリックし、**ホストパラメーター** に移動します。
14. **ホストパラメーター** で、**パラメーターの追加** ボタンを 3 回クリックし、新しいパラメーターフィールドを 3 つ追加します。以下の 3 つのパラメーターを追加します。
    a. **名前** フィールドで、**remote_execution_ssh_keys** を入力します。対応する **値** フィールドで、**cat /usr/share/foreman/.ssh/id_rsa_foreman_proxy.pub** の出力を入力します。
    b. **名前** フィールドで、**remote_execution_ssh_user** を入力します。対応する **値** フィールドで、**ec2-user** を入力します。
    c. **名前** フィールドで、**kt_activation_keys** を入力します。対応する **値** フィールドで、アクティベーションキーを入力します。
15. **送信** をクリックして変更を保存します。

11.8. AMAZON WEB SERVICES と SATELLITE に関する詳細情報

Amazon Web Services EC2 で Red Hat Gold Images を確認する方法の詳細は、How to Locate Red Hat Cloud Access Gold Images on AWS EC2 を参照してください。

Linux で Amazon Web Service Client をインストールして使用する方法の詳細は、Amazon Web Services ドキュメンテーションの Install the AWS Command Line Interface on Linux を参照してください。

Amazon Web Services における仮想マシンのインポートおよびエクスポートに関する詳細は、Amazon Web Services ドキュメンテーションの VM Import/Export を参照してください。

11.9. 本章のまとめ

本章では、Red Hat Satellite 6 を設定して Amazon EC2 サーバーを使用する方法と、Amazon EC2 サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。ここでは、ネットワークベースのホストおよびイメージベースのホストの両方について紹介しました。

Red Hat Satellite 6 で設定が必要なコンピュートリソースが他にない場合は、13章 *プロビジョニングの最終設定* のプロビジョニングについての最終メモを参照してください。

次章では、Red Hat Enterprise Linux Atomic Server でのコンテナーのプロビジョニング方法について説明します。

第12章 コンテナーのプロビジョニング

コンテナー化とは、複数の分離したユーザースペースのインスタンスを提供するためにオペレーティングシステムのカーネルを使用する仮想化メソッドです。Docker とは、Linux コンテナー内のアプリケーションのデプロイメントを自動化するオープンソースプロジェクトであり、アプリケーションとその実行時の依存関係をコンテナーにパッケージ化する機能を提供します。Linux コンテナーは、セキュリティーを強化しつつ、迅速なアプリケーションのデプロイメントを可能にし、テスト、メンテナンスおよびトラブルシューティングを単純化します。

Red Hat Enterprise Linux Atomic Host は、Linux コンテナーの実行のために最適化された安全かつ軽量で、フットプリントを最小限に抑えたオペレーティングシステムです。Red Hat Satellite 6 は、Red Hat Enterprise Linux Atomic Host および他の Docker ベースのサーバーに接続する機能を提供します。これには、イメージからの新規コンテナーの作成が含まれます。本章では、接続を ACME の Red Hat Enterprise Linux Atomic Host に追加し、コンテナーをプロビジョニングします。

12.1. コンテナープロビジョニングの要件定義

Red Hat Enterprise Linux Atomic Host のプロビジョニングの要件には、以下が含まれます。

- コンテナーレジストリーなどのイメージのソース。Red Hat Satellite 6 はコンテナーイメージの 3 つのソースを使用します。
  - Satellite Server のアプリケーションライフサイクルの一部である、同期済みの Docker 形式のコンテナーイメージ。
  - Docker ハブからのパブリックイメージ。
  - Red Hat のコンテナーイメージのレジストリーを含む他の外部レジストリー。これは、「Satellite Server への外部レジストリーの追加」で説明されています。

12.2. RED HAT ENTERPRISE LINUX ATOMIC HOST の設定

Atomic Host では Satellite に追加する前に一部の設定が必要になります。これには、Docker 向け Red Hat API の Satellite Server への公開が含まれます。

Atomic Host にログインし、**/etc/sysconfig/docker** ファイルを編集します。

```
$ vi /etc/sysconfig/docker
```

OPTIONS パラメーターを見つけ、API を公開するように編集します。

```
OPTIONS='--selinux-enabled -H unix:///var/run/docker.sock -H
tcp://0.0.0.0:2375'
```

重要

接続にポート 2375 または 2376 のいずれかを使用します。Satellite Server には、これらのポートへのアクセスを可能にする特殊な SELinux ルールが含まれるためです。他のポートを使用すると、認証が失敗します。

Satellite Server 証明書をインポートします。

```
$ curl http://satellite.example.com/pub/katello-server-ca.crt \
-o /etc/pki/ca-trust/source/anchors/katello-server-ca.crt
$ update-ca-trust
```

docker サービスを再起動します。

```
$ systemctl restart docker
```

ポートが公開されていること確認します。

```
$ netstat -tulnp | grep 2375
```

12.3. SATELLITE SERVER への ATOMIC HOST 接続の追加

このプロセスでは、Satellite Server のコンピュートリソースで Red Hat Enterprise Linux Atomic 接続を追加します。

Web UI を使用する場合

インフラストラクチャー > コンピュートリソース に移動し、**新規のコンピュートリソース** をクリックします。UI には、コンピュートリソースの一連のフィールドがあります。

- **名前:** リソースのテキスト形式の名前。例:**ACME's Atomic**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**Docker** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**ACME's Atomic Host at atomic.example.com**
- Atomic Host で Docker の Red Hat API をポイントする URL。例: **http://atomic.example.com:2375**
- **ユーザー名**、**パスワード**、**Email (電子メール)** - コンテナーレジストリーに関する認証の詳細。Satellite Server は、Atomic ホストがコンテナーレジストリーからイメージをダウンロードするようにこれらの詳細を使用します。パブリックイメージまたは Satellite Server が管理するイメージを使用する場合は、これらの詳細は必要ありません。

ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして Red Hat OpenStack Platform の接続を保存します。

CLI を使用する場合

hammer compute-resource create コマンドで接続を作成します。

```
# hammer compute-resource create --provider docker \
--name "ACME's Atomic" --url "http://atomic.example.com:2375" \
--organizations 'Default Organization' --locations 'Default Location'
```

12.4. SATELLITE SERVER への外部レジストリーの追加

『コンテンツ管理ガイド』では、Red Hat Satellite 6 でコンテンツビューを使って Docker 形式のコン

テナーイメージを同期し、それらを管理する方法を説明しています。ただし、外部レジストリーへのアクセスのみが必要で、コンテンツを同期する必要がないケースもあります。Red Hat Satellite 6 は外部コンテナーレジストリーを追加する機能を提供します。

Web UI を使用する場合

コンテナー > レジストリーに移動し、**新規レジストリー (New Registry)** をクリックします。UI には、新規レジストリーの一連のフィールドが表示されます。

- **名前:** レジストリーのテキスト形式の名前。例:**`Red Hat`**
- **URL:** レジストリーの場所。例:**`https://registry.access.redhat.com`**
- **説明:** レジストリーのテキスト形式の説明。例:**`Red Hat Docker Image Registry`**
- **ユーザー名** および **パスワード:** プライベートレジストリーの認証の詳細。

ロケーション および **組織** タブは現在のコンテキストに自動的に設定されます。他のコンテキストをこれらのタブに追加します。

送信 をクリックして外部レジストリーを保存します。

CLI を使用する場合

`hammer docker registry create` コマンドを使用してレジストリーを作成します。

```
# hammer docker registry create --name "Red Hat" \
--url "https://registry.access.redhat.com" \
--description "Red Hat Docker Image Registry"
```

12.5. SATELLITE SERVER でのコンテナーの作成

コンテナーのプロビジョニングプロセスは、標準のホスト作成プロセスとは異なります。コンテナーの作成は、**ホスト > 新規ホスト** メニューからではなく、**コンテナー > 新規コンテナー** オプションを使用します。

Web UI を使用する場合

コンテナー > 新規コンテナーに移動します。UI には、コンテナーの作成ウィザードが表示されます。

事前 (Preliminary)

このセクションでは、使用する Atomic Host とプロビジョニングコンテキストを定義します。

- コンテナーイメージのコンピュートリソースを選択します。例: **"ACME's Atomic"**
- プロビジョニングコンテキスト **(組織** および **ロケーション)** は現在のコンテキストに自動的に設定されます。例: **ACME** and **`New York`**

イメージ

このセクションでは、以下の 3 つの方法を含むイメージの選択方法が提供されます。

- **コンテンツビュー:** Satellite Server のアプリケーションライフサイクルからイメージを選択します。**ライフサイクル環境**、**コンテンツビュー**、**リポジトリー**、Docker **タグ** および Docker コンテンツを含む **Capsule** Server を選択します。

- **Docker ハブ:** Docker ハブでの Docker イメージの検索機能を提供します。**検索** キーワードを入力し、虫眼鏡アイコンをクリックすると、イメージの一覧が表示されます。イメージを選択し、イメージの **タグ** を選択します。

- **外部レジストリー -** 外部コンテナーレジストリーでの Docker 形式のコンテナーイメージの検索機能を提供します。**検索** キーワードを入力し、虫眼鏡アイコンをクリックすると、イメージの一覧が表示されます。イメージを選択し、イメージの **タグ** を選択します。

設定

このセクションでは、コンテナーの初期設定を提供します。

- **基本オプション:**
  - コンテナーの **名前** を入力します。
  - コンテナーで実行する **コマンド** を入力します。
  - **エントリーポイント** を入力します。デフォルトは **`/bin/sh -c`** です。
- **コンピュートオプション:**
  - 個別の CPU を割り当てる **CPU セット** を入力します。
  - コンテナー化されたタスクで利用できる CPU 時間のシェアを設定する **CPU シェア** を入力します。
  - コンテナーのメモリー使用の割り当てに使用する **メモリー** の容量を入力します。

環境

このセクションでは、コンテナーの実行時に使用する Atomic Host の設定を提供します。

- **環境変数:** 環境変数のセットを定義できます。例: **`LANG=en_US.UTF-8`**
- **公開されたポート (Exposed Ports):** コンテナーでポートを開きます。たとえば、ポート 22 のコンテナーへの SSH 通信を開きます。
- **DNS:** コンテナーの DNS サーバーを入力します。
- **Run? -** 作成後にコンテナーを実行するかどうかを選択します。
- **シェル:** TTY コンソールおよび標準ストリームを含むシェルのオプションを提供します (**`STDIN`**、**`STDOUT`**、および **`STDERR`**)。

ウィザードですべてのオプションを完了した後に、**送信** をクリックします。

CLI を使用する場合

以下は、**`hammer docker container create`** コマンドを使用した 3 つの例です。まず、コンテンツビューからコンテナーを作成します。

```
# hammer docker container create --compute-resource "ACME's Atomic" \
--repository-name "rhel7" --tag "latest" --name "docker-test1" \
--command "bash" --organizations "ACME" --locations "New York"
```

次に Docker ハブからプロビジョニングを実行します。

```
# hammer docker container create --compute-resource "ACME's Atomic" \
--repository-name "docker.io/fedora" --tag latest \
--name "docker-test2" --command bash --organizations "ACME" \
--locations "New York"
```

最後に、外部レジストリーからプロビジョニングを実行します。

```
# hammer docker container create --compute-resource "ACME's Atomic" \
--registry-id 1 --repository-name "rhel" --tag latest \
--name "docker-test3 --command bash --organizations "ACME" \
--locations "New York"
```

これにより、選択されたイメージから新規コンテナーが作成され、選択された **Red Hat Enterprise Linux Atomic Host** で実行されます。

12.6. 本章のまとめ

本章では、**Red Hat Enterprise Linux Atomic Host** を追加し、管理するよう **Red Hat Satellite 6** を設定する方法と **Atomic Host** でコンテナーをプロビジョニングする方法を説明しました。

ここまでで、本書のすべてのプロビジョニングシナリオを扱いました。13章 *プロビジョニングの最終設定* でプロビジョニングについての最終メモを参照してください。

第13章 プロビジョニングの最終設定

新規ホストのプロビジョニングは Red Hat Satellite 6 の機能において中核となる部分を構成します。本章では、本書で扱ったトピックをまとめ、プロビジョニングが他の Red Hat Satellite 6 機能にどのような影響を与えるかについて説明します。

13.1. シナリオ目標の完了

本書では、ACME という架空の企業における複数のシナリオで、以下を実行してきました。

Red Hat Satellite 6 でのプロビジョニングの設定

本書では、プロビジョニングを実行するために Red Hat Satellite 6 のリソースおよびサービスを設定する方法を説明しました。これには、インストールメディア、テンプレート、コンピュートリソースおよびネットワークが含まれます。さらに本書では、PXE ベースのプロビジョニングに DHCP、DNS、および TFTP サービスを使用できるよう Capsule Server を設定する方法を説明しました。

ベアメタルホストのプロビジョニング

本書では、無人プロビジョニング、Discovery ベースのプロビジョニング、および PXE を使用しないプロビジョニングなどの各種の方法でベアメタルホストをプロビジョニングする方法を説明しました。

仮想マシンのプロビジョニング

本書では、KVM サーバー、Red Hat Virtualization および VMware vSphere などの仮想化環境からプロビジョニングを実行する例を紹介しました。

クラウドインスタンスのプロビジョニング

本書では、パブリッククラウド (Amazon EC2) およびプライベートクラウド (Red Hat OpenStack Platform) からクラウドインスタンスをプロビジョニングする方法を説明しました。

コンテナーのプロビジョニング

本書では、Red Hat Enterprise Linux Atomic Host でコンテナーをプロビジョニングする方法について説明しました。

13.2. 他のアプリケーションの統合

Red Hat Satellite 6 は以下のアプリケーションでプロビジョニングプロセスを拡張します。

Puppet

各 Capsule Server (統合 Capsule を含む) は Puppet マスターとして機能します。Satellite Server は Puppet エージェントをそれぞれの新規ホストにインストールします。これにより、ホストでリソースおよびサービスを自動的に設定する方法が提供されます。ホストのプロビジョニングプロセスでは、Puppet クラスを **Puppet クラス** タブで追加できます。詳細は、『Puppet Guide』 を参照してください。

Red Hat CloudForms

Red Hat CloudForms は Red Hat Satellite 6 に接続して、特定のレベルのプロビジョニングおよびホスト管理を制御できます。詳細は、『Red Hat CloudForms Integration with Red Hat Satellite 6 Guide』 を参照してください。

付録A プロビジョニングのサンプル用の初期化スクリプト

『Red Hat Satellite 6 コンテンツ管理ガイド』の例に従っていない場合は、以下の初期化スクリプトを使用してプロビジョニングのサンプル用の環境を作成することができます。

スクリプトファイル (**sat6-content-init.sh**) を作成し、以下を組み込みます。

```
#!/bin/bash

MANIFEST=$1

# Import the content from Red Hat CDN
hammer organization create --name "ACME" --label "ACME" \
--description "Our example organization for managing content."
hammer subscription upload --file ~/$MANIFEST --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (RPMs)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (Kickstart)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Satellite Tools 6.3 (for RHEL 7 Server) (RPMs)" \
--basearch "x86_64" --product "Red Hat Enterprise Linux Server" \
--organization "ACME"
hammer product synchronize --name "Red Hat Enterprise Linux Server" \
--organization "ACME"

# Create our application life cycle
hammer lifecycle-environment create --name "Development" \
--description "Environment for ACME's Development Team" \
--prior "Library" --organization "ACME"
hammer lifecycle-environment create --name "Testing" \
--description "Environment for ACME's Quality Engineering Team" \
--prior "Development" --organization "ACME"
hammer lifecycle-environment create --name "Production" \
--description "Environment for ACME's Product Releases" \
--prior "Testing" --organization "ACME"

# Create and publish our Content View
hammer content-view create --name "Base" \
--description "Base operating system" \
--repositories "Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server,Red
Hat Satellite Tools 6.3 for RHEL 7 Server RPMs x86_64" \
--organization "ACME"
hammer content-view publish --name "Base" \
--description "Initial content view for our operating system" \
--organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 \
--to-lifecycle-environment "Development" --organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 \
--to-lifecycle-environment "Testing" --organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 \
--to-lifecycle-environment "Production" --organization "ACME"
```

- スクリプトに実行可能なパーミッションを設定します。

```
# chmod +x sat6-content-init.sh
```

Red Hat カスタマーポータルから、サブスクリプションマニフェストのコピーをダウンロードし、マニフェストでスクリプトを実行します。

```
# ./sat6-content-init.sh manifest_98f4290e-6c0b-4f37-ba79-3a3ec6e405ba.zip
```

これにより、本書でのプロビジョニングのサンプルに必要な **Red Hat** のコンテンツがインポートされます。

付録B HAMMER CLI の追加のホストパラメーター

本付録では、**hammer host create** コマンドの追加のパラメーターの情報を提供します。

B.1. 共通のインターフェースパラメーター

これらのパラメーターは、すべてのプロビジョニングタイプに対して **--interface** オプションと共に使用されます。

パラメーター	説明	
mac	インターフェースの MAC アドレス	
ip	インターフェースの IP アドレス	
type	インターフェースのタイプ。例: **interface**、**bmc**、または **bond**	
name	このインターフェースに関連付けられたホスト名	
subnet_id	Satellite Server のサブネット ID	
domain_id	Satellite Server のドメイン ID	
identifier	デバイス ID。例: **eth0**	
managed	管理インターフェースのブール値。**true** または **false** に設定します。	
primary	プライマリーインターフェースのブール値。管理ホストにはプライマリーインターフェースを設定する必要があります。**true** または **false** に設定します。	
provision	このインターフェースでプロビジョニングするかどうかについてのブール値。**true** または **false** に設定します。	
virtual	インターフェースが VLAN インターフェースかどうかについてのブール値。**true** または **false** に設定します。	

virtual が **true** の場合に以下のパラメーターを使用します。

パラメーター	説明	
tag	VLAN タグ。この属性はサブネット VLAN ID より優先されます。仮想インターフェースの場合のみ使用されます。	
attached_to	このインターフェースが属するインターフェースの ID。例: **eth1**	

type が **bond** の場合に以下のパラメーターを使用します。

パラメーター	説明	
mode	ボンディングモード。**balance-rr**、**active-backup**、**balance-xor**、**broadcast**、**802.3ad**、**balance-tlb**、**balance-alb** のいずれかになります。	
attached_devices	スレーブインターフェースの ID。例: **[eth1,eth2]**	
bond_options	追加のボンディングオプション	

type が **bmc** の場合に以下のパラメーターを使用します。

パラメーター	説明	
provider	BMC プロバイダー。**IPMI** のみがサポートされます。	
username	BMC デバイスのユーザー名	
password	BMC デバイスのパスワード	

B.2. EC2 パラメーター

--compute-attributes の利用可能なパラメーターです。

パラメーター	説明	
flavor_id	使用する EC2 フレーバー	
image_id	使用するイメージの AMI ID	

パラメーター	説明	
availability_zone	EC2 プロバイダーのリージョンを含む利用可能なゾーン	
security_group_ids	使用するセキュリティーグループの ID	
managed_ip	パブリックまたはプライベート IP の活用	

B.3. LIBVIRT パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明	
cpus	CPU 数	
memory	メモリー容量 (バイト)	
start	マシンを開始するブール値	

--interface の利用可能なキーです。

パラメーター	説明	
compute_type	**bridge** または **network** のいずれか	
compute_network / compute_bridge	ネットワークまたは物理インターフェースの名前	
compute_model	インターフェースモデル。**virtio**、**rtl8139**、**ne2k_pci**、**pcnet**、または **e1000** のいずれかになります。	

--volume の利用可能なキーです。

パラメーター	説明	
pool_name	ボリュームを保管するストレージプール	
capacity	ボリュームの容量。例: **10G**	

パラメーター	説明	
format_type	ディスクタイプ。**raw** または **qcow2** のいずれかになります。	

B.4. RED HAT OPENSTACK PLATFORM パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明	
flavor_ref	使用するフレーバー	
image_ref	使用するイメージ	
tenant_id	使用するテナント	
security_groups	使用するセキュリティーグループの一覧	
network	インスタンスを接続するためのネットワーク	

B.5. RED HAT VIRTUALIZATION パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明	
cluster	ホストを取得するためのクラスター ID	
template	使用するハードウェアプロファイル	
cores	使用する CPU コア数	
memory	メモリー容量 (バイト)	
start	マシンを開始するブール値	

--interface の利用可能なキーです。

パラメーター	説明	
compute_name	インターフェース名。例: `eth0`	
compute_network	使用するクラスター内のネットワーク	

`--volume` の利用可能なキーです。

パラメーター	説明	
size_gb	ボリュームサイズ (GB 単位)	
storage_domain	使用するストレージドメイン	
bootable	ボリュームをブート可能として設定できるブール値。ブート可能なボリュームは 1 つだけです。	

B.6. VMWARE INTERFACE パラメーター

`--compute-attributes` の利用可能なキーです。

パラメーター	説明	
cpus	ホストの CPU 数	
corespersocket	CPU ソケットあたりのコア数。ハードウェアの 10 未満のバージョンを使用するホストにのみ適用されます。	
memory_mb	メモリーサイズ (MB)	
cluster	ホストのクラスター ID	
path	ホストを整理するためのフォルダーへのパス	
guest_id	ゲスト OS ID	
scsi_controller_type	VMware コントローラーの ID	
hardware_version	VMware ハードウェアのバージョン ID	

パラメーター	説明	
start	マシンを開始するブール値	

--interface の利用可能なキーです。

パラメーター	説明	
compute_type	ネットワークアダプターのタイプ。**VirtualVmxnet**、**VirtualVmxnet2**、**VirtualVmxnet3**、**VirtualE1000**、**VirtualE1000e**、**VirtualPCNet32** のいずれかになります。	
compute_network	VMware ネットワーク ID	

--volume の利用可能なキーです。

パラメーター	説明	
datastore	データストア ID	
name	ボリュームの名前	
size_gb	サイズ (GB 単位)	
thin	シンプロビジョニングを可能にするブール値	
eager_zero	Eager Zero シックプロビジョニングを可能にするブール値	

付録C FIPS 準拠ホストのプロビジョニング

Red Hat Satellite 6 は、National Institute of Standards and Technology の 暗号モジュールのセキュリティー要件 (Security Requirements for Cryptographic Modules) 標準 (参照番号 FIPS 140-2、FIPS と呼ばれる) に準拠するプロビジョニングホストをサポートします。

Red Hat Satellite 6 は FIPS が有効なホストではサポートされません。

FIPS に準拠するホストのプロビジョニングを有効にするには、以下の変更を実行します。

- 関連するオペレーティングシステム、ロケーションおよび組織の特定
- FIPS プロビジョニングテンプレートの作成および有効化
- プロビジョニングのパスワードハッシュアルゴリズムの変更
- Puppet のメッセージダイジェストアルゴリズムの変更
- FIPS 有効化パラメーターの設定

これらの変更が完了したら、新規のプロビジョニングテンプレートが指定のオペレーティングシステム、ロケーションおよび組織に関連付けられます。ホストをそれらのオペレーティングシステム、ロケーション、および組織にプロビジョニングする際に、ホストには FIPS 準拠の設定が適用されます。これらの設定が正常に行われていることを確認するには、「FIPS モードの有効化の確認」のステップを実行します。

前提条件

- **『Hammer CLI ガイド』** の 「認証」 セクションにある設定ステップを実行します。これにより、毎回 Satellite ユーザー名およびパスワードを指定せずに Hammer コマンドを実行できます。

C.1. 関連するオペレーティングシステム、ロケーションおよび組織の特定

FIPS 準拠のテンプレートを Satellite で作成する前に、FIPS 準拠のホストをデプロイするロケーション、組織およびオペレーティングシステムを特定する必要があります。たとえば、Red Hat Enterprise Linux 7 ホストを FIPS 準拠のホストとしてデプロイする場合、テンプレートは Red Hat Enterprise Linux 7 にのみ関連付けます。

1. すべてのロケーションを一覧表示します。

   例

   ```
   $ hammer location list
   ---|-----------------
   ID | NAME
   ---|-----------------
   2  | Default Location
   ---|-----------------
   ```

   NAME 列にある、FIPS 準拠のホストをデプロイするロケーションの値をメモします。

2. すべての組織を一覧表示します。

   例

```
---|----------------------|----------------------|------------
ID | NAME                 | LABEL                | DESCRIPTION
---|----------------------|----------------------|------------
1  | Default Organization | Default_Organization |
2  | Sales                | Sales_Department     |
---|----------------------|----------------------|------------
```

NAME 列にある、FIPS 準拠のホストをデプロイする組織の値をメモします。

3. すべてのオペレーティングシステムを一覧表示します。

例

```
$ hammer os list
---|-----------------|-------------|-------
ID | TITLE           | RELEASE NAME | FAMILY
---|-----------------|-------------|-------
2  | RedHat 6.6      |             | Redhat
3  | RedHat 7.1      |             | Redhat
1  | RedHat 7.2      |             | Redhat
4  | RedHat 6.7      |             | Redhat
---|-----------------|-------------|-------
```

TITLE 列にある、FIPS 準拠のホストをデプロイするオペレーティングシステムの値をメモします。

C.2. FIPS プロビジョニングテンプレートの作成および有効化

FIPS プロビジョニングテンプレートは git リポジトリーで提供されます。この手順では、それらを Satellite 環境にインポートした後に、必要なオペレーティングシステム、ロケーションおよび組織に関連付けます。

1. Satellite Server では、FIPS が有効なテンプレートを含む git リポジトリーのクローンを作成してから、リポジトリーのディレクトリーに切り替えます。

```
$ git clone https://github.com/RedHatSatellite/satellite6-fips-
client
$ cd satellite6-fips-client
```

このリポジトリーには、以下の Embedded RuBy (ERB) テンプレートが含まれます。それらはテキスト形式のため、テンプレートに含まれる設定内容の詳細を表示し、確認することができます。

- **Kickstart_Default_PXELinux_FIPS.erb**
  - 更新済みの PXELinux テンプレート
- **fips_packages.erb**
  - FIPS モードで必要なパッケージ (例: **dracut-fips**)
- **Satellite_Kickstart_Default_FIPS.erb**
  - **fips_packages** スニペットを呼び出すための修正を含むキックスタートテンプレート

- **puppet.conf.erb**
  - 更新された (SHA256) メッセージダイジェストアルゴリズムを含む更新済みの **puppet.conf** 設定ファイル

2. **PXELinux FIPS** テンプレートを追加します。

```
$ hammer template create  --name "Kickstart Default PXELinux FIPS" \
  --file Kickstart_Default_PXELinux_FIPS.erb  \
  --locations LOCATIONS \
  --organizations ORGANIZATION \
  --operatingsystems OS \
  --type PXELinux
```

プレースホルダーの値 **LOCATIONS**、**ORGANIZATION**、および **OS** を 「関連するオペレーティングシステム、ロケーションおよび組織の特定」 でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create  --name "Kickstart Default PXELinux FIPS" \
  --file Kickstart_Default_PXELinux_FIPS.erb \
  --locations "Default Location" \
  --organizations "Default Organization","Sales" \
  --operatingsystems "RedHat 6.6","RedHat 7.1","RedHat 7.2","RedHat
6.7" \
  --type PXELinux
```

3. **Satellite Kickstart Default FIPS** テンプレートを追加します。

```
$ hammer template create  --name "Satellite Kickstart Default FIPS"
\
  --file Satellite_Kickstart_Default_FIPS.erb  \
  --locations LOCATIONS \
  --organizations ORGANIZATION \
  --operatingsystems OS \
  --type provision
```

プレースホルダーの値 **LOCATIONS**、**ORGANIZATION**、および **OS** を 「関連するオペレーティングシステム、ロケーションおよび組織の特定」 でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create  --name "Satellite Kickstart Default FIPS"
\
  --file Satellite_Kickstart_Default_FIPS.erb  \
  --locations "Default Location" \
  --organizations "Default Organization","Sales" \
```

```
  --operatingsystems "RedHat 6.6","RedHat 7.1","RedHat 7.2","RedHat
6.7" \
  --type provision
```

4. **FIPS Packages** スニペットを追加します。

```
$ hammer template create  --name "fips_packages" \
  --file fips_packages.erb \
  --locations LOCATIONS \
  --organizations ORGANIZATION \
  --type snippet
```

プレースホールダーの値 **LOCATIONS** および **ORGANIZATION** を 「関連するオペレーティングシステム、ロケーションおよび組織の特定」 でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合は、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create  --name "fips_packages" \
  --file fips_packages.erb \
  --locations "Default Location" \
  --organizations "Default Organization","Sales" \
  --type snippet
```

5. デフォルトの Puppet 設定スニペットを更新します。

```
$ hammer template update --name puppet.conf \
  --file puppet.conf.erb  \
  --type snippet
```

Config template created というメッセージは成功したことを示します。

6. 新規テンプレートを使用するようにオペレーティングシステムオブジェクトを更新します。新規の FIPS テンプレートが Satellite に追加され、必要なオペレーティングシステムの **デフォルト** テンプレートとして設定されます。

   a. **Satellite Kickstart Default FIPS** および **Kickstart Default PXELinux FIPS** テンプレートの ID を特定します。

   例

```
$ hammer template list
---|--------------------------------------|----------
ID | NAME                                 | TYPE
---|--------------------------------------|----------
41 | redhat_register                      | snippet
42 | saltstack_minion                     | snippet
53 | Kickstart Default PXELinux FIPS      | PXELinux
46 | Satellite Kickstart Default          | provision
48 | Satellite Kickstart Default Finish   | finish
54 | Satellite Kickstart Default FIPS     | provision
47 | Satellite Kickstart Default User Data | user_data
50 | subscription_manager_registration    | snippet
```

```
29 | UserData default                          | user_data
30 | WAIK default PXELinux                     | PXELinux
---|-----------------------------------------|----------
```

この例では、ID はそれぞれ 54 および 53 になります。これらの ID はインストールに固有の ID です。

b. FIPS テンプレートをデフォルトとして指定します。

```
$ hammer os set-default-template --config-template-id TEMPLATE \
--id OS
```

プレースホールダー **TEMPLATE** および **OS** を、FIPS テンプレートと先にメモしておいたオペレーティングシステムの ID に置き換えます。FIPS テンプレートとオペレーティングシステムのすべての組み合わせに対してこのコマンドを繰り返し実行します。コンマ区切りの値の一覧は使用できません。

この例では、FIPS テンプレートが前の例で ID 1 と特定された Red Hat Enterprise Linux 7.2 のデフォルトとして設定されています。

例

```
$ hammer os set-default-template --config-template-id 54 --id 1
$ hammer os set-default-template --config-template-id 53 --id 1
```

C.3. プロビジョニングのパスワードハッシュアルゴリズムの変更

プロビジョニングで使用されるパスワードのハッシュアルゴリズムを SHA256 に設定します。この設定は、FIPS 準拠としてデプロイする各オペレーティングシステムに適用される必要があります。

注記

これは Red Hat Satellilte 6 が Satellite 6.1 からアップグレードされた場合 ***のみ*** 必要になります。Satellite 6.3 はデフォルトで SHA256 を使用します。

1. オペレーティングシステム ID を特定します。

例

```
$ hammer os list
---|-----------------|--------------|-------
ID | TITLE           | RELEASE NAME | FAMILY
---|-----------------|--------------|-------
2  | RedHat 6.6      |              | Redhat
3  | RedHat 7.1      |              | Redhat
1  | RedHat 7.2      |              | Redhat
4  | RedHat 6.7      |              | Redhat
---|-----------------|--------------|-------
```

2. 各オペレーティングシステムのパスワードハッシュ値を更新します。

```
$ hammer os update --title OS \
  --password-hash SHA256
```

TITLE 列の一致する値を使用して、必要なオペレーティングシステムのそれぞれについてこのコマンドを繰り返し実行します。コンマ区切りの値の一覧は使用できません。

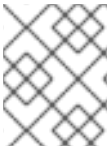
例

```
$ hammer os update --title "RedHat 7.2" \
  --password-hash SHA256
```

C.4. PUPPET の FIPS 準拠メッセージアルゴリズムへの切り替え

Satellite Server やすべての外部 Capsule Server および **すべての** 既存ホストで SHA256 メッセージダイジェストアルゴリズムを使用するように Puppet を設定します。

digest_algorithm = sha256 の行を **[main]** スタンザに追加して、**/etc/puppet/puppet.conf** ファイルを編集します。



注記

この変更は Satellite がアップグレードされるたびに上書きされるため、再度適用する必要があります。

Puppet メッセージダイジェストアルゴリズムは Satellite Server およびすべての Capsule Server で変更されるため、FIPS 準拠でないものも含め、**すべての** ホストで変更される必要があります。

メッセージダイジェストアルゴリズムに不一致がある場合、クライアントはファクトを再度ダウンロードします。これにより、Satellite Server または外部の Capsule Server での負荷が大幅に増大します。

C.5. FIPS 有効化パラメーターの設定

FIPS に準拠したホストをプロビジョニングするには、FIPS テンプレートで **fips_enabled** という名前のパラメーターを **true** に設定する必要があります。これが **true** に設定されていないか、またはこれがない場合は、FIPS 固有の変更は適用されません。このパラメーターは、個別のホストまたはホストグループセットのプロビジョニング時に指定できます。遡及的にホストで FIPS 準拠を有効にすることについては、本書の対象外であり、かつこれを実行すると問題が生じる可能性があります。

ホストのプロビジョニング時にこのパラメーターを設定するには、**--parameters fips_enabled=true** を Hammer コマンドに追加します。

既存のホストグループでこのパラメーターを設定するには、Hammer サブコマンドの **set-parameter** を使用します。詳細は、コマンド **hammer hostgroup set-parameter --help** の出力を参照してください。このホストグループにプロビジョニングされるすべてのホストはホストグループから **fips_enabled** パラメーターを継承します。

例

```
$ hammer hostgroup set-parameter --name fips_enabled \
 --value true \
 --hostgroup prod_servers
```

C.6. FIPS モードの有効化の確認

これらの FIPS 準拠に関する変更が正常に行われたことを確認するには、ホストのプロビジョニングを実行し、その設定を確認する必要があります。

1. FIPS テンプレートを使用してホストをデプロイし、**fips_enabled** という名前のパラメーターが **true** に設定されていることを確認します。

2. root と同等のアカウントで新規ホストにログインします。

3. コマンド **cat /proc/sys/crypto/fips_enabled** を実行します。値 **1** は FIPS モードが有効化されていることを示します。

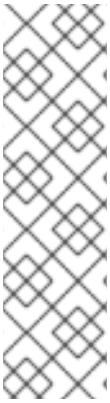
付録D RED HAT SATELLITE 向けクラウドイメージの構築

このセクションを使用して、Red Hat Satellite にイメージを構築して登録します。

事前に設定済みの Red Hat Enterprise Linux KVM ゲスト QCOW2 イメージを使用することができます。

- RHEL 7.4 KVM Guest Image
- RHEL 6.9 KVM Guest Image

これらのイメージには **cloud-init** が含まれます。適切に機能させるには、ec2 互換のメタデータサービスを使用して SSH キーをプロビジョニングする必要があります。



注記

KVM ゲストイメージでは、以下の点に注意してください。

- KVM ゲストイメージでは **root** アカウントが無効になっていますが、**cloud-user** という名前の特別なユーザーに **sudo** アクセスが許可されています。
- このイメージには **root** パスワードは設定されていません。

root パスワードは、**/etc/shadow** で 2 番目のフィールドに **!!** と記載することによりロックされます。

Red Hat Enterprise Linux のカスタムイメージを作成する場合は、Creating a Red Hat Enterprise Linux 7 Image および Creating a Red Hat Enterprise Linux 6 Image を参照してください。

D.1. RED HAT ENTERPRISE LINUX のカスタムイメージの作成

前提条件

- Linux ホストマシンを使用して、イメージを作成します。この例では、Red Hat Enterprise Linux 7 Workstation を使用します。
- ワークステーションで **virt-manager** を使用して、この手順を実行します。リモートサーバーでイメージを作成した場合、**virt-manager** を使用してワークステーションからサーバーに接続します。
- Red Hat Enterprise Linux 7 または 6 の ISO ファイル (Red Hat Enterprise Linux 7.4 Binary DVD または Red Hat Enterprise Linux 6.9 Binary DVD を参照)。

Red Hat Enterprise Linux Workstation のインストールに関する詳細は、『Red Hat Enterprise Linux 7 インストールガイド』 を参照してください。

カスタムのイメージを作成する前に、以下のパッケージをインストールします。

- **libvirt**、**qemu-kvm** およびグラフィカルツールをインストールします。

```
[root@host]# yum install virt-manager virt-viewer libvirt qemu-kvm
```

- 以下のコマンドラインツールをインストールします。

```
[root@host]# yum install virt-install libguestfs-tools-c
```

注記

以下の手順では、**libvirt** 環境をホストするワークステーションで、**[root@host]#** プロンプトを伴うコマンドはすべて入力します。

D.2. RED HAT ENTERPRISE LINUX 7 イメージの作成

このセクションを使用して、Red Hat Enterprise Linux 7 の ISO ファイルを使って QCOW2 形式のイメージを作成します。

1. Web ブラウザーを使って、Red Hat Enterprise Linux バイナリー ISO ファイルをテンポラリーの場所にダウンロードします。たとえば、**Downloads** ディレクトリーなどにダウンロードします。

2. Red Hat Enterprise Linux バイナリー ISO ファイルを **/var/lib/libvirt/images/** ディレクトリーにコピーします。

```
[root@host]# cp ~/home/user/Downloads/rhel-server-7.4-x86_64-dvd.iso
/var/lib/libvirt/images/
```

3. **virtbr0** が仮想ブリッジであることを確認します。

```
[root@host]# ip a
```

4. **libvirtd** を開始します。

```
[root@host]# systemctl start libvirtd
```

5. **/var/lib/libvirt/images/** ディレクトリーに移動します。

```
[root@host]# cd /var/lib/libvirt/images/
```

6. QEMU イメージの準備

```
[root@host]# qemu-img create -f qcow2 rhel7.qcow2 8G
```

7. **virt-install** を使用してインストールを開始します。以下の例はガイドとして使用します。

```
[root@host]# virt-install --virt-type qemu --name rhel7 --ram 2048 \
--cdrom rhel-server-7.4-x86_64-dvd.iso \
--disk rhel7.qcow2,format=qcow2 \
--network=bridge:virbr0 --graphics vnc,listen=0.0.0.0 \
--noautoconsole --os-type=linux --os-variant=rhel7
```

注記

GUI ユーザーは、インスタンスが自動的に起動しない場合には、**virt-manager** コマンドを入力して、コンソールを確認します。

```
[root@host]# virt-manager
```

8. Red Hat Enterprise Linux インストールウィザードの手順を実行します。

   a. インストールソースについては、Red Hat Satellite のリポジトリーに HTTP リンクを追加します。例: **satellite.example.com/pub/export/RHEL7/content/dist/rhel/server/7/7Server/x86_64/os/**

   b. インストールに使用するデバイスのタイプについては、**Auto-detected installation media** を選択します。

   c. インストール先のタイプについては、**Local Standard Disks** を選択します。

   d. その他のストレージオプションについては、**Automatically configure partitioning** を選択します。

   e. ソフトウェアの選択については、**Minimal Install** を選択します。

   f. ネットワークインターフェースを **ON** に設定し、システムの開始時にインターフェースが起動することを確認します。

   g. ホスト名を入力し、**Apply** をクリックします。

   h. **root** パスワードを入力します。

9. インストールの完了後、インスタンスを再起動して root ユーザーとしてログインします。

10. ネットワークインターフェースが起動し、IP アドレスが割り当てられていることを確認します。

```
# ip a
```

11. ホスト名が正しいことを確認します。

```
# hostname
```

12. **/etc/NetworkManager/conf.d/XX-cloud-image.conf** ファイルを作成します。このファイルの **XX** は、優先順位を示す 2 桁の番号になります。ファイルに以下のコンテンツを追加します。

```
[main]
dns=none
```

13. ホスト登録の設定 へ進みます。

D.3. RED HAT ENTERPRISE LINUX 6 イメージの作成

このセクションを使用して、Red Hat Enterprise Linux 6 の ISO ファイルを使って QCOW2 形式のイメージを作成します。

1. **virt-install** でインストールを開始します。

```
[root@host]# qemu-img create -f qcow2 rhel6.qcow2 4G
[root@host]# virt-install --connect=qemu:///system --
network=bridge:virbr0 \
--name=rhel6 --os-type linux --os-variant rhel6 \
```

```
--disk path=rhel6.qcow2,format=qcow2,size=10,cache=none \
--ram 4096 --vcpus=2 --check-cpu --accelerate \
--hvm --cdrom=rhel-server-6.8-x86_64-dvd.iso
```

このコマンドによりインスタンスが起動し、インストールプロセスが開始します。

注記

インスタンスが自動的に起動しない場合には、**virt-viewer** コマンドを入力して、コンソールを確認します。

```
[root@host]# virt-viewer rhel6
```

2. 仮想マシンを以下のように設定します。
   a. インストーラーの初期起動メニューで、**Install or upgrade an existing system** のオプションを選択します。
   b. 適切な **言語** および **キーボード** オプションを選択します。
   c. インストールに使用するデバイスタイプを尋ねるプロンプトが表示されたら、**基本ストレージデバイス** を選択します。
   d. デバイスの **ホスト名** を選択します。デフォルトのホスト名は **localhost.localdomain** です。
   e. root パスワードの設定
   f. ディスク上の空き容量に応じて、インストールタイプを選択します。
   g. **基本サーバー** のインストールを選択します。これには SSH サーバーが含まれます。
3. インスタンスを再起動して、**root** ユーザーとしてログインします。
4. **/etc/sysconfig/network-scripts/ifcfg-eth0** ファイルを編集して、以下の値のみが記載されている状態にします。

```
TYPE=Ethernet
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
NM_CONTROLLED=no
```

5. サービスネットワークを再起動します。

```
# service network restart
```

6. ホスト登録の設定 へ進みます。

D.4. ホスト登録の設定

Red Hat Enterprise Linux 仮想マシンは、デフォルトでカスタマーポータルの Subscription Management に登録されます。それぞれの仮想マシンの設定を更新して、適切な Satellite Server または Capsule Server から更新を受け取れるようにする必要があります。

前提条件

- ホストは以下の Red Hat Enterprise Linux バージョンを使用している必要があります。
  - 6.4 以上
  - 7.0 以上
- Red Hat Enterprise Linux のすべてのアーキテクチャーがサポートされます (i386、x86_64、s390x、ppc_64)。
- Satellite Server、任意の Capsule Server、およびホストで時刻同期ツールが有効になっており、実行していることを確認します。
  - Red Hat Enterprise Linux 6 の場合:

```
# chkconfig ntpd on; service ntpd start
```

  - Red Hat Enterprise Linux 7 の場合:

```
# systemctl enable chronyd; systemctl start chronyd
```

- デーモン **rhsmcertd** が有効になっており、ホストで実行されていることを確認します。
  - Red Hat Enterprise Linux 6 の場合:

```
# chkconfig rhsmcertd on; service rhsmcertd start
```

  - Red Hat Enterprise Linux 7 の場合:

```
# systemctl start rhsmcertd
```

ホストを登録するための設定:

1. Satellite Server または Capsule Server の完全修飾ドメイン名 (FQDN) をメモしておきます (例: **server.example.com**)。
2. ホストで、root ユーザーとしてターミナルに接続します。
3. ホストを登録する Satellite Server または Capsule Server からコンシューマー RPM をインストールします。コンシューマー RPM は、ホストのコンテンツソースのロケーションを更新し、ホストが、Red Hat Satellite に指定したコンテンツソースからコンテンツをダウンロードできるようにします。

```
# rpm -Uvh http://server.example.com/pub/katello-ca-consumer-
latest.noarch.rpm
```

D.5. ホストの登録

前提条件

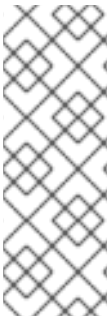
- ホストに対して、適切なコンテンツビューおよび環境に関連しているアクティベーションキーが存在していることを確認します。詳細は、『**コンテンツ管理ガイド**』の「アクティベーションキーの管理」を参照してください。アクティベーションキーでは、デフォルトで

auto-attach 機能が有効になっています。この機能は、一般的にハイパーバイザーとして使用されるホストで使用されます。

- **subscription-manager** ユーティリティーのバージョンが 1.10 以上であることを確認します。パッケージは標準の Red Hat Enterprise Linux リポジトリーで利用できます。

1. Red Hat Enterprise Linux Workstation で、ルートユーザーとしてターミナルに接続します。
2. Red Hat Subscription Manager を使ってホストを登録します。

```
# subscription-manager register --org="My_Organization" --
activationkey="MyKey"
```



注記

アクティベーションキーで定義したコンテンツビューとライフサイクル環境を上書きするには、**--environment** オプションを使用します。たとえば、「開発」ライフサイクル環境のコンテンツビュー「MyView」にホストを登録するには、以下を実行します。

```
 # subscription-manager register --org --environment
Development/MyView --activationkey
```

注記

Red Hat Enterprise Linux 6.3 ホストの場合、リリースバージョンはデフォルトで Red Hat Enterprise Linux 6 Server になり、これが 6.3 リポジトリーに設定されている必要があります。

1. Red Hat Satellite で、**ホスト > コンテンツホスト** を選択します。
2. 変更が必要なホストの名前を選択します。
3. **コンテンツホストのコンテンツ**セクションで、**リリースバージョン** の右側にある編集アイコンをクリックします。
4. **リリースバージョン** のドロップダウンメニューで、**"6.3"** を選択します。
5. **保存** をクリックします。

D.6. KATELLO エージェントのインストール

以下の手順を使用して、Satellite 6 に登録されているホストに Katello エージェントをインストールします。**katello-agent** パッケージは、**goferd service** を提供する gofer パッケージによって異なります。Red Hat Satellite Server または Capsule Server が、コンテンツホストに適用できるエラータに関する情報を提供できるように、このサービスを有効にしておく必要があります。

前提条件

Satellite Tools リポジトリーは必要なパッケージを提供するため、これを有効にし、Red Hat Satellite Server に同期させてホストで利用できるようにする必要があります。Satellite Tools の有効化に関する詳細は、**『ホストの管理』** の 「Katello エージェントのインストール」 を参照してください。

Katello エージェントをインストールするには、以下を実行します

Katello エージェントをインストールするには、以下を実行します。

1. 以下のコマンドを使って **katello-agent** RPM パッケージをインストールします。

```
# yum install katello-agent
```

2. goferd が実行していることを確認します。

```
# systemctl start goferd
```

D.7. PUPPET エージェントのインストール

本セクションを使用し、ホストに Puppet エージェントをインストールして設定します。 Puppet エージェントを適切にインストールして設定している場合は、**ホスト > すべてのホスト** に移動して、Red Hat Satellite Server に表示されるすべてのホストの一覧を表示します。

1. 以下のコマンドを使用して Puppet エージェントの RPM パッケージをインストールします。

```
# yum install puppet
```

2. 起動時に開始する Puppet エージェントを設定します。
Red Hat Enterprise Linux 6 の場合:

```
# chkconfig puppet on
```

Red Hat Enterprise Linux 7 の場合:

```
# systemctl enable puppet
```

D.8. RED HAT ENTERPRISE LINUX 7 イメージの完了

1. システムを更新します。

```
# yum update
```

2. **cloud-init** パッケージをインストールします。

```
# yum install cloud-utils-growpart cloud-init
```

3. **/etc/cloud/cloud.cfg** 設定ファイルを開きます。

```
# vi /etc/cloud/cloud.cfg
```

4. **cloud_init_modules** のタイトルの下に、以下を追加します。

```
 - resolv-conf
```

resolv-conf オプションは、インスタンスの初回起動時に **resolv.conf** を自動的に設定します。このファイルには、**nameservers**、**domain**、その他のオプションなどのインスタンスに関連した情報が記載されています。

5. **/etc/sysconfig/network** ファイルを開きます。

```
# vi /etc/sysconfig/network
```

6. 以下の行を追加し、EC2 メタデータサービスへのアクセスで問題が発生するのを回避します。

```
NOZEROCONF=yes
```

7. 仮想マシンの登録を解除して、作成されるイメージをベースにクローン作成されるインスタンスすべてに同じサブスクリプション情報が含まれないようにします。

```
# subscription-manager repos --disable=*
# subscription-manager unregister
```

8. インスタンスの電源をオフにします。

```
# poweroff
```

9. Red Hat Enterprise Linux Workstation で、ルートユーザーとしてターミナルに接続し、**/var/lib/libvirt/images/** ディレクトリーに移動します。

```
[root@host]# cd /var/lib/libvirt/images/
```

10. **virt-sysprep** コマンドでイメージのリセットおよびクリーニングをして、問題なくインスタンスの作成に使用できるようにします。

```
[root@host]# virt-sysprep -d rhel7
```

11. **virt-sparsify** コマンドを使用してイメージのサイズを縮小します。このコマンドにより、ディスクイメージ内の空き容量は、ホスト内の空き容量に戻ります。

```
[root@host]# virt-sparsify --compress rhel7.qcow2 rhel7-cloud.qcow2
```

これにより、コマンドを実行する場所に新しい **rhel7-cloud.qcow2** ファイルが作成されます。

D.9. RED HAT ENTERPRISE LINUX 6 イメージの完了

1. システムを更新します。

```
# yum update
```

2. **cloud-init** パッケージをインストールします。

```
# yum install cloud-utils-growpart cloud-init
```

3. **/etc/cloud/cloud.cfg** 設定ファイルを編集して、**cloud_init_modules** の下に以下を追加します。

```
- resolv-conf
```

resolv-conf オプションは、インスタンスの初回起動時に**resolv.conf** 設定ファイルを自動的に設定します。このファイルには、**nameservers**、**domain**、その他のオプションなどのインスタンスに関連した情報が記載されています。

4. ネットワークの問題が発生するのを防ぐために、以下のように **/etc/udev/rules.d/75-persistent-net-generator.rules** ファイルを作成します。

```
# echo "#" > /etc/udev/rules.d/75-persistent-net-generator.rules
```

これにより、**/etc/udev/rules.d/70-persistent-net.rules** ファイルが作成されるのを防ぎます。**/etc/udev/rules.d/70-persistent-net.rules** が作成されてしまうと、スナップショットからのブート時にネットワークが適切に機能しなくなる可能性があります (ネットワークインターフェースが「eth0」ではなく「eth1」として作成され、IP アドレスが割り当てられません)。

5. **/etc/sysconfig/network** に以下の行を追加し、EC2 メタデータサービスへのアクセスで問題が発生するのを回避します。

```
NOZEROCONF=yes
```

6. 仮想マシンの登録を解除して、作成されるイメージをベースにクローン作成されるインスタンスすべてに同じサブスクリプション情報が含まれないようにします。

```
# subscription-manager repos --disable=*
# subscription-manager unregister
# yum clean all
```

7. インスタンスの電源をオフにします。

```
# poweroff
```

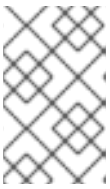
8. Red Hat Enterprise Linux Workstation でルートとしてログインし、**virt-sysprep** コマンドを使用してイメージのリセットとクリーニングをし、問題なくインスタンスの作成に使用できるようにします。

```
[root@host]# virt-sysprep -d rhel6
```

9. **virt-sparsify** コマンドを使用してイメージのサイズを縮小します。このコマンドにより、ディスクイメージ内の空き容量は、ホスト内の空き容量に戻ります。

```
[root@host]# virt-sparsify --compress rhel6.qcow2 rhel6-cloud.qcow2
```

これにより、コマンドを実行する場所に新しい **rhel6-cloud.qcow2** ファイルが作成されます。



注記

インスタンスに適用されているフレーバーのディスクスペースに応じて、イメージをベースとするインスタンスのパーティションを手動でリサイズする必要があります。

D.10. 次のステップ

- Satellite とプロビジョニングしたいすべてのイメージで、この手順を繰り返します。
- 後で使うために保管したい場所にイメージを移動します。