



# Red Hat Satellite 6.5

## プロビジョニングガイド

Red Hat Satellite Server での物理ホストと仮想ホストのプロビジョニングガイド



# Red Hat Satellite 6.5 プロビジョニングガイド

---

Red Hat Satellite Server での物理ホストと仮想ホストのプロビジョニングガイド

Red Hat Satellite Documentation Team

[satellite-doc-list@redhat.com](mailto:satellite-doc-list@redhat.com)

## 法律上の通知

Copyright © 2019 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

## 概要

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解説します。本書には、必要なネットワークポロジのセットアップや、必要なサービスの設定、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。

## 目次

<b>第1章 はじめに</b> .....	<b>5</b>
1.1. プロビジョニングタイプの概要	5
1.2. プロビジョニングワークフローの概要	5
<b>第2章 プロビジョニングコンテキストの設定</b> .....	<b>7</b>
2.1. プロビジョニングコンテキストの概要	7
2.2. プロビジョニングコンテキストの設定	7
<b>第3章 プロビジョニングリソースの設定</b> .....	<b>8</b>
3.1. オペレーティングシステムの作成	8
3.2. アーキテクチャーの作成	9
3.3. ハードウェアモデルの作成	9
3.4. ホストのオペレーティングシステム用の同期済みキックスタートリポジトリの使用	10
3.5. パーティションテーブルの作成	10
3.6. プロビジョニングテンプレートの作成	11
3.6.1. プロビジョニングのタイプ	11
3.6.2. テンプレート構文と管理	13
3.6.3. 手順	13
3.7. コンピュートプロファイルの作成	14
3.8. ホストへのデフォルト暗号化 ROOT パスワードの設定	15
3.9. サードパーティーのインストールメディア使用	15
3.10. NOVNC を使用した仮想マシンへのアクセス	16
<b>第4章 ネットワークの設定</b> .....	<b>18</b>
4.1. イメージベースのプロビジョニングの要件	18
4.2. ネットワークサービスの設定	19
4.2.1. ネットワーク設定の DHCP、DNS および TFTP オプション	20
4.2.2. NAT 経由での TFTP サービスの使用	21
4.3. SATELLITE SERVER へのドメインの追加	22
4.4. SATELLITE SERVER へのサブネットの追加	23
4.5. プロビジョニング時間を削減するための IPXE の設定	24
4.5.1. 仮想マシンのチェーンブート	24
4.5.2. iPXE ディレクトリーのチェーンブート	26
<b>第5章 DHCP および DNS プロバイダーとしての INFOBLOX の使用</b> .....	<b>30</b>
5.1. 制限事項	30
5.2. 前提条件	30
5.3. CAPSULE SERVER への INFOBLOX の CA 証明書のインストール	30
5.4. DHCP INFOBLOX モジュールのインストール	31
5.5. DNS INFOBLOX モジュールのインストール	32
<b>第6章 ベアメタルホストのプロビジョニング</b> .....	<b>33</b>
6.1. ベアメタルプロビジョニングの前提条件	33
6.2. セキュリティートークンの有効期間の設定	34
6.3. 無人プロビジョニングによるホストの作成	34
6.4. RED HAT SATELLITE の DISCOVERY サービスの設定	35
6.4.1. Capsule Server での Discovery サービスの有効化	36
6.4.2. テンプレート PXELinux Discovery スニペットのプロビジョニング	37
6.4.3. テンプレートとスニペットの変更	37
6.4.4. 検出されたホストの自動コンテキスト	39
6.5. 検出されたホストからのホストの作成	39
6.6. DISCOVERY ルールの作成	40

6.7. PXE を使用しないプロビジョニングによるホストの作成	42
6.8. PXE を使用しない DISCOVERY の実装	44
6.9. プロビジョニング中の SSH キーのデプロイ	47
6.10. SATELLITE DISCOVERY イメージの構築	48
<b>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</b>	<b>51</b>
7.1. KVM プロビジョニングの要件	51
7.2. SATELLITE SERVER の KVM 接続設定	52
7.3. SATELLITE SERVER への KVM 接続の追加	52
7.4. SATELLITE SERVER での KVM イメージの追加	53
7.5. KVM の詳細のコンピュートプロファイルへの追加	54
7.6. KVM サーバーでのホストの作成	55
<b>第8章 RED HAT VIRTUALIZATION での仮想マシンのプロビジョニング</b>	<b>58</b>
8.1. RED HAT VIRTUALIZATION プロビジョニングの要件	58
8.2. RED HAT VIRTUALIZATION ユーザーの作成	58
8.3. SATELLITE SERVER への RED HAT VIRTUALIZATION 接続の追加	59
8.4. SATELLITE SERVER への RED HAT VIRTUALIZATION イメージの追加	61
8.5. RED HAT VIRTUALIZATION の詳細のコンピュートプロファイルへの追加	62
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	62
<b>第9章 VMWARE VSPHERE での仮想マシンのプロビジョニング</b>	<b>65</b>
9.1. VMWARE VSPHERE プロビジョニングの要件	65
9.2. VMWARE VSPHERE ユーザーの作成	65
9.3. SATELLITE SERVER への VMWARE VSPHERE 接続の追加	65
9.4. SATELLITE SERVER への VMWARE VSPHERE イメージの追加	67
9.5. VMWARE VSPHERE の詳細のコンピュートプロファイルへの追加	67
9.6. VMWARE VSPHERE サーバーでのホストの作成	68
9.7. コンピュートリソースのキャッシング	71
9.7.1. コンピュートリソースのキャッシングの有効化	71
9.7.2. コンピュートリソースのキャッシュのリフレッシュ	71
<b>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</b>	<b>72</b>
10.1. RED HAT OPENSTACK PLATFORM プロビジョニングの要件	72
10.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM 接続の追加	72
10.3. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM イメージの追加	73
10.4. RED HAT OPENSTACK PLATFORM の詳細のコンピュートプロファイルへの追加	74
10.5. RED HAT OPENSTACK PLATFORM でのイメージベースのホストの作成	75
<b>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</b>	<b>77</b>
11.1. AMAZON EC2 プロビジョニングの要件	77
11.2. SATELLITE SERVER への AMAZON EC2 接続の追加	77
11.3. コンピュートリソースと HTTP SMART PROXY との連携	78
11.4. SATELLITE SERVER への AMAZON EC2 イメージの追加	79
11.5. AMAZON EC2 の詳細のコンピュートプロファイルへの追加	79
11.6. AMAZON EC2 でのイメージベースのホストの作成	80
11.7. SSH での AMAZON EC2 インスタンスへの接続	81
11.8. AMAZON WEB SERVICE EC2 環境向け FINISH テンプレートの設定	82
11.9. AMAZON WEB SERVICES と SATELLITE に関する詳細情報	83
<b>第12章 コンテナのプロビジョニング</b>	<b>84</b>
12.1. コンテナプロビジョニングの要件	84
12.2. コンテナイメージのインポート	84
12.3. SATELLITE SERVER への外部レジストリーの追加	86
12.4. コンテナ名のパターンの管理	86

---

12.5. SATELLITE でのコンテナイメージの管理	87
12.6. RED HAT ENTERPRISE LINUX ATOMIC HOST の設定	88
12.7. SATELLITE SERVER への ATOMIC HOST 接続の追加	89
12.8. コンテナの作成	90
12.9. コンテナレジストリーの認証管理	92
12.10. コンテナの起動、コミット、削除	92
<b>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</b>	<b>94</b>
<b>付録B HAMMER CLI の追加のホストパラメーター</b>	<b>96</b>
B.1. 共通のインターフェースパラメーター	96
B.2. EC2 パラメーター	97
B.3. LIBVIRT パラメーター	98
B.4. RED HAT OPENSTACK PLATFORM パラメーター	99
B.5. RED HAT VIRTUALIZATION パラメーター	99
B.6. VMWARE INTERFACE パラメーター	100
<b>付録C FIPS 準拠ホストのプロビジョニング</b>	<b>102</b>
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	102
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	103
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	106
C.4. PUPPET の FIPS 準拠メッセージアルゴリズムへの切り替え	107
C.5. FIPS 有効化パラメーターの設定	107
C.6. FIPS モードの有効化の確認	107
<b>付録D RED HAT SATELLITE 向けクラウドイメージの構築</b>	<b>109</b>
D.1. RED HAT ENTERPRISE LINUX のカスタムイメージの作成	109
D.2. RED HAT ENTERPRISE LINUX 7 イメージの作成	110
D.3. RED HAT ENTERPRISE LINUX 6 イメージの作成	111
D.4. ホスト登録の設定	112
D.5. ホストの登録	113
D.6. KATELLO エージェントのインストール	114
D.7. PUPPET エージェントのインストール	115
D.8. RED HAT ENTERPRISE LINUX 7 イメージの完了	115
D.9. RED HAT ENTERPRISE LINUX 6 イメージの完了	116
D.10. 次のステップ	117





# 第1章 はじめに

プロビジョニングとは、ベアメタルの物理または仮想マシンから、オペレーティングシステムの全設定を行い、使用準備を整えるまでのプロセスを指します。Red Hat Satellite を使用して、多数のホストに対して細かく設定を指定したプロビジョニングを定義し、自動化できます。

プロビジョニングには多数の方法があります。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、ホストの作成、管理、更新など Red Hat Satellite 6 アプリケーションライフサイクルに含まれます。

## 1.1. プロビジョニングタイプの概要

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。

### ベアメタルプロビジョニング

Satellite は、主に PXE ブートおよび MAC アドレス特定により、ベアメタルホストをプロビジョニングします。ホストエントリを作成して、プロビジョニングする物理ホストの MAC アドレスを指定してください。また、空のホストを起動して、Satellite の Discovery サービスを使用することで、プロビジョニングの準備が整ったホストのプールを作成できます。また、PXE なしの方法で、ホストを起動してプロビジョニングすることもできます。

### クラウドプロバイダー

Satellite は、プライベートおよびパブリッククラウドプロバイダーに接続して、クラウド環境に保存されているイメージからホストのインスタンスをプロビジョニングします。これには、使用するハードウェアプロファイルまたはフレーバーの選択も含まれます。

### 仮想化インフラストラクチャー

Satellite 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

Satellite では、Red Hat Enterprise Linux Atomic Server でコンテナを作成し、管理できます。

## 1.2. プロビジョニングワークフローの概要

プロビジョニングプロセスは、基本的なワークフローに従います。

1. ホストを作成します。Satellite は、サブネットに関連付けられている DHCP Capsule Server から未使用の IP アドレスを要求し、この IP アドレスを「ホスト作成」ウィンドウの **IP アドレス** のフィールドにロードします。新規ホストの全オプションを入力したら、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を予約します。
3. Satellite 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 に作成されません。
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 の **next-server**、**filename** オプションを返します。
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 からプロビジョニングテンプレートまたはスクリプトを要求します。
10. Satellite 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 に登録して管理ツール (例: **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 に保存されるモジュールを使用して設定を行います。

ワークフローはカスタムのオプションにより異なります。以下に例を示します。

## Discovery

Discovery サービスを使用している場合、Satellite 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 Capsule 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 なしのプロビジョニング

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。

## コンピュータリソース

コンピュータリソースはホストの仮想マシンを作成し、MAC アドレスを Satellite 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には、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を使用せず、コンピュータリソースで、使用する新規ホストのイメージのコピーを作成します。

## コンテナ

コンテナ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。

## 第2章 プロビジョニングコンテキストの設定

Satellite を使用してホストをプロビジョニングする前に、Satellite のホストの配置ストラテジーを理解しておく必要があります。これは、プロビジョニングコンテキストとして知られています。

### 2.1. プロビジョニングコンテキストの概要

プロビジョニングコンテキストは、Satellite コンポーネントに指定する組織やロケーションの組み合わせのことです。コンポーネントが所属する組織やロケーションを使用して、対象のコンポーネントに対する所有権やアクセス権を設定します。

組織は、所有者、目的、コンテンツ、セキュリティレベルその他の区分に基づいて Red Hat Satellite 6 コンポーネント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、コンポーネント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コンポーネント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。

ロケーションの機能は、組織によく似ています。相違点は、ロケーションは物理的または地理的な設定をもとにしている点です。ユーザーは、階層にロケーションをネスト化することができます。

ロケーションに関する詳細情報は、『[コンテンツ管理ガイド](#)』の「[ロケーションの管理](#)」を参照してください。

### 2.2. プロビジョニングコンテキストの設定

コンテキストの設定時に、プロビジョニングホスト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メニューは、Satellite Web UI の左上のメニューバーにあります。使用する組織またはロケーションを選択していない場合には、メニューには **任意の組織** および **任意のロケーション** と表示されます。

#### 手順

- プロビジョニングコンテキストを設定するには、**任意の組織** および **任意のロケーション** ボタンをクリックして、使用する組織およびロケーションを選択します。

各ユーザーはアカウント設定でデフォルトのプロビジョニングコンテキストを設定できます。Satellite Web UI の右上のユーザー名をクリックし、**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 をご利用の場合

CLI を使用する場合は、オプションとして、**--organization** または **--organization-label** と **--location** または **--location-id** を追加します。以下に例を示します。

```
# hammer host list --organization "Default_Organization" --location "Default_Location"
```

このコマンドは、Default\_Organization および Default\_Location に割り当てられたホストを出力します。

## 第3章 プロビジョニングリソースの設定

Red Hat Satellite 6 には、ホスト作成に使用可能なプロビジョニングリソースが含まれています。

### 3.1. 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たものです。

Red Hat の CDN からオペレーティングシステムをインポートすると、**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。

以下の手順を使用して、カスタムのオペレーティングシステムも追加できます。

#### 手順

1. Satellite Web UI で **ホスト > オペレーティングシステム** に移動して、**新規オペレーティングシステム** をクリックします。
2. **名前** フィールドには、オペレーティングシステムエントリーの名前を入力します。
3. **メジャー** フィールドには、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数字を入力します。
4. **マイナー** フィールドには、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数字を入力します。
5. **説明** フィールドには、オペレーティングシステムの説明を加えます。
6. **ファミリー** リストから、オペレーティングシステムのファミリーを選択します。
7. **root パスワードのハッシュ** リストから、Root パスワードのエンコード方法を選択します。
8. **アーキテクチャー** リストから、オペレーティングシステムがシステムが使用するアーキテクチャーを選択します。
9. **パーティションテーブル** タブをクリックして、対象のオペレーティングシステムに適用するパーティションテーブルを選択します。
10. オプション: Red Hat 以外のコンテンツを使用する場合には、**インストールメディア** タブをクリックして、対象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。詳しい情報は、[「サードパーティーのインストールメディア使用」](#)を参照してください。
11. **テンプレート** タブをクリックして、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 **finish テンプレート** を選択します。プロビジョニングに iPXE を使用予定の場合には、**iPXE テンプレート** など、他のテンプレートを選択してください。
12. **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 をご利用の場合

**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"
```

## 3.2. 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、基本的な i386 と x86\_64 のアーキテクチャーが事前設定されています。

以下の手順を使用して Satellite のアーキテクチャーを作成します。

### 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をサポートするのは 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。

### 手順

アーキテクチャーを作成するには、以下の手順を行います。

1. Satellite Web UI で **ホスト > アーキテクチャー** に移動して、アーキテクチャーウィンドウで、**アーキテクチャーの作成** をクリックします。
2. **名前** フィールドに、アーキテクチャーの名前を入力します。
3. **オペレーティングシステム** リストから、オペレーティングシステムを選択します。利用できるオペレーティングシステムがない場合には、作成して **ホスト > オペレーティングシステム** の下に割り当てることができます。
4. **送信** をクリックします。

### CLI をご利用の場合

**hammer architecture create** コマンドを入力して、アーキテクチャーを作成します。このアーキテクチャーに含める名前と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"
```

## 3.3. ハードウェアモデルの作成

以下の手順を使用して、Satellite でハードウェアを作成し、ホストが使用するハードウェアモデルを指定できるようにします。

### 手順

ハードウェアモデルを作成するには、以下の手順を行います。

1. Satellite Web UI で **ホスト > ハードウェアモデル** に移動して、ハードウェアモデルウィンドウで、**モデルの作成** をクリックします。
2. **名前** フィールドには、ハードウェアモデルの名前を入力します。

3. オプションで、**ハードウェアモデル** と **ベンダークラス** フィールドに、お使いのシステムに適した情報を入力します。
4. **情報** フィールドには、ハードウェアモデルの説明を入力します。
5. **送信** をクリックしてハードウェアモデルを保存します。

## CLI をご利用の場合

**hammer model create** コマンドを使用して、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** オプション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"
```

## 3.4. ホストのオペレーティングシステム用の同期済みキックスタートリポジトリの使用

Satellite には、プロビジョニングされたホストのオペレーティングシステムのインストールに使用する、同期済みのキックスタートリポジトリが含まれています。

キックスタートリポジトリを設定するには、以下の手順を実行します。

1. 既存のコンテンツビューに、使用を希望する同期済みのキックスタートリポジトリを追加するか、コンテンツビューを新たに作成してキックスタートリポジトリを追加します。
2. キックスタートリポジトリを追加した新しいバージョンのコンテンツビューを公開し、必要なライフサイクル環境にプロモートします。詳しい情報は、『[コンテンツ管理ガイド](#)』の「[コンテンツビューの管理](#)」を参照してください。
3. ホストの作成時に、**オペレーティングシステム** タブの **メディアの選択** で **同期済みのコンテンツ** チェックボックスを選択します。

### キックスタートツリーの表示

キックスタートツリーを表示するには、以下のコマンドを入力します。

```
# hammer medium list --organization "your_organization"
```

## 3.5. 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、任意のパーティションスキームの設定やパーティションテーブルのエントリー作成を行い、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。

### 手順

パーティションテーブルを作成するには、以下の手順を行います。

1. Satellite Web UI で **ホスト > パーティションテーブル** に移動して、パーティションテーブルウィンドウで、**パーティションテーブルの作成** をクリックします。

2. **名前** フィールドには、パーティションテーブルの名前を入力します。
3. 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するには、**デフォルト** のチェックボックスを選択します。
4. 他のパーティションテーブルで再利用可能なスニペットとして、テンプレートを特定するには、**スニペット** のチェックボックスを選択します。
5. **オペレーティングシステムの種類** リストから、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を選択します。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。
6. **テンプレートエディター** フィールドには、ディスクパーティションのレイアウトを入力します。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。

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。

7. **監査コメント** には、パーティションレイアウトへの変更の概要を追加します。
8. **組織** と **ロケーション** タブをクリックして、パーティションテーブルに関連付ける、他のプロビジョニングコンテキストを追加します。Sattelite により、現在のプロビジョニングコンテキストに、そのパーティションテーブルが追加されます。
9. **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 をご利用の場合

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 "My_Organization" \
--locations "My_Location"
```

## 3.6. 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。

### 3.6.1. プロビジョニングのタイプ

プロビジョニングテンプレートにはさまざまな種類があります。

#### 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

正しいカーネルオプションが指定されたインストーラーをホストで使用されるように、サブネットに関連付けられたテンプレート Capsule にデプロイする PXE ベースのテンプレート。BIOS のプロビジョニングの場合は、PXELinux テンプレートを、UEFI プロビジョニングの場合は、PXEGrub2 を選択します。

## Finish

主なプロビジョニングプロセスの完了時に使用するプロビジョニング後の設定スクリプト。これは、SSH タスクとして完了します。Finish テンプレートは、仮想環境でのイメージベースのプロビジョニングにのみ使用できます。Foreman Discovery ISO のイメージと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。このコンテキストのイメージは、デプロイメントの簡素化を目指した、仮想化環境でのインストールイメージを指します。

## Bootdisk

PXE 以外の起動方法に使用するテンプレート

## Kernel Execution (kexec)

PXE 以外の起動方法に使用するカーネル実行テンプレート



### 注記

カーネル実行は、テクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat サービスレベルアグリーメント (SLA) では完全にサポートされていません。これらは、機能的に完全でない可能性があり、実稼働環境での使用を目的とはしていませんが、近々発表予定のプロダクトイノベーションをリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

## 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。



### 3.6.2. テンプレート構文と管理

Red Hat Satellite には、多数のテンプレートサンプルが含まれています。Satellite Web UI で、**ホスト > プロビジョニングテンプレート** に移動して、サンプルを表示します。テンプレートを作成するか、テンプレートのクローンを作成して、編集できます。テンプレートに関するサポートが必要な場合には、**ホスト > プロビジョニングテンプレート > テンプレートの作成 > ヘルプ** に移動します。

Embedded Ruby (ERB) 構文を受け入れるテンプレート。詳細情報は、『**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。詳細情報は、『**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。

Satellite Server と Git リポジトリまたはローカルディレクトリの間でテンプレートを同期できます。詳細情報は、『**コンテンツ管理ガイド**』の「[付録 F. Git を使用したテンプレートの同期](#)」を参照してください。

#### 変更ログと履歴

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ー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。

### 3.6.3. 手順

テンプレートを作成するには、以下の手順を行います。

- 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、プロビジョニングテンプレートウィンドウで**テンプレートの作成** をクリックします。

**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。

または、テンプレートのクローンを作成して、そのクローンに更新を追加するには、以下の手順を行います。

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、使用するテンプレートを検索します。
2. 使用するテンプレートで、**クローン** をクリックして、テンプレートをもう1つ作成します。
3. **名前** フィールドには、プロビジョニングテンプレートの名前を入力します。
4. 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するには、**デフォルト** のチェックボックスを選択します。
5. **テンプレートエディター** フィールドには、プロビジョニングテンプレートのボディーを入力します。**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。

6. **監査コメント** には、監査を目的とするプロビジョニングテンプレートへの変更についての概要を入力します。
7. **タイプ** タブをクリックして、テンプレートがスニペットの場合には、**スニペット** のチェックボックスを選択します。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可能なプロビジョニングテンプレートに含まれています。
8. **タイプ** リストから、**プロビジョニングテンプレート** など、テンプレートの種類を選択します。
9. **関連付け** タブをクリックして、**適用可能なオペレーティングシステム** リストから、このプロビジョニングテンプレートに関連付けるオペレーティングシステム名を選択します。
10. 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と、プロビジョニングテンプレートに関連付けることができます。
11. **組織** および **ロケーション** タブをクリックして、テンプレートに別のコンテキストを追加します。
12. **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 をご利用の場合

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 "My_Organization" \
--locations "My_Location"
```

## 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タ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**

### 手順

1. Satellite Web UI で **インフラストラクチャー** > **コンピュートプロファイル** に移動して、コンピュートプロファイルのウィンドウで **コンピュートプロファイルの作成** をクリックします。
2. **名前** フィールドに、プロファイルの名前を入力して **送信** をクリックします。

### CLI をご利用の場合

Red Hat Satellite 6.5 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. 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。

### 3.9. サードパーティーのインストールメディア使用

インストールメディアは、Satellite Server がサードパーティーの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サードパーティー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には、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。

#### 手順

インストールメディアを作成するには、以下の手順を行います。

1. Satellite Web UI で **ホスト** > **インストールメディア** に移動して、**メディアの作成** をクリックします。
2. **名前** フィールドには、インストールメディアエントリーの名前を入力します。
3. **パス** には、インストールツリーを含む URL または NFS シェアを入力します。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
  - **\$version**: オペレーティングシステムのバージョン
  - **\$major**: オペレーティングシステムのメジャーバージョン
  - **\$minor**: オペレーティングシステムのマイナーバージョン
- HTTP パスの例:

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

NFS パスの例:

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

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。

4. **オペレーティングシステムの種類** リストから、メディアのディストリビューションまたはファミリーを選択します。たとえば、CentOS、および Fedora は、**Red Hat** ファミリーに属しません。
5. **組織** と **ロケーション** タブをクリックして、プロビジョニングコンテキストを変更します。Satellite Server により、設定されたプロビジョニングコンテキストにインストールメディアを追加します。
6. **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 をご利用の場合

**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/centos/$version/Server/$arch/os/' \  
--organizations "My_Organization" --locations "My_Location"
```

## 3.10. NOVNC を使用した仮想マシンへのアクセス

ブラウザを使用して、Satellite が作成した仮想マシンの VNC コンソールにアクセスできます。

Satellite は、以下の仮想化プラットフォームで noVNC の使用をサポートします。

- VMware
- Libvirt
- RHV

以下の手順に従い、仮想化プラットフォームおよびブラウザを設定して NoVNC コンソールを使用できるようにします。

### 前提条件

- Satellite で仮想マシンを作成しておく必要があります。
- 既存の仮想マシンの場合は、**コンピュートリソース** 設定の **ディスプレイのタイプ** が **VNC** になっていることを確認します。
- Satellite Server に Katello root CA 証明書をインポートする必要があります。ブラウザにセキュリティの例外を追加するだけでは、noVNC の使用の要件を満たしません。詳細情報は、『**Red Hat Satellite の管理**』ガイドの「[Katello Root CA 証明書のインストール](#)」のセクションを参照してください。

### 手順

1. 仮想マシンのホストシステムで、ポート 5900 から 5930 まで VNC サービスが許可され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. 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、コンピュートリソースの名前を選択します。
3. **仮想マシン** タブで、仮想マシンホ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。

## 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ネットワーク設定が必要です。本章を参照して、Satellite Server が統合された Capsule でネットワークサービスを設定します。

新規ホストが Capsule Server にアクセスできる必要があります。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、またはコンテンツと Capsule Server とを同期している場合には、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。

1. 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
2. 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。

以下の説明は、特定のネットワークを管理するスタンドアロンの Capsule Server の設定にも同様に当てはまります。Satellite が外部の DHCP、DNS および TFTP サービスを使用するように設定する方法は、『[オンラインネットワークからの Satellite Server のインストール](#)』の「[外部サービスの設定](#)」を参照してください。

### DHCP の範囲

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せんが、各サービスに同じサブネット内の別の範囲を使用することを推奨します。

## 4.1. イメージベースのプロビジョニングの要件

### 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、管理された DHCP サーバーが必要です。ホストは DHCP Capsule と関連付けられたサブネットで作成する必要があり、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。

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をトラブルシューティングする場合には、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ること。
- サブネットには、Satellite Server に割り当てられた DHCP Capsule があること。

- ホストには、Satellite Server に割り当てられた有効な IP アドレスがあること。
- 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すること。
- イメージから作成された仮想マシンは、SSH リクエストに応答すること。
- イメージから作成された仮想マシンは、SSH を介して、デプロイされたイメージと関連付けられている、ユーザーとパスワードを承認すること。
- Satellite Server で SSH キーを使用して仮想マシンにアクセスできること。これは、仮想マシンが Satellite Server から設定後のスクリプトを受信するために必要です。

## 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をトラブルシューティングする場合には、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 サーバーがあること。
- 仮想マシンには **cloud-init** サービスがインストールされ、有効化されていること。

仮想マシンイメージの **finish** および **finish** 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 サービスを使用します。たとえば、ネットワークで Capsule Server を DHCP サーバーとして機能させる必要がある場合があります。また、ネットワークで PXE ブートサービスを使用して、新規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ことも可能です。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 を使用できるように Capsule Server を設定する必要があります。

上記のオプションを指定して **satellite-installer** スクリプトを実行し、Satellite Server でこれらのサービスを設定します。

外部の 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。

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。

### 手順

Satellite の統合 Capsule でネットワークサービスを設定するには、以下の手順を行います。

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-managed 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 \  
--foreman-proxy-tftp-managed true
```

2. 設定する Capsule Server を検索します。

```
# hammer proxy list
```

3. Capsule Server の機能をリフレッシュして変更を表示します。

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

4. 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"
```

#### 4.2.1. ネットワーク設定の DHCP、DNS および TFTP オプション

##### DHCP オプション

###### --foreman-proxy-dhcp

DHCP サービスを有効にします。このオプションは、**true** または **false** 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** または **false** に設定します。

###### --foreman-proxy-dhcp-gateway

DHCP プールのゲートウェイ。これは、プライベートネットワークにあるホスト用の外部ゲートウェイのアドレスに指定します。

###### --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。これは、**eth1** に設定します。

###### --foreman-proxy-dhcp-nameservers

DHCP で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。これは、**eth1** の Satellite Server のアドレスに設定します。

###### --foreman-proxy-dhcp-range

Discovered および Unmanaged サービスのスペース区切りの DHCP プール範囲

###### --foreman-proxy-dhcp-server

管理する DHCP サーバーのアドレスを設定します。

##### DNS オプション



**--foreman-proxy-dns**

DNS サービスを有効にします。このオプションは、**true** または **false** に設定します。

**--foreman-proxy-dns-managed**

DNS 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** または **false** に設定します。

**--foreman-proxy-dns-forwarders**

DNS フォワーダーを設定します。これは、お使いの DNS サーバーに設定します。

**--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。これは、**eth1** に設定します。

**--foreman-proxy-dns-reverse**

DNS 逆引きゾーン名です。

**--foreman-proxy-dns-server**

管理する DNS サーバーのアドレスを設定します。

**--foreman-proxy-dns-zone**

DNS ゾーン名に設定します。

**TFTP オプション****--foreman-proxy-tftp**

TFTP サービスを有効にします。このオプションは、**true** または **false** に設定します。

**--foreman-proxy-tftp-managed**

TFTP 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** または **false** に設定します。

**--foreman-proxy-tftp-servername**

使用する TFTP サーバーを設定します。Capsule Server の IP アドレスを使用していることを確認してください。

**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。

**4.2.2. NAT 経由での TFTP サービスの使用**

NAT 経由で Satellite TFTP サービスを使用できます。これには、全 NAT ルートまたはファイアウォールで、UDP のポート番号 69 の TFTP サービスを有効にし、TFTP の状態追跡機能も有効にする必要があります。詳細情報は、お使いの NAT デバイスのドキュメントを参照してください。

NAT ルートまたはファイアウォールが Red Hat Enterprise Linux を使用する場合には、全デバイスで以下の手順を実行してください。

**Red Hat Enterprise Linux 7 の場合:**

以下のコマンドを使用して、UDP のポート番号 69 で TFTP サービスを許可し、カーネルの TFTP 状態追跡モジュールを読み込んで、変更を永続化します。

```
# firewall-cmd --add-service=tftp && firewall-cmd --runtime-to-permanent
```

**Red Hat Enterprise Linux 6 の場合:**

1. UDP のポート番号 69 で TFTP サービスを許可するように、ファイアウォールを設定します。

```
# iptables -A OUTPUT -i eth0 -p udp --sport 69 -m state \
--state ESTABLISHED -j ACCEPT
# service iptables save
```

2. **ip\_contrack\_tftp** カーネルの TFTP 状態モジュールを読み込みます。/etc/sysconfig/iptables-config ファイルで、**IPTABLES\_MODULES** の場所を特定して以下のように **ip\_contrack\_tftp** を追加します。

```
IPTABLES_MODULES="ip_contrack_tftp"
```

### 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。Satellite Server には、ドメイン名を割り当てる Capsule Server とドメインに関する情報が必要です。

#### 既存のドメインの確認

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。

#### 手順

Satellite にドメインを追加するには、以下の手順を行います。

1. Satellite Web UI で、**インフラストラクチャー > ドメイン** に移動して、**ドメインの作成** をクリックします。
2. **DNS ドメイン** フィールドに、完全な DNS ドメイン名を入力します。
3. **フルネーム** フィールドで、プレーンテキストのドメイン名を入力します。
4. **パラメーター** タブでは、任意のドメインレベルのパラメーターを設定し、このドメイン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す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。
5. **パラメーターの追加** をクリックし、**名前** および **値** フィールドに入力します。
6. **ロケーション** タブをクリックして、ドメインがある場所を追加します。
7. **組織** タブをクリックして、ドメイン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。

#### CLI をご利用の場合

**hammer domain create** コマンドを使用して、ドメインを作成します。

```
# hammer domain create --name "domain_name.com" \
--description "My example domain" --dns-id 1 \
--locations "My_Location" --organizations "My_Organization"
```

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 ID です。

## 4.4. SATELLITE SERVER へのサブネットの追加

Satellite は、新規ホストのインターフェースを設定するので、サブネットごとの情報を Satellite Server に追加する必要があります。インターフェースを設定するには、Satellite Server には、これらのインターフェースを接続するネットワークに関する全情報を含める必要があります。

### 手順

Satellite Server にサブネットを追加するには、以下の手順を行います。

1. Satellite Web UI で、**インフラストラクチャー** > **サブネット** に移動して、サブネットウィンドウで **サブネットの作成** をクリックします。
2. **名前** フィールドに、サブネットの名前を入力します。
3. **説明** フィールドには、サブネットの説明を入力します。
4. **ネットワークアドレス** フィールドには、サブネットのネットワークアドレスを入力します。
5. **ネットワークのプレフィックス** フィールドには、サブネットのネットワークプレフィックスを入力します。
6. **ネットワークマスク** フィールドには、サブネットのネットワークマスクを入力します。
7. **ゲートウェイアドレス** フィールドには、サブネットの外部ゲートウェイを入力します。
8. **プライマリー DNS サーバー** フィールドには、サブネットのプライマリー DNS を入力します。
9. **セカンダリー DNS サーバー** には、サブネットのセカンダリー DNS を入力します。
10. **IPAM** には、IP アドレス管理 (IPAM) に使用するメソッドを選択します。
  - **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 では IP アドレスの割り当てを管理し、内部データベースに IP アドレスを記録できます。
  - **なし**: IP アドレス管理がありません。
11. 選択した IPAM メソッドの情報を入力します。
12. **リモート実行** タブをクリックして、リモート実行を制御する Capsule を選択します。
13. **ドメイン** タブをクリックして、このサブネットに適用するドメインを選択します。
14. **Capsule** タブをクリックして、DHCP、TFTP、および逆引き DNS サービスなど、サブネットの各サービスに適用するカプセルを選択します。
15. **パラメーター** タブをクリックして、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。
16. **ロケーション** タブをクリックして、この Capsule を使用するロケーションを選択します。
17. **組織** タブをクリックして、この Capsule を使用する組織を選択します。
18. **送信** をクリックしてサブネットの情報を保存します。

## CLI をご利用の場合

以下のコマンドでサブネットを作成します。

```
# hammer subnet create --name "My_Network" \  
--description "your_description" \  
--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 "My_Location" --organizations "My_Organization"
```



### 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 6.5 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 6.5 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で仮想マシンをチェーンブートする方法
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する方法
3. UNDI を介して PXELinux を使用し、HTTP でカーネルとベアメタルホストの初期 RAM ディスクを移動する方法

### 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](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 テンプレート Capsule 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。

使用する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する場合は、テンプレートをクローンし、そのクローンを編集します。

#### 手順

Satellite が iPXE を使用するよう設定するには、以下の手順を行います。

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する場合は、**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. 使用する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、使用する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、使用する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

この手順を使用して、iPXE がネットワーク通信または UNDI インターフェースに内蔵のドライバーを使用するように設定します。Satellite Server と Capsule が iPXE を使用するように設定する方法は異なります。

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 テンプレート Capsule 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する場合は、テンプレートをクローンし、そのクローンを編集します。

### 手順

Satellite が UNDI ワークフローで iPXE を使用するように設定するには、以下の手順を行います。

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** または UNDI には **PXELinux chain iPXE UNDI** を入力し、**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する場合は、**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. 使用する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する場合は、**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. 使用する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、使用する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、使用する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、使用する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** を選択して、テンプレートが正しいことを確認します。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.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。

## 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。

### 手順

Capsule が iPXE をチェーンブートするように設定するには、以下の手順を行います。

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 root ディレクトリーにコピーします。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。

- 直接チェーンブートするには、以下のコマンドを入力します。

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

- UNDI の場合は、以下のコマンドを入力します。

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

3. ファイルのコンテンツを修正します。

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

## 第5章 DHCP および DNS プロバイダーとしての INFOBLOX の使用

Capsule Server を使用して Infoblox アプリケーションに接続し、DHCP および DNS レコードの作成や管理、IP アドレスの確保ができます。

NIOS 8.0 以降および Satellite 6.5 以降で、Infoblox バージョンはサポートされます。

### 5.1. 制限事項

DHCP および DNS レコードはすべて、単一のネットワークまたは DNS ビューでのみ管理できます。**satellite-installer** コマンドで、Capsule に Infoblox をインストールしてビューを設定した後にビューを編集できます。

Capsule Serverは、標準の HTTPS Web API を使用して単一の Infoblox ノードと通信します。クラスタリングと高可用性の設定をするには、Infoblox で設定してください。

Infoblox の TFTP 機能を使用した PXE 関連のファイルのホストはサポートされません。PXE プロビジョニングには、TFTP サービスとして Capsule を使用する必要があります。詳細情報は、[4章 ネットワークの設定](#)を参照してください。

### 5.2. 前提条件

DHCP と DNS エントリーを Satellite で管理するには、Infoblox アカウントの認証情報が必要です。

**DHCP Admin** および **DNS Admin** という名前で、Infoblox の管理ロールが設定されているようにします。

管理ロールにはパーミッションを割り当てるか、対象アカウントによる Infoblox API でのタスク実行を許可する管理グループに所属している必要があります。

### 5.3. CAPSULE SERVER への INFOBLOX の CA 証明書のインストール

Infoblox アプリケーションと統合する全 Capsule のベースシステムに、Infoblox HTTPS CA 証明書をインストールする必要があります。

Infoblox Web UI または以下の OpenSSL コマンドを使用して証明書をダウンロードできます。

```
# update-ca-trust enable
# openssl s_client -showcerts -connect Infoblox_hostname:443 </dev/null | \
openssl x509 -text >/etc/pki/ca-trust/source/anchors/infoblox.crt
# update-ca-trust extract
```

- **Infoblox\_hostname** エントリーは、X509 証明書の Infoblox アプリケーションのホスト名と一致する必要があります。

CA 証明書をテストするには、CURL クエリーを使用します。

```
# curl -u admin:infoblox https://Infoblox_hostname/wapi/v2.0/network
```

証明書に問題がない場合の応答例:

```
[
```

```
{
  "_ref":
  "network/ZG5zLm5ldHdvcmskMTkyLjE2OC4yMDluMC8yNC8w:Infoblox_hostname/24/default",
  "network": "192.168.202.0/24",
  "network_view": "default"
}
]
```

以下の Red Hat ナレッジベースアートを 사용하여証明書をインストールします (「[Red Hat Enterprise Linux 6 および 7 に CA 証明書をインストールする](#)」)。

## 5.4. DHCP INFOBLOX モジュールのインストール

Capsule に DHCP の Infoblox モジュールをインストールします。別のビューを使用したレコードの管理はできません。

この手順と、「[DNS Infoblox モジュールのインストール](#)」の手順を組み合わせ、DHCP と DNS Infoblox モジュールを同時にインストールすることも可能です。

### DHCP Infoblox のレコードタイプに関する考慮事項

**--foreman-proxy-plugin-dhcp-infoblox-record-type host** オプションのみを使用して DHCP および DNS モジュールを設定します。

**fixedaddress** のレコードタイプ設定で、DHCP および DNS Infoblox モジュールを設定すると、DNS の競合が発生するので、この設定はサポートされていません。 **--foreman-proxy-plugin-dhcp-infoblox-record-type** オプションを **fixedaddress** に設定して、Capsule Server に Infoblox モジュールをインストールするには、DNS Capsule と Reverse DNS Capsule の両方のオプションの設定を解除する必要があります。これは、Infoblox 自体が DNS を管理するためです。Satellite でホストの名前変更ができないなど、競合が発生させずに、**fixedaddress** オプションを使用できません。

### 手順

DHCP 向けの Infoblox モジュールをインストールするには、以下の手順を実行します。

1. Capsule で、以下のコマンドを実行します。

```
# satellite-installer --enable-foreman-proxy-plugin-dhcp-infoblox \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-managed false \
--foreman-proxy-dhcp-provider infoblox \
--foreman-proxy-plugin-dhcp-infoblox-record-type host \
--foreman-proxy-dhcp-server Infoblox_hostname \
--foreman-proxy-plugin-dhcp-infoblox-username admin \
--foreman-proxy-plugin-dhcp-infoblox-password infoblox \
--foreman-proxy-plugin-dhcp-infoblox-network-view default \
--foreman-proxy-plugin-dhcp-infoblox-dns-view default
```

2. Satellite Web UI で、**インフラストラクチャー > Capsule** に移動し、Infoblox DHCP モジュールを設定した Capsule を選択し、**更新** をクリックします。
3. **dhcp** 機能が表示されていることを確認します。
4. 全ドメインを Infoblox で管理する場合には、DNS Capsule が対象のドメインに設定されていることを確認してください。確認するには、Satellite Web UI で **インフラストラクチャー > ドメイン** に移動して、各ドメインの設定をチェックします。

5. 全サブネットを Infoblox で管理する場合には、DNS Capsule および Reverse DNS Capsule が設定されていることを確認してください。確認するには、Satellite Web UI で **インフラストラクチャー > サブネット** に移動して、各サブネットの設定をチェックします。

## 5.5. DNS INFOBLOX モジュールのインストール

以下の手順を使用して、Capsule に DNS Infoblox モジュールをインストールします。この手順と、「[DHCP Infoblox モジュールのインストール](#)」の手順を組み合わせると、DHCP と DNS Infoblox モジュールを同時にインストールすることも可能です。

DNS レコードがデフォルトの DNS ビューでのみ管理されている場合には、使用する DNS ビューを指定できません。

DHCP Infoblox モジュールをインストールするには、以下の手順を実行します。

1. Capsule で、以下のコマンドを実行します。

```
# satellite-installer --enable-foreman-proxy-plugin-dns-infoblox \  
--foreman-proxy-dns true \  
--foreman-proxy-dns-managed false \  
--foreman-proxy-dns-provider infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-server Infoblox_hostname \  
--foreman-proxy-plugin-dns-infoblox-username admin \  
--foreman-proxy-plugin-dns-infoblox-password infoblox
```

2. Satellite Web UI で、**インフラストラクチャー > Capsule** に移動し、Infoblox DNS モジュールを設定した Capsule を選択し、**更新** をクリックします。
3. **dns** 機能が表示されていることを確認します。

## 第6章 ベアメタルホストのプロビジョニング

Red Hat Satellite 6.5 では、ベアメタルインターフェースのプロビジョニングには主に 4 つの方法があります。

### 無人プロビジョニング

新規ホストは MAC アドレスで特定され、Satellite Server は PXE ブートプロセスを使用してホストをプロビジョニングします。

### Discovery を使用した無人プロビジョニング

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 なしのプロビジョニング

新規ホストは、ブートディスクまたは Satellite Server が生成する PXE なしの検出イメージでプロビジョニングされます。

### Discovery を使用した PXE なしのプロビジョニング

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。

## BIOS および UEFI サポート

Red Hat Satellite では、BIOS および UEFI ベースの PXELinux プロビジョニングの両方を実行できます。

BIOS および UEFI インターフェースはいずれも、コンピューターのオペレーティングシステムとファームウェアの間のインタープリターとして機能し、ブート時にハードウェアコンポーネントを初期化して、オペレーティングシステムを起動します。

BIOS は、初期化する次のアドレスを含むハードドライブの最初のセクションを読み取り、UEFI はファームウェアの代わりに、`.efi` ファイルに初期化および起動に関する情報をすべて保存します。UEFI システムは、比較的新しく、一般的になってきています。

UEFI で PXELinux プロビジョニングを実行するには、Intel x86\_64 が搭載されている Red Hat Enterprise Linux Server 7 以降を使用する必要があります。Satellite では、UEFI を使用した PXELinux プロビジョニングは、ベアメタルシステムでのみサポートされます。GRUB 関連の制約が原因で、UEFI を使用して完全なホストイメージでプロビジョニングできません。仮想マシンでは UEFI がサポートされておらず、UEFI SecureBoot もサポートされません。

Satellite プロビジョニングでは、DHCP ファイル名の PXE ロードャーは、PXE プロビジョニング時に TFTP を使用して読み込むファイルを定義します。BIOS システムの場合は、このファイルは `pxelinux.0` で、UEFI システムの場合はこのファイルは `grub2/grubx64.efi` です。

BIOS プロビジョニングでは、PXELinux テンプレートとオペレーティングシステムを関連付ける必要があります。

UEFI プロビジョニングでは、PXEGrub2 テンプレートとオペレーティングシステムを関連付ける必要があります。

PXELinux と PXEGrub2 のテンプレートの両方を関連付ける場合は、PXE ロードャー間で簡単に切り替えができるように、Satellite 6 で、両テンプレートの設定ファイルを TFTP にデプロイできます。

### 6.1. ベアメタルプロビジョニングの前提条件

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。

- 使用する Red Hat Enterprise Linux バージョンの同期済みのコンテンツリポジトリ。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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に、Satellite Server 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。
- ホスト登録用のアクティベーションキー。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作成](#)」を参照してください。
- 空のベアメタルホスト

無人および PXE なしのプロビジョニングのセキュリティトークンに関する情報は、「[セキュリティトークンの有効期間の設定](#)」を参照してください。

## 6.2. セキュリティトークンの有効期間の設定

無人および PXE なしのプロビジョニングを行う場合には、セキュリティ措置として、Satellite は自動的に一意のトークンを生成し、このトークンをキックスタートプロビジョニングプロセス時にダウンロードした ISO イメージの URL に追加します。

デフォルトでは、トークンの有効期限は 360 分です。ホストのプロビジョニング時に、この時間内にホストを再起動するようにしてください。トークンの有効期限が切れると、トークンが削除されて、404 エラーが送出されます。

トークンの有効期限を調節するには、Satellite Web UI で **管理** > **設定** に移動して、**プロビジョニング** タブをクリックします。トークンの**期間** オプションを検索して、編集アイコンをクリックし、期間を変更するか、**0** と入力してトークンの生成を無効にします。

## 6.3. 無人プロビジョニングによるホストの作成

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。

このホストのプロビジョニングの方法では、プロセス中の対話が最小限になっています。

### 手順

無人プロビジョニングでホストを作成するには、以下の手順を行います。

1. Satellite web UI で、**ホスト** > **ホストの作成** に移動します。
2. **名前** フィールドには、ホストの名前を入力します。
3. **組織** および **ロケーション** タブをクリックし、要件に合うようにコンテキストを変更します。
4. **ホストグループ** リストから、フォームの生成に使用するホストグループを選択します。
5. **インターフェース** タブをクリックして、ホストのインターフェースで、**編集** をクリックします。
6. フィールドに値が投入されていることを確認します。特に以下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。
7. **MAC アドレス** フィールドには、ホストの MAC アドレスを入力します。これにより、PXE ブートプロセス中のホストが識別されます。
  8. Satellite Server が、ホストの最初のインターフェースに **Managed (管理)**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、選択してください。
  9. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。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。
  10. **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。
  11. オプション: VLAN タグ付けを使用する場合には、VLAN ID を PXELinux または PXEGrub2 テンプレートに追加する必要があります。 **APPEND** の行に、 **vlanid=example\_vlanid** を追加します。  
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。
  12. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
  13. **送信** をクリックしてホストの詳細を保存します。

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。インストールが完了すると、ホストは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。

## CLI をご利用の場合

**hammer host create** コマンドでホストを作成します。

```
# hammer host create --name "My_Unattended_Host" --organization "My_Organization" \
--location "My_Location" --hostgroup "My_Host_Group" --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。

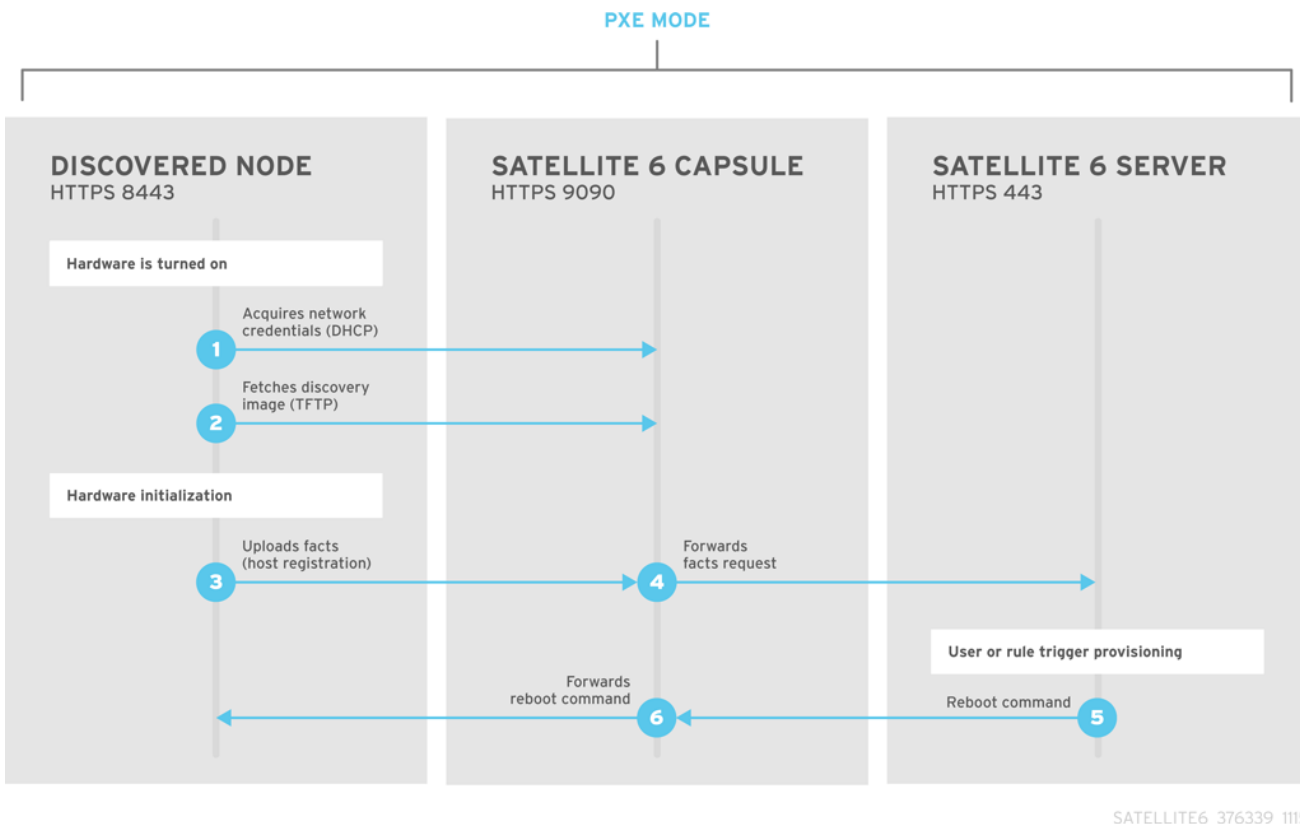
```
# hammer host interface update --host "test1" --managed true \
--primary true --provision true
```

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

Red Hat Satellite には、Satellite インベントリーに含まれていないネットワーク上に存在する空のホストを自動的に検出する方法があります。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 に送り返す検出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リストが作成されます。

Satellite Server では Discovery サービスはデフォルトで有効になっていますが、グローバルテンプレートのデフォルト設定では、ローカルのハードドライブからブートされます。Discovery を使用するには、テンプレートのデフォルトのエントリを `discovery` に変更する必要があります。



Satellite Server を使用して Discovery イメージを提供するには、**foreman-discovery-image** および **rubygem-smart\_proxy\_discovery** 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** パッケージには、Satellite Server の統合 Capsule など Capsule Server を、Discovery サービスのプロキシとして機能するように設定します。

インストールが完了したら、**ホスト > 検出されたホスト** に移動して、新規メニューのオプションを表示できます。

### 6.4.1. 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. Capsule Server で以下のコマンドを入力します。

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

```
# foreman-maintain service restart
```

2. Satellite Web UI にログインし、**インフラストラクチャー > Capsule** に移動します。



3. Capsule Server をクリックして、アクション リストから **リフレッシュ** を選択します。機能リストで **Discovery** を特定して、Discovery サービスが実行中であることを確認します。

## サブネット

検出可能なホストを含むすべてのサブネットには、検出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。

これを確認するには、インフラストラクチャー > **Capsules** に移動して、使用する Capsule Server で Discovery 機能が表示されていることを確認します。表示されていない場合には、**Refresh features (機能の更新)** をクリックします。

Satellite web UI で、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、使用する **Discovery プロキシ** 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。

### 6.4.2. テンプレート PXELinux Discovery スニペットのプロビジョニング

BIOS プロビジョニングの場合は、**ホスト > テンプレートのプロビジョニング** ウィンドウの **PXELinux global default** テンプレートには、スニペット **pxelinux\_discovery** が含まれます。このスニペットには、以下のような行が含まれています。

```

LABEL discovery
  MENU LABEL Foreman Discovery Image
  KERNEL boot/fdi-image/vmlinuz0
  APPEND initrd=boot/fdi-image/initrd0.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=<%= foreman_server_url %> proxy.type=foreman
  IPAPPEND 2

```

**KERNEL** および **APPEND** オプションを指定して、検出イメージおよび ramdisk 起動します。**APPEND** オプションには、**proxy.url** パラメーターと、引数として **foreman\_server\_url** マクロが含まれます。このマクロは、Satellite Server の完全な URL を解決します。

UEFI プロビジョニングの場合は、**ホスト > テンプレートのプロビジョニング** ウィンドウの **PXEgrub2 global default** テンプレートに、スニペット **pxelinux\_discovery** が含まれます。

```

menuentry 'Foreman Discovery Image' --id discovery {
  linuxefi boot/fdi-image/vmlinuz0 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=<%=
  foreman_server_url %> proxy.type=foreman BOOTIF=01-$mac
  initrdefi boot/fdi-image/initrd0.img
}

```

Capsule を使用して検出の手順をプロキシ化するには、**/var/lib/tftpboot/pxelinux.cfg/default** または **/var/lib/tftpboot/grub2/grub.cfg** を編集して、使用する Capsule Server の FQDN に URL を変更します。

グローバルテンプレートは、Satellite Server と、TFTP 機能が有効化されている Capsules すべてで利用できます。

### 6.4.3. テンプレートとスニペットの変更

テンプレートを使用するには、Satellite Web UI で、**管理 > 設定** に移動して、**プロビジョニング** タブをクリックし、使用するテンプレートを設定します。

テンプレートとスニペットは変更されないようにロックされています。テンプレートまたはスニペットを編集するには、クローンを作成して、一意の名前で保存してから、作成したクローンを編集してください。

含めるテンプレートまたはスニペットを変更する場合には、変更は、Satellite Server のデフォルトの PXE テンプレートに伝搬する必要があります。**ホスト > テンプレートのプロビジョニング** に移動して、**PXE デフォルトの構築** をクリックします。これにより、Satellite Server の PXE テンプレートが更新されます。

## proxy.url の引数

Satellite のインストールプロセス中に、デフォルトオプション **--enable-foreman-plugin-discovery** を使用する場合には、テンプレートの **proxy.url** 引数を編集して Discovery サービスを提供する Capsule Server の URL を設定できます。**proxy.url** の引数を、使用する別のプロビジョニング Capsule の IP アドレスまたは FQDN に変更できますが、**9090** などポート番号を追加してください。Satellite のインストール時に **--foreman-proxy-ssl-port** オプションで別のポート番号を使用した場合には、このポート番号を追加する必要があります。Satellite IP アドレスまたは FQDN を使用するように、**proxy.url** 引数を編集して、検出されたホストを直接 Satellite Server と通信させることも可能です。

## proxy.type の引数

**proxy.url** の引数に Capsule Server の FQDN を使用する場合には、**proxy.type** 引数を **proxy** に指定します。Satellite の FQDN を使用する場合は、**proxy.type** の引数を **foreman** に更新してください。

```
proxy.url=https://capsule.example.com:9090 proxy.type=proxy
```

## Capsule のホスト名のレンダリング

Satellite 6 は、全 TFTP Capsules に同じテンプレートをデプロイし、Capsule のホスト名をレンダリングする変数やマクロがありません。ハードコードされた **proxy.url** は、複数の TFTP Capsule を連携できません。回避策として、**PXE デフォルトの構築** をクリックするたびに、SSH を使用して TFTP ディレクトリーの設定ファイルを編集してください。

## Discovery サービスのデフォルト設定

BIOS、UEFI いずれの場合も、現在の Satellite インベントリーに存在しないホスト用に起動するデフォルトのサービスとして Discovery サービスを設定するには、以下の手順を実行します。

1. Satellite Web UI で、**管理 > 設定** に移動して、**プロビジョニング** タブをクリックします。
2. **デフォルトの PXE グローバルテンプレートエントリー** の場合は、**値** のコラムに **discovery** と入力します。

## タグ付けされた VLAN プロビジョニング

タグ付けされた VLAN プロビジョニングを使用して、Discovery サービスにより検出要求が送信されるようにする場合には、以下の情報を、Discovery テンプレートの **KERNEL** オプションに追加します。

```
fdi.vlan.primary=example_VLAN_ID
```

## 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。ブートメニューに、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。

Satellite Web UI で、**ホスト > 検出されたホスト** に移動すると、新たに検出されたホストが表示されます。検出されたホストは 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 アドレスが `ab:cd:ef:12:34:56` の検出されたホストに、**macabcdef123456** というホスト名を設定します。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。

#### 6.4.4. 検出されたホストの自動コンテキスト

Satellite Server は以下の規則の順番に従って、組織とロケーションを検出されたホストに割り当てます。

1. **Discovery 組織** または **Discovery ロケーション** の値を設定する場合には、**管理 > 設定 > Discovered** に移動します。
2. ホストの **foreman\_organization** または **foreman\_location** ファクトを設定する場合には、**管理 > 設定 > Puppet** セクションのファクト名を、**デフォルトの組織** および **デフォルトのロケーション** のファクト設定に指定します。
3. 検出されたホストが Satellite で定義されたサブネットを使用する場合には、このホストは、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。
4. 上記の条件に何も該当しない場合には、Satellite は最初の組織とロケーションを名前順に割り当てます。

組織またはロケーションは、**検出されたホスト** ページの一括処理メニューを使用して変更できます。検出されたホストを選択し、**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。

## 6.5. 検出されたホストからのホストの作成

検出された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、検出された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。

### 手順

検出されたホストからホストを作成するには、以下の手順を行います。

1. Satellite Web UI で **ホスト > 検出されたホスト** に移動します。使用するホストを選択して、リストの右側にある **プロビジョニング** をクリックします。
2. 以下の2つのオプションから1つ選択します。
  - ホストグループからホストをプロビジョニングするには、ホストグループ、組織、場所を選択してから、**ホストの作成** をクリックします。
  - さらにカスタマイズしてホストをプロビジョニングするには、**ホストのカスタマイズ** をクリックして、新規ホストに指定する追加情報を入力します。
3. フィールドに値が投入されていることを確認します。特に以下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。
4. Satellite Server が、ホストの最初のインターフェースに **Managed (管理)**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、選択してください。
  5. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。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。
  6. **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。  
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。
  7. **送信** をクリックしてホストの詳細を保存します。

ホストのプロビジョニングが完了したら、検出されたホストはコンテンツホストになります。このホストを表示するには、**ホスト > コンテンツホスト** に移動します。

## CLI をご利用の場合

1. プロビジョニング用の検出されたホストを特定します。

```
# hammer discovery list
```

2. ホストを選択し、ホストグループを使用してプロビジョニングします。新しいホスト名は、**--new-name** オプションを使用して設定します。

```
# hammer discovery provision --name "host_name" \  
--new-name "new_host_name" --organization "My_Organization" \  
--location "My_Location" --hostgroup "My_Host_Group" --build true \  
--enabled true --managed true
```

これにより、検出されたホストの一覧からホストが削除され、プロビジョニング設定を使用してホストのエントリが作成されます。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。インストールが完了すると、ホストは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。

## 6.6. DISCOVERY ルールの作成

検出された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、検出されたホストが、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。

### NIC の考慮事項

現在、自動プロビジョニングでは NIC の設定はできません。すべてのシステムは、Discovery 時に検出された NIC 設定でプロビジョニングされていますが、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、NIC を後で設定することができます。

## 手順

ルールを作成するには、以下の手順を行います。

1. Satellite Web UI で **設定 > Discovery ルール** に移動します。 **ルールの作成** をクリックし、以下の詳細を入力します。
2. **名前** フィールドには、ルールの名前を入力します。
3. **検索** には、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を入力します。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`
4. **ホストグループ** リストから、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を選択します。
5. **ホスト名** フィールドには、複数ホストのホスト名を決定するためのパターンを入力します。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。
  - `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、文字列と連結できない整数を返すため、`to_s` 関数を使用して整数を文字列に変更します。ホスト名のパターンを作成する時に、作成されるホスト名が一意の名前であることを確認してください。ホスト名は、数字で始めることや、アンダースコアあるいはドットを含めることができません。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。
6. **ホストの制限** フィールドには、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を入力します。無制限に設定するには **0** を使用します。
7. **優先度** フィールドには、ルール間の優先度を設定する数値を入力します。値が低いルールほど優先度が高くなります。
8. **有効化** リストから、ルールを有効化するかどうかを選択します。
9. ルールに異なるプロビジョニングコンテキストを設定するには、**組織** および **ロケーション** タブをクリックして、使用するコンテキストを選択します。
10. **送信** をクリックしてルールを保存します。
11. **ホスト > 検出されたホスト** に移動して、以下の2つのオプションから1つ選択します。
  - 右側の **検出されたホスト** リストから、**自動プロビジョニング** を選択して、単一のホストを自動的にプロビジョニングします。
  - ウィンドウの右上の **すべてを自動プロビジョニング** をクリックして、全ホストを自動的にプロビジョニングします。

## CLI をご利用の場合

**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 "My_Host_Group" \  
--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"
```

## 6.7. PXE を使用しないプロビジョニングによるホストの作成

一部のハードウェアには PXE ブートインターフェースがありません。Red Hat Satellite 6 は、DHCP や TFTP など、PXE を使用しない Discovery サービスを提供します。Satellite では、PXE ブートなしでホストをプロビジョニングできます。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 が生成されます。この ISO を使用して、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 のタイプ

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。

### 手順

PXE なしのプロビジョニングでホストを作成するには、以下の手順を行います。

1. Satellite web UI で、**ホスト > ホストの作成** に移動します。
2. **名前** フィールドには、プロビジョニングされたシステムのホスト名に設定する名前を入力します。
3. **組織** および **ロケーション** タブをクリックし、要件に合うようにコンテキストを変更します。
4. **ホストグループ** リストから、フォームの生成に使用するホストグループを選択します。

5. インターフェース タブをクリックして、ホストのインターフェースで、**編集** をクリックします。
6. フィールドに値が投入されていることを確認します。特に以下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。
7. **MAC アドレス** フィールドには、ホストの MAC アドレスを入力します。
8. Satellite Server が、ホストの最初のインターフェースに **Managed (管理)**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、選択してください。
9. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。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。
10. **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選定できることを確認します。  
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。
11. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
12. **送信** をクリックしてホストの詳細を保存します。

これで、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。

ウィンドウの右上にあるオプションは、**ブートディスク** メニューです。このメニューから、**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** のイメージの1つをダウンロードできます。



#### 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。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場合は、[http://ipxe.org/appnote/hardware\\_drivers](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想定されているハードウェアドライブの一覧を参照してください。

### CLI をご利用の場合

**hammer host create** コマンドでホストを作成します。

```
# hammer host create --name "My_Bare_Metal" --organization "My_Organization" \
--location "My_Location" --hostgroup "My_Host_Group" --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。

```
# 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、使用するホストのブート ISO が作成されます。

必要に応じて、**dd** ユーティリティまたは **livecd-tools** を使用して USB ストレージデバイスに ISO を書き込みます。

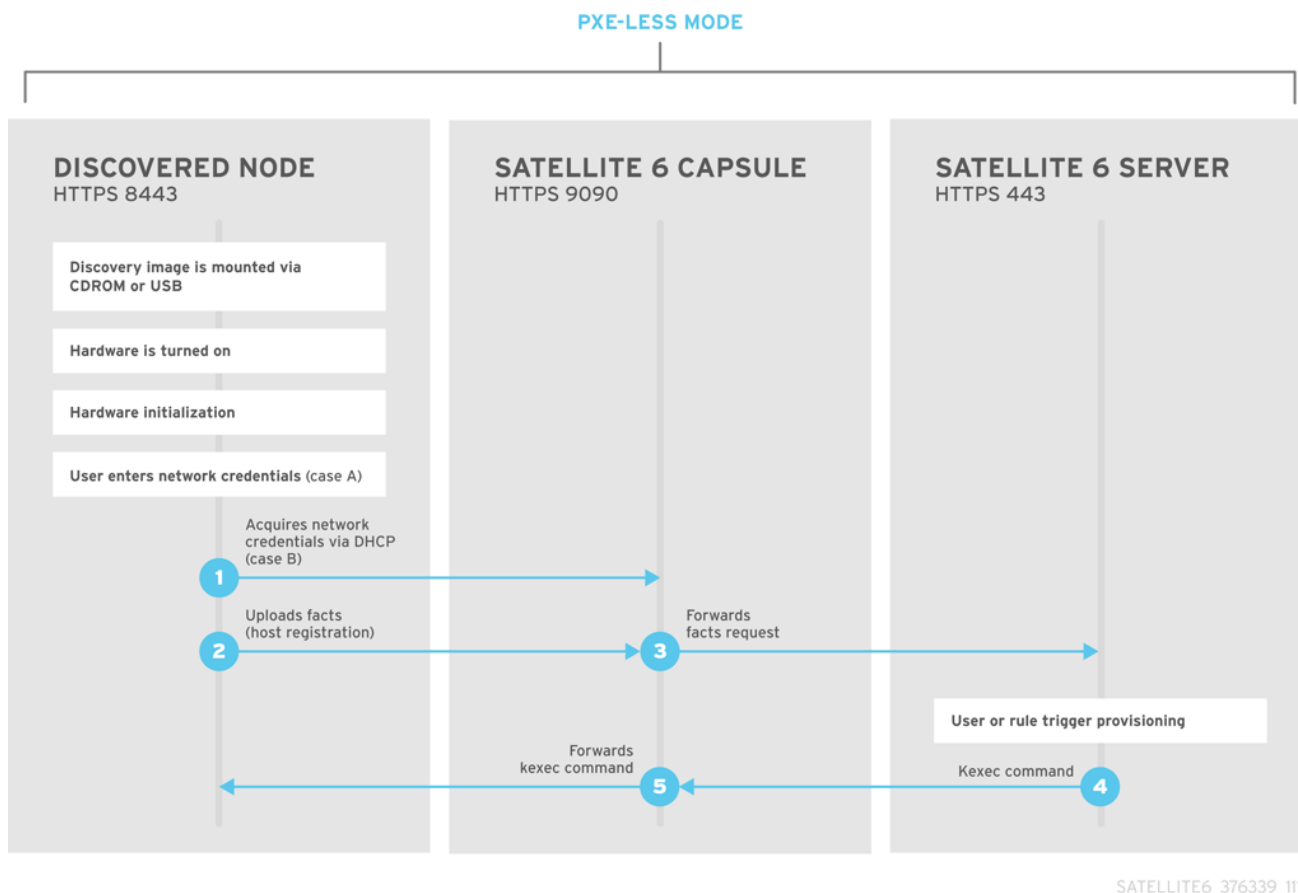
物理ホストを起動して、ISO または USB ストレージデバイスからブートする場合には、ホストは Satellite Server に接続し、キックスタートツリーからオペレーティングシステムのインストールを開始します。

インストールが完了したら、ホストは、アクティベーションキーを使用して Satellite Server も登録し、必要な設定と管理ツールを **Red Hat Satellite Tools** リポジトリからインストールします。

## 6.8. 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-discovery-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-3.4.4-5.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** を選択する場合には、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。これらの詳細を入力した後に、**次へ** を選択します。
- Discovery with DHCP** を選択する場合には、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ています。

プライマリ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、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>

**Connection type** を **Proxy** に設定し、**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** の形式で入力します。必要なカスタムファクトを指定し、**確認** を選択して続きます。

Satellite は Satellite Server の Discovery サービスとの通信が正常であることを報告します。ホスト > **検出されたホスト** に移動して、新たに検出されたホストを表示します。

検出されたホストに関する詳細情報は、「[検出されたホストからのホストの作成](#)」を参照してください。

## 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-3.4.4-5.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/** ディレクトリー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-3.4.4-5.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。

検出されたホストに関する詳細情報は、「[検出されたホストからのホストの作成](#)」を参照してください。

## 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。

## 6.9. プロビジョニング中の SSH キーのデプロイ

この手順を使用して、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します。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。

### 手順

プロビジョニング時に SSH キーをデプロイするには、以下の手順を実行します。

1. Satellite Web UI で、**ホスト > プロビジョニングテンプレート** に移動します。
2. プロビジョニングテンプレートを作成するか、クローンを作成して既存の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. 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が関連付けられた OS を使用してホストを再ビルドします。詳細は、『**ホストの管理ガイド**』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。

## 6.10. SATELLITE DISCOVERY イメージの構築

この手順を使用して、Satellite Discovery イメージを構築するか、設定ファイルを変更した場合にはイメージを再構築します。

実稼働環境の Satellite や Capsule でこの手順は使用しないでください。

### 前提条件

**livecd-tools** パッケージをインストールします。

```
# yum install livecd-tools
```

Anaconda インストーラーは、HTTPS では公開できないので、Kickstart リポジトリの場合は HTTP での公開を有効にする必要があります。

1. Satellite Web UI で、**コンテンツ** > **製品** に移動して、**製品** ウィンドウで、**リポジトリ** タブを選択します。
2. Kickstart リポジトリと、**Publish via HTTP** オプションを選択して、**編集** アイコンをクリックし、チェックボックスを選択して **保存** をクリックします。
3. Satellite リポジトリにも先ほどの手順を繰り返します。

HTTP 経由で公開すると、どの Red Hat リポジトリにも適用されないことに注意してください。

### 手順

Satellite Discovery イメージを構築するには、以下の手順を実行します。

1. **/usr/share/foreman-discovery-image/foreman-discovery-image.ks** ファイルを開いて編集します。

```
# vim /usr/share/foreman-discovery-image/foreman-discovery-image.ks
```

2. `repo --name=rhel --baseurl=http://download/00000` は、独自の RHEL のレポに、Satellite はご利用のリポジトリ URL に置き換えます。URL を特定するには、**コンテンツ** > **製品** に移動して、**リポジトリ** タブをクリックし、両リポジトリの URL をファイルにコピーします。

```
repo --name=rhel --baseurl=http://download/released/RHEL-7/7.4/Server/x86_64/os/
repo --name=sat --baseurl=http://download2/nightly/Satellite/6.5/candidate/latest-Satellite-6.5-RHEL-7/compose/Satellite/x86_64/os/
```

3. `livecd-creator` ツールを実行します。

```
# livecd-creator --title="Discovery-Image" \
--compression-type=xz \
--cache=var/cache/build-fdi \
--config /usr/share/foreman-discovery-image/foreman-discovery-image.ks \
--fslabel fdi \
--tmpdir /var/tmp
```

`--fslabel` オプションで `fdi` を変更した場合には、イメージの読み込み時に kernel コマンドラインの `root` ラベルを変更する必要があります。`fdi` または別の名前を、この手順の一部で作成した `.iso` ファイルに追加します。PXE Discovery ツールは、`.iso` から PXE に変換時に、この名前を使用します。

このプロセスには 3 GB 近くの容量が必要で、システムの swap 領域が少ない場合に `/tmp` で問題が発生する可能性があるため、`/var/tmp` を使用します。

4. `fdi.iso` ファイルが作成されていることを確認します。

```
# ls *.iso -h
```

`.iso` ファイルを作成した場合には、`.iso` ファイルをネットワークでもローカルでも起動できます。以下の手順のいずれかを実行してください。

#### ネットワーク経由で iso ファイルを起動する場合:

1. ネットワーク経由で初期の ramdisk と kernel ファイルを `.iso` ファイルから抽出するには、以下のコマンドを入力します。

```
# discovery-iso-to-pxe fdi.iso
```

2. ブートファイルを保存するディレクトリを作成します。

```
# mkdir /var/lib/tftpboot/boot/myimage
```

3. `initrd0.img` および `vmlinuz0` ファイルを新しいディレクトリにコピーします。
4. `/var/lib/tftpboot/pxelinux.cfg` ファイルで **KERNEL** と **APPEND** エントリを編集して、独自の初期 ramdisk および kernel ファイルに関する情報を追加します。

#### ローカルで iso ファイルを起動する場合:

ローカルで起動するハイブリッドの `.iso` ファイルを作成する場合には、以下の手順を行います。

1. `.iso` ファイルを `.iso` のハイブリッドファイルに変換して PXE プロビジョニングを行うには、以下のコマンドを入力します。

```
# isohybrid --partok fdi.iso
```

**grub2** パッケージをインストールする場合には、以下のコマンドを使用して、**grub2** ブートローダーにインストールすることも可能です。

```
# isohybrid --partok --uefi fdi.iso
```

2. **md5** チェックサムを **.iso** ファイルに追加して、Satellite でインストールメディアのバリデーションテストに合格するには、以下のコマンドを入力します。

```
# implantisomd5 fdi.iso
```

## 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。

### 7.1. KVM プロビジョニングの要件

KVM プロビジョニングの要件には以下が含まれます。

- 使用する Red Hat Enterprise Linux バージョンの同期済みのコンテンツリポジトリ。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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。
- ホスト登録用のアクティベーションキー。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。

#### Libvirt コンピュートリソースをプロビジョニングするためのユーザーロールとパーミッション

Satellite で Libvirt ホストをプロビジョニングするには、以下のロールをユーザーアカウントに指定する必要があります。

- **ホストの編集**
- **ホストの表示**

詳しい情報は、『[Red Hat Satellite 管理ガイド](#)』の「[ユーザーへのロールの割り当て](#)」を参照してください。

また、以下のパーミッションを指定したカスタムロールを作成する必要があります。

- **view\_compute\_resources**
- **destroy\_compute\_resources\_vms**
- **power\_compute\_resources\_vms**
- **create\_compute\_resources\_vms**
- **view\_compute\_resources\_vms**
- **view\_locations**

- `view_subnets`

ロール作成に関する詳細は、「[ロールの作成](#)」を参照してください。ロールにパーミッションを追加する方法については、『Red Hat Satellite 管理ガイド』の「[ロールへのパーミッションの追加](#)」を参照してください。

## 7.2. SATELLITE SERVER の KVM 接続設定

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。

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

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

2. キーペアを生成します。

```
$ ssh-keygen
```

3. 公開キーを KVM サーバーにコピーします。

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

4. **foreman** ユーザーのバッシュシェルを終了します。

```
$ exit
```

5. **libvirt-client** パッケージをインストールします。

```
# yum install libvirt-client
```

6. 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。

```
# su foreman -s /bin/bash -c '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。以下に例を示します。

```
qemu+ssh://root@kvm.example.com/system
```

## 7.3. SATELLITE SERVER への KVM 接続の追加

この手順を使用して、Satellite Server のコンピュータリソースに KVM 接続を追加します。

### 手順

Satellite に KVM 接続を追加するには、以下の手順を行います。

1. 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** に移動して、コンピュータリソースのウィンドウで **コンピュータリソースの作成** をクリックします。
2. **名前** フィールドには、新規コンピュータリソースの名前を入力します。



3. **プロバイダー** の一覧から **Libvirt** を選択します。
4. **説明** フィールドには、コンピュータリソースの説明を入力します。
5. **URL** フィールドには、KVM サーバーへの **libvirt** への接続 URL を入力します。以下に例を示します。

```
qemu+ssh://root@kvm.example.com/system
```

6. **ディスプレイタイプ** リストから、**VNC** または **Spice** を選択します。
7. **オプション**: 無作為に生成したパスワードで、新規ホストのコンソールアクセスのセキュリティを確保するには、**ディスプレイ接続時にランダムに生成されたパスワードを設定します** というチェックボックスを選択します。以下のコマンドの出力から、**libvirtd** ホストからゲスト仮想マシンコンソールにアクセスするための、VNC コンソールのパスワードを取得できます。

```
# virsh edit your_VM_name
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'
passwd='your_randomly_generated_password'>
```

パスワードは、virt-managerなどで、仮想マシンのコンソールを開くたびに無作為に生成されます。

8. **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。
9. **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ていることを確認します。他のコンテキストをこれらのタブに追加します。
10. **送信** をクリックして KVM 接続を保存します。

## CLI をご利用の場合

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

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

## 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ホストを作成する場合は、イメージの情報を Satellite Server に追加する必要があります。これには、アクセスの情報およびイメージの場所が含まれます。

### 手順

Satellite Server に KVM イメージを追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー** > **コンピュータリソース** に移動して、コンピュータリソースのウィンドウで KVM 接続の名前をクリックします。
2. **イメージ** タブをクリックして、**イメージの作成** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。

4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **アーキテクチャー** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
7. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
8. **ユーザーデータ** リストから、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。
9. **イメージパス** フィールドには、KVM サーバーのイメージを参照する完全パスを入力します。以下に例を示します。

```
/var/lib/KVM/images/TestImage.qcow2
```

10. **送信** をクリックしてイメージの詳細を保存します。

## CLI をご利用の場合

**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 version" --architecture "x86_64" --username root \
--user-data false --uuid "/var/lib/libvirt/images/TestImage.qcow2" \
--compute-resource "My_KVM_Server"
```

## 7.5. KVM の詳細のコンピュータプロファイルへの追加

コンピュータプロファイルにこれらのハードウェア設定を追加して、KVM ベースの仮想マシンに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。

### 手順

Red Hat Virtualization の詳細をコンピュータプロファイルに追加するには、以下の手順を行います。

1. Satellite Web UI で、**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。
2. コンピュートプロファイルのウィンドウで、既存のコンピュータリソースの名前をクリックするか、**コンピュータプロファイルの作成** をクリックして、コンピュータプロファイルの作成に使用するコンピュータリソースを選択します。
3. **CPU** フィールドには、新規ホストに割り当てる CPU の数を入力します。
4. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を入力します。
5. **イメージ** リストから、イメージベースのプロビジョニングを実行する場合には、使用するイメージを選択します。
6. **ネットワークインターフェース** リストから、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選択します。ネットワークインターフェースは複数作成できますが、最低でも1つのインターフェースは Capsule が管理するネットワークを参照する必要があります。

7. **ストレージ** エリアには、ホストのストレージパラメーターを入力します。ホストのボリュームは複数作成できます。
8. **送信** をクリックしてコンピュートプロファイルの設定を保存します。

## CLI をご利用の場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.5 ではまだ実装されていません。代わりに、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 サーバーでのホストの作成

Satellite では、KVM プロビジョニングを使用してネットワーク接続経由でおよび、既存のイメージをもとに、ホストを作成できます。

既存のイメージでホストを作成する場合は、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。

ネットワーク接続経由でホストを作成する場合には、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、新規ホストは KVM 仮想マシン上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスする必要があります。この新しいホストエントリーにより、KVM サーバーが仮想マシンを作成して起動するようトリガーします。仮想マシンが仮想ネットワークで定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、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。

## DHCP の競合

ネットワークベースのプロビジョニングでは、KVM サーバーの仮想ネットワークをプロビジョニングに使用する場合には、DHCP 割り当てを行わないネットワークを選択します。これにより、新規ホストの起動時に、Satellite Server と DHCP が競合してしまうためです。

## 手順

KVM ホストを作成するには、以下の手順を行います。

1. Satellite web UI で、**ホスト > ホストの作成** に移動します。
2. **名前** フィールドには、プロビジョニングされたシステムのホスト名に設定する名前を入力します。
3. **組織** および **ロケーション** タブをクリックして、プロビジョニングのコンテキストが現在のコンテキストに自動的に設定されていることを確認します。
4. **ホストグループ** リストから、フォームの生成に使用するホストグループを選択します。
5. **デプロイ先** リストから、KVM 接続を選択します。
6. **コンピュートプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
7. **インターフェース** タブをクリックして、ホストのインターフェースで **編集** をクリックします。
8. フィールドに値が自動投入されていることを確認します。特に以下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。

9. **MAC アドレス** フィールドは空白であることを確認します。KVM サーバーは、MAC アドレスをホストに割り当てます。
10. ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションが自動的に選択されていることを確認します。選択されていない場合は、それらを選択してください。
11. インターフェースウィンドウで、KVM 固有のフィールドにコンピュータプロファイルの設定が入力されていることを確認します。必要に応じてこれらの設定を変更します。
12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. ネットワークベースのプロビジョニングでは、**プロビジョニング方法** が **ネットワークベース** に設定されており、イメージベースのプロビジョニングでは **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。
14. **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。
15. **仮想マシン** タブをクリックして、これらの設定には、ホストグループおよびコンピュータプロファイルからの情報が入力されていることを確認し、必要に応じて変更してください。
16. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
17. **送信** をクリックしてホストエントリーを保存します。

## CLI をご利用の場合

**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。

```
# hammer host create --name "kvm-test1" --organization "My_Organization" \  
--location "New York" --hostgroup "Base" \  
--compute-resource "My_KVM_Server" --provision-method build \  
--build true --enabled true --managed true \  
--interface  
"managed=true,primary=true,provision=true,compute_type=network,compute_network=examplenetwork" \  
--compute-attributes="cpus=1,memory=1073741824" \  
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```

**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。

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

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、「[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。

## 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) および Red Hat Enterprise Virtualization (バージョン 3.6 以前) は、Red Hat Enterprise Linux 上に構築する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 では、Red Hat Virtualization REST API バージョン 3 で仮想化機能を管理できます。仮想マシンの作成や電源状態の制御など、REST API バージョン 4 はまだ Satellite 6 ではサポートされていません。

以下の手順を使用して、Red Hat Virtualization 環境に接続を追加して、仮想マシンをプロビジョニングします。

### 8.1. RED HAT VIRTUALIZATION プロビジョニングの要件

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

- 使用する Red Hat Enterprise Linux バージョンの同期済みのコンテンツリポジトリ。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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。
- ホスト登録用のアクティベーションキー。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

この手順を使用して、Satellite Server のコンピュータリソースに Red Hat Virtualization 接続を追加します。

#### 手順

Red Hat Virtualization 接続を Satellite に追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー** > **コンピュータリソース** に移動して、コンピュータリソースのウィンドウで **コンピュータリソースの作成** をクリックします。
2. **名前** フィールドには、新規コンピュータリソースの名前を入力します。
3. **プロバイダー** の一覧から **RHV** を選択します。
4. **説明** フィールドには、コンピュータリソースの説明を入力します。
5. **URL** フィールドには、Red Hat Virtualization Manager API への接続 URL を入力します。  
たとえば、RHEV 3.6 以前では、URL は <https://rhvm.example.com/ovirt-engine/api> の形式、  
RHV 4.0 以降では <https://rhvm.example.com/ovirt-engine/api/v3> の形式となっています。
6. オプションで **APIv4 の使用 (プレビュー)** のチェックボックスを選択して、新しいエンジン API を評価します。



### 警告

このセクションに記載する項目は、テクノロジープレビューとして提供しています。テクノロジープレビューの適用範囲のステータスに関する詳細情報およびそれに伴うサポートへの影響については、「[テクノロジープレビュー機能のサポート範囲](#)」を参照してください。

7. **ユーザー** フィールドには、ユーザー名と、Red Hat Virtualization Manager のリソースにアクセスするためのパーミッションを入力します。
8. **パスワード** フィールドで、ユーザーのパスワードを入力します。
9. **データセンターのロード** をクリックして、**データセンター** リストに Red Hat Virtualization 環境のデータセンターを入力します。
10. **データセンター** リストからデータセンターを選択します。
11. **クォータ ID** から 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。
12. **X509 証明機関** フィールドには、SSL/TLS アクセスの証明機関を入力します。または、フィールドを空白にすると、サーバーによる最初の API 要求時に、自己署名証明書が生成されます。
13. **ロケーション** タブをクリックして、使用するロケーションを選択します。
14. **組織** タブをクリックして、使用する組織を選択します。
15. **送信** をクリックしてコンピュートリソースを保存します。

### CLI をご利用の場合

Red Hat Virtualization 接続を作成するには、**--provider** で **Ovirt** を、**--uuid** で使用するデータセンターの UUID を指定して、**hammer compute-resource create** コマンドを入力します。

```
# hammer compute-resource create \
--name "My_RHV" --provider "Ovirt" \
--description "RHV server at rhvm.example.com" \
--url "https://rhvm.example.com/ovirt-engine/api" \
--use-v4 "false" --user "Satellite_User" \
--password "My_Password" \
--locations "New York" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

オプションで新規エンジン API を評価するには、**--use-v4** オプションを **false** から **true** に変更します。



**警告**

このセクションに記載する項目は、テクノロジープレビューとして提供しています。テクノロジープレビューの適用範囲のステータスに関する詳細情報およびそれに伴うサポートへの影響については、「[テクノロジープレビュー機能のサポート範囲](#)」を参照してください。

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

Red Hat Virtualization は、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ホストを作成する場合には、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。

**手順**

Red Hat Virtualization 接続を Satellite Server に追加するには、以下の手順を行います。

1. 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** に移動して、コンピュートリソースのウィンドウで Red Hat Virtualization 接続の名前をクリックします。
2. **イメージ** タブをクリックして、**新規イメージ** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。
4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **アーキテクチャー** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
7. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
8. **イメージ** リストから Red Hat Virtualization のイメージの名前を選択します。
9. **送信** をクリックしてイメージの詳細を保存します。

**CLI をご利用の場合**

**hammer compute-resource image create** コマンドでイメージを作成します。**--uuid** オプションを使用して、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 "My_RHV"
```

## 8.5. RED HAT VIRTUALIZATION の詳細のコンピュータプロファイルへの追加

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タプロファイルに追加することで実行できます。

### 手順

Red Hat Virtualization の詳細をコンピュータプロファイルに追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー** > **コンピュータプロファイル** に移動して、コンピュータプロファイルのウィンドウで Red Hat Virtualization 接続の名前をクリックします。
2. **クラスター** リストから Red Hat Virtualization 環境のターゲットホストクラスターを選択します。
3. **テンプレート** リストから **コア数** および **メモリー** 設定に使用する RHV テンプレートを選択します。
4. **コア数** フィールドには、新規ホストに割り当てる CPU コア数を入力します。
5. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を入力します。
6. **イメージ** リストから、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を選択します。
7. **ネットワークインターフェース** エリアには、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入力します。ネットワークインターフェースは、複数作成することができますが、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参照している必要があります。各ネットワークインターフェースには、以下の情報を入力します。
  - a. **名前** フィールドに、ネットワークインターフェースの名前を入力します。
  - b. **ネットワーク** リストから、使用する論理ネットワークを選択します。
8. **ストレージ** エリアには、ホストのストレージパラメーターを入力します。ホストのボリュームは複数作成できます。ボリュームごとに以下の情報を入力します。
  - a. **Size (GB)** には、新しいボリュームのサイズを GB 単位で入力します。
  - b. **ストレージドメイン** リストからボリュームのストレージドメインを選択します。
  - c. **ディスクの事前割り当て** から、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のいずれかを選択します。
  - d. **Bootable** リストから、ブータブルか、ブータブル以外のボリュームのいずれかを選択します。
9. **送信** をクリックしてコンピュータプロファイルを保存します。

### CLI をご利用の場合

コンピュータプロファイルの CLI コマンドは、Red Hat Satellite 6.5 ではまだ実装されていません。代わりに、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

Satellite では、ネットワーク接続経由か、既存のイメージをもとに、Red Hat Virtualization ホストを作成できます。

ネットワーク接続経由でホストを作成する場合には、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、新規ホストは Red Hat Virtualization 仮想マシン上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスする必要があります。この新しいホストエントリーにより、Red Hat Virtualization サーバーが仮想マシンを作成して起動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で定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、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。

## DHCP の競合

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。

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。

## 手順

Red Hat Virtualization Server のホストを作成するには、以下の手順を行います。

1. Satellite web UI で、**ホスト** > **新規ホスト** に移動します。
2. **名前** フィールドには、プロビジョニングされたシステムのホスト名に設定する名前を入力します。
3. **組織** および **ロケーション** タブをクリックして、プロビジョニングのコンテキストが現在のコンテキストに自動的に設定されていることを確認します。
4. **ホストグループ** リストから、フォームの生成に使用するホストグループを選択します。
5. **デプロイ先** リストから、Red Hat Virtualization 接続を選択します。
6. **コンピュートプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
7. **インターフェース** タブをクリックして、ホストのインターフェースで **編集** をクリックします。
8. フィールドに値が自動投入されていることを確認します。特に以下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。
9. **MAC アドレス** を空白であることを確認します。Red Hat Virtualization サーバーが MAC アドレスをホストに割り当てます。
10. ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションが自動的に選択されていることを確認します。選択されていない場合は、それらを選択してください。
11. インターフェースウィンドウで、Red Hat Virtualization 固有のフィールドにコンピュートプロファイルの設定が入力されていることを確認します。必要に応じてこれらの設定を変更します。

12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. ネットワークベースのプロビジョニングでは、**プロビジョニング方法** が **ネットワークベース** に設定されており、イメージベースのプロビジョニングでは **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。
14. **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。
15. **仮想マシン** タブをクリックして、これらの設定には、ホストグループおよびコンピュータプロファイルからの情報が入力されていることを確認し、必要に応じて変更してください。
16. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
17. **送信** をクリックしてホストエントリーを保存します。

## CLI をご利用の場合

ネットワークベースのプロビジョニングでホストを作成するには、**--provision-method build** を指定して **hammer host create** コマンドを実行します。

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

イメージベースのプロビジョニングでホストを作成するには、**--provision-method image** を指定して **hammer host create** コマンドを実行します。

```
# hammer host create --name "rhv-test2" --organization "My_Organization" \  
--location "New York" --hostgroup "Base" \  
--compute-resource "My_RHV" --provision-method image \  
--image "Test_RHV_Image" --enabled true --managed true \  
--interface  
"managed=true,primary=true,provision=true,compute_name=eth0,compute_network=satnetwork" \  
--compute-attributes="cluster=Default,cores=1,memory=1073741824,start=true" \  
--volume="size_gb=20G,storage_domain=Data,bootable=true"
```

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、「[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。

## 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。

### 9.1. VMWARE VSPHERE プロビジョニングの要件

VMware vSphere プロビジョニングの要件には以下が含まれます。

- 使用する Red Hat Enterprise Linux バージョンの同期済みのコンテンツリポジトリ。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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)。
- ホスト登録用のアクティベーションキー。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)
- 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

この手順を使用して、Satellite Server のコンピュートリソースに VMware vSphere 接続を追加します。

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。

#### 手順

接続を追加するには、以下の手順を行います。

1. 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** に移動して、**コンピュートリソースのウィンドウで コンピュートリソースの作成** をクリックします。
2. **名前** フィールドに、リソースの名前を入力します。
3. **プロバイダー** の一覧から **VMware** を選択します。
4. **説明** フィールドには、リソースの説明を入力します。
5. **VCenter/Server (サーバー)** フィールドには、vCenter サーバーの IP アドレスまたはホスト名を入力します。
6. **ユーザー** フィールドには、ユーザー名と、vCenter のリソースにアクセスするためのパーミッションを入力します。
7. **パスワード** フィールドで、ユーザーのパスワードを入力します。
8. **データセンターのロード** をクリックして、VMware vSphere 環境からデータセンターリストを生成します。
9. **データセンター** リストから、このリストから管理する特定のデータセンターを選択します。
10. **フィンガープリント** フィールドは、データセンターからのフィンガープリントが投入されていることを確認します。
11. **ディスプレイタイプ** リストから、**VNC** や **VMRC** などのコンソールタイプを選択します。VNC コンソールは VMware ESXi 6.5 以降ではサポートされません。
12. オプション: **VNC コンソールパスワード** フィールドでは、**ディスプレイ接続時にランダムに生成されたパスワードを設定します** チェックボックスを選択して、無作為に作成されたパスワードで、新規ホストへのアクセスのセキュリティを確保します。**libvirt** ホストからゲスト仮想マシンコンソールにアクセスするための VNC コンソールのパスワードは、以下のコマンドの出力から取得できます。

```
# virsh edit your_VM_name
<graphics type='vnc' port='-1' autoport='yes' listen='0.0.0.0'
passwd='your_randomly_generated_password'>
```

パスワードは、virt-manager などで、仮想マシンのコンソールを開くたびに無作為に生成されます。

13. **キャッシングの有効化** リストからコンピュートリソースのキャッシングを有効化するかどうかを選択できます。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。
14. **ロケーション** および **組織** タブをクリックして、値が現在のコンテキストに自動的に設定されていることを確認します。また別のコンテキストを追加することもできます。
15. **送信** をクリックして接続を保存します。

## CLI をご利用の場合

**hammer compute-resource create** コマンドで接続を作成します。--provider で **Vmware** を選択し、-uid 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 "My_vSphere" \
```

```
--provider "Vmware" \
--description "vSphere server at vsphere.example.com" \
--server "vsphere.example.com" --user "SatelliteUser" \
--password "p@55w0rd!" --locations "New York" --organizations "My_Organization" \
--uuid 72cb9454-81cd-4231-a863-d9baf0f399f8
```

## 9.4. SATELLITE SERVER への 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。

### 手順

イメージを追加するには、以下のステップを実行します。

1. 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** に移動して、コンピュートリソースのウィンドウで VMware vSphere 接続をクリックします。
2. **名前** フィールドに、イメージの名前を入力します。
3. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
4. **アーキテクチャー** リストから、オペレーティングシステムのアーキテクチャーを選択します。
5. **ユーザー** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
6. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
7. **ユーザーデータ** リストから、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。
8. **イメージ** フィールドには vSphere 環境でのテンプレートの相対パスおよび名前を入力します。相対パスにデータセンターを追加しないでください。
9. **送信** をクリックしてイメージの詳細を保存します。

### CLI をご利用の場合

**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 "My_vSphere"
```

## 9.5. VMWARE VSPHERE の詳細のコンピュータプロファイルへの追加

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タプロファイルに追加することで実行できます。

### 手順

VMware vSphere の詳細をコンピュータプロファイルに追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー > コンピュートプロファイル** に移動して、コンピュータプロファイルのウィンドウで vSphere 接続の名前をクリックします。
2. **CPU** フィールドには、新規ホストに割り当てる CPU の数を入力します。
3. **1ソケットあたりのコア数** フィールドには各 CPU に割り当てるコアの数を入力します。
4. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を入力します。
5. **クラスター** フィールドには、VMware 環境のターゲットホストクラスター名を入力します。
6. **リソースプール** リストから、ホストに対して利用可能なリソース割り当てを選択します。
7. **フォルダー** フィールドには、ホストを整理するためのフォルダーを入力します。
8. **ゲスト OS** リストから、VMware vSphere で使用するオペレーティングシステムを選択します。
9. **SCSI コントローラー** リストから、ホストのディスクアクセスの方法を選択します。
10. **仮想ハードウェアのバージョン** リストから、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。
11. 仮想マシンの電源をオンにしたままリソースを追加するには **メモリーホット追加** または **CPU ホット追加** のチェックボックスを選択します。
12. **イメージ** リストから、イメージベースのプロビジョニングを実行する場合には、使用するイメージを選択します。
13. **ネットワークインターフェース** リストから、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選択します。ネットワークインターフェースは複数作成できますが、最低でも1つのインターフェースは Capsule が管理するネットワークを参照する必要があります。
14. Eager Zero シックプロビジョニングを使用するには、**Eager Zero** チェックボックスを選択します。これにチェックを付けない場合には、ディスクは Lazy Zero シックプロビジョニングを使用します。
15. **送信** をクリックしてコンピュータプロファイルを保存します。

## CLI をご利用の場合

コンピュータプロファイルの CLI コマンドは、Red Hat Satellite 6.5 ではまだ実装されていません。代わりに、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 サーバーでのホストの作成

VMware vSphere プロビジョニングプロセスでは、ネットワーク接続経由で、あるいは既存のイメージを使用してホストを作成するオプションがあります。

ネットワークベースのプロビジョニングでは、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、新規ホストは VMware vSphere 仮想マシン上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスする必要があります。この新しいホストエントリーにより、



VMware vSphere サーバーが仮想マシンを作成して起動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で定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、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。

## DHCP の競合

プロビジョニング目的で VMware vSphere 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。

イメージベースのプロビジョニングでは、既存のイメージを新規ボリュームのベースとして使用しません。

## 手順

VMware vSphere サーバーのホストを作成するには、以下の手順を行います。

1. Satellite web UI で、**ホスト** > **新規ホスト** に移動します。
2. **名前** フィールドには、プロビジョニングされたシステムのホスト名に設定する名前を入力します。
3. **組織** および **ロケーション** タブをクリックして、プロビジョニングのコンテキストが現在のコンテキストに自動的に設定されていることを確認します。
4. **ホストグループ** リストから、フォームの生成に使用するホストグループを選択します。
5. **デプロイ先** リストから、VMware vSphere 接続を選択します。
6. **コンピュートプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
7. **インターフェース** タブをクリックして、ホストのインターフェースで **編集** をクリックします。
8. フィールドに値が自動投入されていることを確認します。特に以下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。
9. **MAC アドレス** フィールドが空白であることを確認します。VMware vSphere サーバーは MAC アドレスをホストに割り当てます。
10. ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションが自動的に選択されていることを確認します。選択されていない場合は、それらを選択してください。
11. インターフェースウィンドウには、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。
12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. ネットワークベースのプロビジョニングでは、**プロビジョニング方法** が **ネットワークベース** に設定されており、イメージベースのプロビジョニングでは **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。

14. **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。
15. **仮想マシン** タブをクリックして、これらの設定には、ホストグループおよびコンピュータプロファイルからの情報が入力されていることを確認し、必要に応じて変更してください。
16. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
17. **送信** をクリックしてホストエントリーを保存します。

## CLI をご利用の場合

**hammer host create** コマンドでネットワークからホストを作成し、**--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 "_My_Organization_" \
--location "New York" --hostgroup "Base" \
--compute-resource "_My_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

このコンピュータ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。

## CLI をご利用の場合

**hammer host create** コマンドでイメージからホストを作成し、**--provision-method image** を指定して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 "My_Organization" \
--location "New York" --hostgroup "Base" \
--compute-resource "My_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"
```

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、「[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。

## 9.7. 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。

### 9.7.1. 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を有効化または無効化します。

1. 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** に移動します。
2. 更新する 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. **キャッシュを有効にする** のチェックボックスを選択します。

### 9.7.2. 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。

#### 手順

1. 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する 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 をご利用の場合

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
```

**hammer compute-resource list** コマンドを使用して、コンピュートリソースのキャッシュを更新する VMware サーバーの ID を判断します。

## 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。

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

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。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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。
- ホスト登録用のアクティベーションキー。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

この手順では、Satellite Server のコンピュータリソースに Red Hat OpenStack Platform 接続を追加します。

#### 手順

コンピュータリソースを追加するには、以下の手順を実行します。

1. 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](#) に移動して、コンピュータリソースのウィンドウで [コンピュートリソースの作成](#) をクリックします。
2. **名前** フィールドには、今後使用する時にコンピュータリソースを特定できるように名前を入力します。
3. **プロバイダー** リストから **RHEL OpenStack Platform** を選択します。
4. **説明** フィールドには、リソースの説明を入力します。
5. **URL** フィールドには、**tokens** リソースの OpenStack 認証 keystone サービスの API を参照する URL を入力します。<http://openstack.example.com:5000/v2.0/tokens> の形式を使用します。
6. **ユーザー名** および **パスワード** フィールドには、環境にアクセスするための Satellite の認証ユーザーおよびパスワードを入力します。
7. **ドメイン** フィールドには、V3 認証のドメインを入力します。

8. **テナント** リストから Satellite Server が管理するテナントまたはプロジェクトを選択します。
9. ホストのプライマリーネットワークとして、外部ネットワークを使用するには、**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)** チェックボックスを選択します。
10. **ロケーション** および **組織** タブをクリックして、使用するロケーションと組織が現在のコンテキストに自動的に設定されていることを確認します。他のコンテキストは、これらのタブに追加します。
11. **送信** をクリックして Red Hat OpenStack Platform の接続を保存します。

## CLI をご利用の場合

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

```
# hammer compute-resource create --name "My_OpenStack" \
--provider "OpenStack" \
--description "My OpenStack environment at openstack.example.com" \
--url "http://openstack.example.com:5000/v2.0/tokens" --user "My_Username" \
--password "My_Password" --tenant "openstack" --locations "New York" \
--organizations "My_Organization"
```

## 10.3. SATELLITE SERVER への 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。アクセスの詳細およびイメージの場所など、イメージの詳細を Satellite Server に追加する必要があります。

### 手順

Red Hat OpenStack Platform イメージを追加するには、以下の手順を実行します。

1. 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** に移動して、コンピュートリソースのウィンドウで Red Hat OpenStack Platform 接続の名前をクリックします。
2. **名前** フィールドに、イメージの名前を入力します。
3. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
4. **アーキテクチャー** リストから、オペレーティングシステムのアーキテクチャーを選択します。
5. **ユーザー** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
6. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
7. **イメージ** リストから、OpenStack Image Storage のイメージを選択します。
8. **ユーザーデータ** リストから、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の設定を選択します。
9. **送信** をクリックしてイメージの詳細を保存します。

## CLI をご利用の場合

**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" \  
--user root --user-data true \  
--compute-resource "My_OpenStack_Platform"
```

## 10.4. RED HAT OPENSTACK PLATFORM の詳細のコンピュータプロファイルへの追加

Satellite では、Red Hat OpenStack Platform のインスタンスに特定のハードウェア設定を定義できます。これらのハードウェア設定は、コンピュータプロファイルに追加します。

### 手順

Red Hat OpenStack Platform の詳細をコンピュータプロファイルに追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー > コンピュータプロファイル** に移動して、コンピュータプロファイルのウィンドウで使用するプロファイル名をクリックします。
2. **フレーバー** リストから、ホストに使用する OpenStack Platform のハードウェアのプロファイルを選択します。
3. **利用可能ゾーン (Availability zone)** リストから OpenStack Platform 環境内で使用するターゲットクラスターを選択します。
4. **イメージ** リストから、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を選択します。
5. **テナント** リストから、OpenStack Platform インスタンスのテナントまたはプロジェクトを選択します。
6. **セキュリティグループ** リストから、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を選択します。
7. **内部ネットワーク** から、ホストが参加するプライベートネットワークを選択します。
8. **Floating IP ネットワーク** から、ホストが参加する外部ネットワークを選択して、Floating IP アドレスを割り当てます。
9. **ボリュームからの起動** から、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には、インスタンスはイメージを直接起動します。
10. **新規起動ボリュームサイズ (GB)** フィールドには、新規起動ボリュームのサイズ (GB) を入力します。
11. **送信** をクリックしてコンピュータプロファイルを保存します。

## CLI をご利用の場合

コンピュータプロファイルの CLI コマンドは、Red Hat Satellite 6.5 ではまだ実装されていません。代わりに、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

Satellite では、Red Hat OpenStack Platform サーバーの既存のイメージから、Red Hat OpenStack Platform ホストをプロビジョニングできます。

### 手順

ホストをプロビジョニングするには、以下の手順を行います。

1. Satellite web UI で、**ホスト** > **新規ホスト** に移動します。
2. **名前** フィールドには、プロビジョニングされたシステムのホストに割り当てる名前を入力します。
3. **ホストグループ** リストから、ホストのフィールドに投入するホストグループを選択します。
4. **デプロイ先** リストから、OpenStack Platform 接続を選択します。
5. **コンピュートプロファイル** リストから、クラウドインスタンスベースの設定を自動的に投入するために使用するプロファイルを選択します。
6. **インターフェース** タブをクリックして、ホストのインターフェースの **編集** をクリックします。ホスト タブの **名前** が **DNS 名** になっていること、また Satellite Server が自動的に新規ホストに IP アドレスを割り当てていることを確認します。
7. **MAC アドレス** フィールドが空白であることを確認します。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。
8. 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が、選択されていない場合は、それらを選択してください。
9. **オペレーティングシステム** タブをクリックして、オペレーティングシステムの各項目にデータが自動的に投入されていることを確認します。
10. コンピュートプロファイルから自動的にデータが投入されるイメージを変更するには、新規ホストの root ボリュームをベースとする別のイメージを **イメージ** リストから選択します。
11. **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。
12. **仮想マシン** タブをクリックして、これらの設定には、ホストグループおよびコンピュートプロファイルからの情報が入力されていることを確認し、必要に応じて変更してください。
13. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。
14. **送信** をクリックして変更を保存します。

この新規ホストのエントリは、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。

### CLI をご利用の場合

**hammer host create** コマンドでホストを作成し、**--provision-method image** オプションを指定してイメージベースのプロビジョニングを使用します。

```
# hammer host create --name "openstack-test1" --organization "My_Organization" \  
--location "New York" --hostgroup "Example_Hostgroup" \  
--compute-resource "My_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=default,network=mynetwork"
```

このコンピュートリソースに対する他のホスト作成パラメーターの情報は、「[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。



## 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 を使用すると、Amazon EC2 のパブリック API で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。本章の手順を使用して、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

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。詳細は、『[コンテンツ管理ガイド](#)』の「[Red Hat リポジトリの同期](#)」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) を使用します。
- イメージベースのプロビジョニング用の Amazon Machine Image (AMI)
- ホスト登録用のアクティベーションキー。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

この手順を使用して、Satellite Server のコンピュートリソースに Amazon EC2 接続を追加します。

#### 時間設定および Amazon Web Services

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。Satellite Server の時間が正常に同期されていること、さらに、`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。時間の同期に関する詳細は、『[インストールガイド](#)』の「[時間の同期](#)」を参照してください。

#### 手順

Amazon EC2 接続を追加するには、以下の手順を実行します。

1. 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](#) に移動して、コンピュートリソースのウィンドウで [コンピュートリソースの作成](#) をクリックします。
2. **名前** フィールドには、Amazon EC2 コンピュートリソースを識別するための名前を入力します。
3. **プロバイダー** の一覧から **EC2** を選択します。
4. **説明** フィールドには、今後使用する時にリソースを特定できるように名前を入力します。
5. **HTTP プロキシ** リストから、外部 API サービスに接続する HTTP プロキシを選択します。このリストからプロキシを選択する前に、HTTP プロキシを追加する必要があります。詳細は、『[接続ネットワークからの Satellite Server のインストール](#)』の「[HTTP プロキシでの Satellite Server の設定](#)」を参照してください。

6. **アクセスキー** と **シークレットキー** フィールドで、Amazon EC2 アカウントのアクセスキーを入力します。詳細情報は、Amazon ドキュメントの Web サイトで [「Managing Access Keys for your AWS Account」](#) を参照してください。
7. **リージョンのロード (Load Regions)** ボタンをクリックして **リージョン** リストにデータを入力します。
8. **リージョン** リストから、使用する Amazon EC2 リージョンとデータセンターを選択します。
9. **ロケーション** タブをクリックして、使用するロケーションが選択されていることを確認します。あるいは、別のロケーションを追加します。
10. **組織** タブをクリックして、使用する組織が選択されていることを確認します。あるいは、別のロケーションを追加します。
11. **送信** をクリックして Amazon EC2 接続を保存します。

## CLI をご利用の場合

**hammer compute-resource create** コマンドで接続を作成します。--user および --password オプションを使用して、アクセスキーとシークレットキーをそれぞれ追加します。

```
# hammer compute-resource create --name "My_EC2" --provider "EC2" \  
--description "Amazon EC2 Public Cloud" --user "user_name" \  
--password "secret_key" --region "us-east-1" --locations "New York" \  
--organizations "My_Organization"
```

## 11.3. コンピュートリソースと HTTP SMART PROXY との連携

使用する EC2 コンピュートリソースで、Satellite との通信に特定の HTTP プロキシが必要となる場合があります。Satellite では、HTTP プロキシエントリを作成して、そのプロキシをお使いの EC2 コンピュートリソースに割り当てることができます。

ただし、**管理 > 設定** で Satellite のプロキシを設定してから、お使いのコンピュートリソース用にプロキシを追加した場合には、**管理 > 設定** で定義したプロキシが優先されます。

### 手順

コンピュートリソースに割り当て可能な HTTP プロキシを追加するには、以下の手順を実行します。

1. Satellite Web UI で、**インフラストラクチャー > HTTP Capsule** に移動して、**新規 HTTP プロキシ** をクリックします。
2. **Name** フィールドには、プロキシ名を入力します。
3. **URL** フィールドには、ポート番号を含めてプロキシの URL を入力します。
4. オプション: プロキシに認証が必要な場合には、プロキシに対する認証に使用するユーザーとパスワードを入力します。
5. **テスト接続** をクリックして Satellite からプロキシに対する接続できることを確認します。
6. **ロケーション** タブで、ロケーションを追加します。
7. **組織** タブをクリックして、組織を追加します。
8. **送信** をクリックします。

## 11.4. SATELLITE SERVER への 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。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。

### 手順

1. 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、Amazon EC2 接続を選択します。
2. **イメージ** タブをクリックして、**新規イメージ** をクリックします。
3. **名前** フィールドには、今後使用する時にイメージを特定できるように名前を入力します。
4. **オペレーティングシステム** リストから、追加するイメージに対応するオペレーティングシステムを選択します。
5. **アーキテクチャー** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
7. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
8. **イメージ ID** フィールドには、イメージの Amazon Machine Image (AMI) ID を入力します。通常、この形式は **ami-xxxxxxx** になります。
9. **ユーザーデータ** リストから、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、Finish 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。Finish 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。
10. **IAM ロール** リストから、イメージの作成に使用する Amazon のセキュリティーロールを選択します。
11. **送信** をクリックしてイメージの詳細を保存します。

### CLI をご利用の場合

**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-my_ami_id" --compute-resource "My_EC2"
```

## 11.5. AMAZON EC2 の詳細のコンピュータプロファイルへの追加

Amazon EC2 のインストールのハードウェア設定をコンピュータプロファイルに追加します。

### 手順

ハードウェア設定を追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー > コンピュートプロファイル** に移動して、プロファイルの名前をクリックして EC2 接続の名前をクリックします。
2. **フレーバー** リストから、ホストに使用する EC2 のハードウェアのプロファイルを選択します。
3. **イメージ** リストから、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を選択します。
4. **利用可能ゾーン (Availability zone)** リストから、EC2 リージョン内で使用するターゲットクラスターを選択します。
5. **サブネット** リストから EC2 インスタンスのサブネットを追加します。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。
6. **セキュリティグループ** リストから、ホストに適用するポートおよび IP アドレスのクラウドベースのアクセスルールを定義します。
7. **管理 IP** リストから、**パブリック IP** または **プライベート IP** のいずれかを選択します。
8. **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 をご利用の場合

コンピュートプロファイルの CLI コマンドは、Red Hat Satellite 6.5 ではまだ実装されていません。代わりに、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。

## 11.6. 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。

### 手順

1. Satellite web UI で、**ホスト > 新規ホスト** に移動します。
2. **名前** フィールドには、ホストの名前を入力します。
3. **ホストグループ** リストから、ホストグループを選択して、新規ホストのフィールドの大部分にデータを投入することができます。
4. **デプロイ先** リストから、EC2 接続を選択します。
5. **コンピュートプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
6. **インターフェース** タブをクリックしてから、ホストのインターフェースの **編集** をクリックし、フィールドに値が投入されていることを確認します。**Mac アドレス** のフィールドは空白にします。Satellite Server は自動的に、ホストの最初のインターフェースの IP アドレスと **Managed, Primary** および **Provision** オプションを選択します。
7. **オペレーティングシステム** タブをクリックして、全フィールドに値が投入されていることを確認します。
8. **仮想マシン** タブをクリックして、全フィールドに値が投入されていることを確認します。
9. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。提供されていない場合には、アクティベーションキーを追加します。

10. **送信** をクリックして変更を保存します。

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。

### CLI をご利用の場合

**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。

```
# hammer host create --name "ec2-test1" --organization "My_Organization" \
--location "New York" --hostgroup "Base" \
--compute-resource "My_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"
```

このコンピュートリソースのホスト作成パラメーターに関する情報は、[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。

## 11.7. SSH での AMAZON EC2 インスタンスへの接続

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が、Red Hat Satellite を介してプロビジョニングする 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、使用する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
```

## 11.8. AMAZON WEB SERVICE EC2 環境向け FINISH 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 finish テンプレートを使用できます。

Amazon EC2 向けの finish 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、使用する 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、**finish** テンプレート リストから、finish 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、作成する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、**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。

## 11.9. 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。

## 第12章 コンテナのプロビジョニング

Red Hat Satellite 6 では、オンプレミスレジストリーを作成して複数のソースからイメージをインポートし、コンテンツビューを使用してこれらのイメージをコンテナに配信することができます。Satellite Server は、コンテナ実行用のサーバーとして機能する1つ以上の Docker コンピュートリソースの作成をサポートします。

イメージのインポート、このイメージをベースにしたコンテナの起動、コンテナのアクティビティの監視、さらに伝搬可能な新規イメージ階層への状態のコミットが可能です。

コンテナに関する情報は、Red Hat Enterprise Linux Atomic Host 7の「[Getting Started with Containers](#)」ガイドを参照してください。

コンテナイメージのインポートやこれらのイメージを使用したコンテナの作成については、この章をご利用ください。

### 12.1. コンテナプロビジョニングの要件

Red Hat Enterprise Linux Atomic Host を使用してコンテナをプロビジョニングする前に、コンテナレジストリーなど、イメージのソースが必要です。

Red Hat Satellite 6 はコンテナイメージの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 形式のコンテナイメージ。
- Docker Hub からのパブリックイメージ。
- Red Hat のコンテナ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. コンテナイメージのインポート

Red Hat Container Catalog または Satellite に接続する他のイメージレジストリーからコンテナイメージリポジトリーをインポートできます。

以下の手順はリポジトリー検出を使用して、インポートするコンテナイメージを検索します。製品およびリポジトリーを手動で作成する情報は、『[コンテンツ管理ガイド](#)』の「[カスタム製品の作成](#)」を参照してください。

#### 手順

コンテナイメージリポジトリーをインポートして、製品を作成するか、製品と関連付けるには、以下の手順を実行します。

1. Satellite Web UI で **コンテンツ** > **製品** に移動して、**リポジトリーの作成** をクリックします。
2. **リポジトリータイプ** リストから **コンテナイメージ** を選択します。
3. **検出するレジストリー** フィールドには、イメージのインポート元となるレジストリーの URL を入力します。
4. **レジストリーのユーザー名** フィールドには、コンテナのイメージレジストリーのユーザー名に対応する名前を入力します。



5. **レジストリーのパスワード** フィールドには、入力したユーザ名に対応するパスワードを入力します。
6. **レジストリー検索パラメーター** フィールドには、検索の絞り込みに使用する検索条件を入力して、**検出** をクリックします。
7. オプション: **検出されたリポジトリ** リストをさらに絞り込むには、**フィルター** フィールドに、使用する追加の検索条件を入力します。
8. **検出されたリポジトリ** リストからインポートするリポジトリを選択して、**選択項目の作成** をクリックします。
9. オプション: **製品** リストから製品を作成するには、**新しい製品** を選択します。
10. **名前** フィールドに製品名を入力します。
11. オプション: **リポジトリ名** と **リポジトリラベル** のコラムで、リポジトリ名とラベルを編集できます。
12. **リポジトリ作成の実行** をクリックします。
13. リポジトリの作成が完了したら、各新規リポジトリをクリックして詳細情報を確認できます。
14. オプション: リポジトリにインポートするコンテンツをフィルタリングするには、リポジトリをクリックして、**同期タグの制限** に移動します。これをクリックし、Satellite への同期コンテンツを制限するタグを編集または追加します。
15. **コンテンツ** > **製品** に移動し、製品名を選択します。
16. 新規リポジトリを選択し、**同期開始** をクリックして同期プロセスを開始します。

同期の進捗状況を表示するには、**コンテンツ** > **同期の状態** に移動して、リポジトリツリーを展開します。

同期が完了したら、**Docker マニフェストの管理** をクリックして利用可能なマニフェストを一覧表示します。また、必要のなくなったマニフェストは、このリストから削除できます。

## CLI をご利用の場合

1. カスタムの **Red Hat Container Catalog** 製品を作成します。

```
# hammer product create \
--name "Red Hat Container Catalog" \
--sync-plan "Example Plan" \
--description "Red Hat Container Catalog content" \
--organization "My_Organization"
```

2. コンテナイメージ用のリポジトリを作成します。

```
# hammer repository create \
--name "RHEL7" \
--content-type "docker" \
--url "http://registry.access.redhat.com/" \
```

```
--docker-upstream-name "rhel7" \
--product "Red Hat Container Catalog" \
--organization "My_Organization"
```

3. リポジトリを同期します。

```
# hammer repository synchronize \
--name "RHEL7" \
--product "Red Hat Container Catalog" \
--organization "My_Organization"
```

## 12.3. SATELLITE SERVER への外部レジストリーの追加

外部レジストリーのイメージからコンテナを作成する場合は、先に Satellite にレジストリーを追加する必要があります。

外部のコンテナレジストリーを追加するには、以下の手順を行います。

1. Satellite Web UI で、**コンテナ** > **レジストリー** に移動して、**レジストリーの作成** をクリックします。
2. **名前** フィールドには、レジストリーの名前を入力します。
3. **URL** フィールドには、<https://registry.access.redhat.com> などのレジストリーの場所を入力します。
4. オプション:**説明** フィールドには、レジストリーエントリーの説明を入力します。
5. **ユーザー名** フィールドには、レジストリーのユーザーアカウントに対応するユーザ名を入力します。
6. **パスワード** フィールドには、レジストリーのユーザーアカウントに対するパスワードを入力します。
7. **ロケーション** タブで、ロケーションを選択します。
8. **組織** タブで、組織を選択します。
9. **送信** をクリックして外部レジストリーを保存します。

### CLI をご利用の場合

**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 Container Image Registry" \
--organization "Default_Organization" \
--location "Default_Location"
```

## 12.4. コンテナ名のパターンの管理

Satellite を使用してコンテナの作成や管理を行う場合には、コンテナはコンテンツビューのバージョン間や異なるステージの Satellite ライフサイクル環境間を移動するので、コンテナ名はステージごとに変化します。たとえば、アップストリームのリポジトリから、**ssh** 名を使用してコンテナイ

イメージを同期する場合には、そのイメージを Satellite 製品と組織に追加してコンテンツビューの一部として公開する時に、コンテナイメージは **my\_organization\_production-custom\_spin-my\_product-custom\_ssh** という名前になる可能性があります。これが原因で、コンテナイメージをプルする時に問題が発生する可能性があります。理由は、コンテナレジストリーに、コンテナ名のインスタンスが1つしか含まれていない可能性があるためです。Satellite の命名規則の問題を回避するには、デフォルト名を上書きするようにレジストリー名のパターンを設定して、コンテナ名が後で使用するとき明確になるようにします。

## 制限事項

レジストリー名のパターンを使用してコンテナの命名規則を管理する場合には、レジストリーの命名パターンが理由でグローバル一意名を生成する必要があるため、命名時に競合の問題が発生する可能性があります。以下の例を示します。

- **repository.docker\_upstream\_name** のレジストリー名パターンを設定した場合に、**Production** ライフサイクルと同じレジストリー名のコンテナコンテンツが含まれるコンテンツビューを公開またはプロモートできません。
- **lifecycle\_environment.name** のレジストリー名パターンを設定した場合には、同じ名前を指定して2つ目のコンテナレジストリーを作成することができません。

コンテナに対してレジストリーの命名パターンを定義する場合には、注意を払って進めて行く必要があります。

## 手順

レジストリー名パターンで、コンテナの命名を管理するには、以下の手順を実行します。

1. Satellite Web UI で、**コンテンツ > ライフサイクル環境** に移動して、ライフサイクル環境を作成するか、編集するライフサイクル環境を選択します。
2. **コンテナイメージのレジストリー** エリアで、**レジストリー名のパターン** の右側にある編集アイコンをクリックします。
3. 変数一覧と例を使用して、必要とされるレジストリー名のパターンを判断します。
4. **レジストリー名のパターン** フィールドに、使用するレジストリー名のパターンを入力します。たとえば、**repository.docker\_upstream\_name** を使用するには、以下を入力します。

```
<%= repository.docker_upstream_name %>
```

5. **保存** をクリックします。

コンテナのレジストリーは即時に更新され、**docker pull** を使用して、指定の名前でコンテナイメージをプルできます。

## 12.5. SATELLITE でのコンテナイメージの管理

コンテンツビューでコンテナイメージを管理するには、以下の手順を行います。

1. Satellite web UI で **コンテンツ > コンテンツビュー** に移動して、**新規ビューの作成** をクリックします。
2. **名前** フィールドには、**コンテナ** を入力します。これにより、自動的に **ラベル** フィールドにデータが投入されます。

3. **説明** フィールドには、**Container image for Red Hat Enterprise Linux 7** など、説明を入力します。
4. 複合コンテンツビューを使用して他のコンテンツビューを使用する場合には、**複合ビュー** のチェックボックスを選択します。
5. オプション: **複合コンテンツビュー** を選択する場合は、新しいバージョンの複合コンテンツビューが作成された場合に、複合ビューを自動公開するかどうかを選択します。
6. **保存** をクリックして、コンテンツビューを作成します。
7. **Docker コンテンツ** のサブタブに移動して、**追加** をクリックします。
8. Red Hat Enterprise Linux 7 Server iイメージのコンテナーリポジトリを選択します。
9. **リポジトリの追加** をクリックします。
10. **バージョン** に移動して、**新規バージョンの公開** をクリックします。

バージョンの **説明** を入力します。わかりやすい説明を入力しておくことで、新しいコンテンツバージョンの記録をするのに役立ちます。

Satellite Server がビューの新バージョンを作成し、ライブラリー環境に公開します。

**プロモート** をクリックして、アプリケーションライフサイクルの環境でこのコンテンツビューをプロモートすることもできます。

### CLI をご利用の場合

1. リポジトリ ID 一覧を取得します。

```
# hammer repository list --organization "My_Organization"
```

2. コンテンツビューを作成して、リポジトリを追加します。

```
# hammer content-view create \
--name "Containers" \
--description "Container image for Red Hat Enterprise Linux 7" \
--repository-ids 8 \
--organization "My_Organization"
```

3. ビューを公開します。

```
# hammer content-view publish \
--name "Containers" \
--description "Initial Content View for our container image" \
--organization "My_Organization"
```

## 12.6. RED HAT ENTERPRISE LINUX ATOMIC HOST の設定

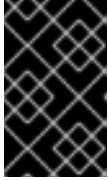
Docker 形式のコンテナー向けの Red Hat API を Satellite Server に公開するなど、Satellite に接続する前に Atomic Host を設定します。コンテナーに関する情報は、**Red Hat Enterprise Linux Atomic Host 7** の『[Getting Started with Containers](#)』ガイドを参照してください。

1. 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
```

2. **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

ポート 2375 または 2376 のいずれかを使用して接続します。Satellite Server には、これらのポートへのアクセスを許可する特定の SELinux ルールが含まれます。別のポートを使用すると、認証が失敗してしまいます。

3. ポート 2375 のファイアウォールを開放します。

```
# firewall-cmd --add-port=2375/tcp
# firewall-cmd --add-port=2375/tcp --permanent
```

4. 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
```

5. **docker** サービスを再起動します。

```
# systemctl restart docker
```

## 12.7. SATELLITE SERVER への ATOMIC HOST 接続の追加

Satellite Server のコンピュータリソースに Red Hat Enterprise Linux Atomic 接続を追加するには、以下の手順を行います。

1. Satellite Web UI で **インフラストラクチャー** > **コンピュータリソース** に移動して、**コンピュータリソースの作成** をクリックします。
2. **名前** フィールドに、リソースの名前を入力します。
3. **プロバイダー** の一覧から **Docker** を選択します。
4. オプション:**説明** フィールドには、リソースの説明を追加できます。
5. **ロケーション** タブで、使用するロケーションを選択します。
6. **組織** タブで、使用する組織を選択します。
7. **送信** をクリックして保存します。

### CLI をご利用の場合

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

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

## 12.8. コンテナの作成

この手順を使用して Satellite にコンテナを作成します。Satellite Web UI には、作成プロセスを順を追って進めるウィザードが含まれています。外部レジストリー、コンテンツビュー、Docker Hub から、Satellite に追加するコンテナイメージを選択できます。

1. Satellite web UI で、**コンテナ > コンテナの作成** に移動します。
2. **デプロイ先** リストから、使用するコンピュートリソースを選択します。
3. **ロケーション** タブで、ロケーションを選択します。
4. **組織** タブで、組織を選択します。
5. **次の手順** をクリックします。

コンテナを作成する方法には3つのオプションがあります。

- コンテンツビューのコンテナイメージからの作成
- Docker Hub のコンテナイメージからの作成
- 外部レジストリーのコンテナイメージからの作成

### コンテンツビューからのコンテナの作成

1. **ライフサイクル環境** リストから、使用するライフサイクル環境を選択します。
2. **コンテンツビュー** リストから、使用するコンテナイメージを選択します。
3. **レジストリー** リストから、使用するレジストリーを選択します。
4. **タグ** リストから、使用するタグを選択します。
5. **Capsule** リストから、使用する Capsule を選択します。
6. **次の手順** をクリックします。

### Docker Hub からのコンテナの作成

1. **Docker Hub** タブを選択して、**検索** フィールドに、使用する Docker コンテナを入力します。
2. **タグ** フィールドには、使用するコンテナイメージタグの名前を入力し、**イメージの検索** をクリックします。
3. 使用するコンテナイメージを選択して、**次の手順** をクリックします。

### 外部レジストリーからのコンテナの作成

1. **レジストリー** リストから、使用するレジストリーを選択します。

2. **検索** フィールドには、使用するコンテナイメージの名前を入力します。
3. **タグ** フィールドには、使用するコンテナイメージに関連付けられたタグを入力します。
4. **イメージの検索** をクリックして使用するイメージを選択し、**次の手順** をクリックします。

### コンテナ作成プロセスの完了

1. **名前** フィールドには、新規コンテナの名前を入力します。
2. **command** フィールドには、コンテナで実行するコマンドを入力します。
3. **エントリーポイント** フィールドには、コンテナの起動時に自動的にコンテナが実行するコマンドを入力します。デフォルトのエントリーポイントは、**/bin/sh -c** です。
4. **コンピュートオプション** タブを選択します。
5. **CPU セット** フィールドで、コンテナに CPU を割り当てます。たとえば、0-2,16 は CPU 0、1、2 および 16 を表します。
6. **CPU シェア** フィールドでは、コンテナの CPU シェアを割り当てます。これにより、コンテナ化されたタスクに提供される CPU 時間のシェアが設定されます。
7. **メモリー** フィールドには、コンテナに割り当てるメモリーサイズを入力します。
8. **次の手順** をクリックします。
9. **環境変数** フィールドには、**LANG=en\_US.UTF-8** などの環境変数セットを定義します。
10. **公開されたポート (Exposed Ports)** フィールドには、コンテナで開くポートの番号を入力します。たとえば、コンテナへの SSH 通信は、ポート 22 を開きます。
11. **DNS** フィールドには、コンテナの DNS サーバーを入力します。
12. **実行** のチェックボックスを選択して、作成後に自動的にコンテナを起動します。
13. **送信** をクリックして、コンテナを作成します。

### CLI をご利用の場合

以下では、**hammer docker container create** コマンドの例を 3 つ紹介しています。

- コンテンツビューからコンテナを作成します。

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

- 次に Docker Hub からプロビジョニングします。

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

- 外部レジストリーからプロビジョニングします。

■

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

## 12.9. コンテナレジストリーの認証管理

デフォルトでは、Satellite のイメージレジストリーのコンテナにアクセスするには認証が必要です。**docker login** コマンドを使用して Satellite にログインします。次に、**docker pull** と **docker search** コマンドで、Satellite パーミッションをもとにコンテナにアクセスできます。

ライフサイクル環境で管理する Satellite イメージレジストリーのコンテナイメージにアクセスするのにユーザーの認証をするかどうかを指定できます。たとえば、認証要件なしに **Production** ライフサイクルからコンテナイメージをプルできるようにしたり、認証済みのユーザーだけに **Development** および **QA** 環境へのアクセスを制限したりすることもできます。

### 手順

Satellite イメージレジストリーの認証設定を管理するには、以下の手順を実行します。

1. Satellite Web UI で、**コンテンツ > ライフサイクル環境** に移動して、認証管理するライフサイクル環境を選択します。
2. このライフサイクル環境に認証なしでアクセスできるようにするには、**非認証のプル** のチェックボックスを選択します。認証なしのアクセスを制限するには、**非認証のプル** チェックボックスのチェックを外します。
3. **保存** をクリックします。

## 12.10. コンテナの起動、コミット、削除

### コンテナの起動または停止

コンテナの作成時には、デフォルトの状態は無効となっています。コンテナを有効化して、コンピュータリソースでコンテナ化されたアプリケーションのプロセスを開始します。ホストは、Web アプリケーションの場合と同様にコンテナと通信できます。

1. Satellite web UI で、**コンテナ > すべてのコンテナ** に移動します。
2. 既存のコンテナリストから、任意のコンテナを選択し、**電源オン** をクリックします。コンテナを停止するには、**電源オフ** をクリックします。

### コンテナのコミット

イメージからコンテナを起動すると、書き込み可能な階層がこのイメージの上層に追加されます。コンテナをコミットすると、コンテナのステータスを保存するイメージ階層が作成されます。コンテナをコミットするたびに、変更を保存するために新しいイメージ階層が追加されます。たとえば、コンテナが Docker Hub からプルしたイメージをもとにしている場合には、コミットされた変更は Docker Hub にプッシュし直されます。

1. Satellite web UI で、**コンテナ > すべてのコンテナ** に移動します。
2. 既存のコンテナリストから、任意のコンテナを選択し、**コミット** をクリックします。
3. **user/my-rhel-image** などのリポジトリ名を入力します。



4. タグをイメージに割り当てます。
5. 連絡先情報を入力します。
6. コミットについて分かりやすいコメントを入力します。
7. **送信** をクリックします。

#### コンテナの削除

1. Satellite web UI で、**コンテナ** > **すべてのコンテナ** に移動します。
2. 既存のコンテナリストから、任意のコンテナを選択し、**削除** をクリックします。
3. 警告ウィンドウで **OK** をクリックして削除を確定します。

## 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』の例を使用していない場合は、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 {ProductVersion} (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
{ProductVersion} 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: <b>interface</b> 、 <b>bmc</b> 、または <b>bond</b>
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: <b>eth0</b>
managed	管理インターフェースのブール値。 <b>true</b> または <b>false</b> に設定します。
primary	プライマリ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インターフェースを設定する必要があります。 <b>true</b> または <b>false</b> 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。 <b>true</b> または <b>false</b> 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。 <b>true</b> または <b>false</b> 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: <b>eth1</b>

**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。 <b>balance-rr</b> 、 <b>active-backup</b> 、 <b>balance-xor</b> 、 <b>broadcast</b> 、 <b>802.3ad</b> 、 <b>balance-tlb</b> 、 <b>balance-alb</b> 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: <b>[eth1,eth2]</b>
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。 <b>IPMI</b> 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	<b>bridge</b> または <b>network</b> 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。 <b>virtio</b> 、 <b>rtl8139</b> 、 <b>ne2k_pci</b> 、 <b>pcnet</b> 、または <b>e1000</b> 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

パラメーター	説明
capacity	ボリュームの容量。例: <b>10G</b>
format_type	ディスクタイプ。 <b>raw</b> または <b>qcow2</b> 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: <b>eth0</b>
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。 <b>VirtualVmxnet</b> 、 <b>VirtualVmxnet2</b> 、 <b>VirtualVmxnet3</b> 、 <b>VirtualE1000</b> 、 <b>VirtualE1000e</b> 、 <b>VirtualPCNet32</b> 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。ホストをこ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 リポジトリのクローンを作成してから、リポジトリのディレクトリに切り替えます。

```

$ 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 モードに必要なパッケージ (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 Satellite 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 6.5 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。

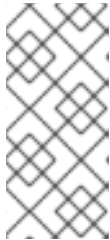
### D.3. RED HAT ENTERPRISE LINUX 6 イメージの作成

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 パスワードを設定します。
  - 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 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="My_Organization" \
--environment=Development/MyView \
--activationkey="MyKey"
```

### 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。

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する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する場所にイメージを移動します。