



# Red Hat Satellite 6.10

## 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 6.10 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 © 2023 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. プロビジョニングの概要	5
1.2. ネットワークブートのプロビジョニングワークフロー	5
<b>第2章 プロビジョニングリソースの設定</b> .....	<b>8</b>
2.1. プロビジョニングコンテキスト	8
2.2. プロビジョニングコンテキストの設定	8
2.3. オペレーティングシステムの作成	8
2.4. 複数のオペレーティングシステムの詳細更新	9
2.5. アーキテクチャーの作成	10
2.6. ハードウェアモデルの作成	11
2.7. 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	11
2.8. SATELLITE へのインストールメディアの追加	12
2.9. パーティションテーブルの作成	13
2.10. 動的パーティションの例	14
2.11. プロビジョニングテンプレート	15
2.12. プロビジョニングのタイプ	15
2.13. プロビジョニングテンプレートの作成	17
2.14. プロビジョニングテンプレートのクローン作成	17
2.15. コンピュートプロファイルの作成	18
2.16. ホストへのデフォルト暗号化 ROOT パスワードの設定	19
2.17. NOVNC を使用した仮想マシンへのアクセス	19
<b>第3章 ネットワークの設定</b> .....	<b>21</b>
3.1. ネットワークリソース	21
3.2. SATELLITE および DHCP オプション	22
3.3. SATELLITE で DHCP の問題をトラブルシューティングする手順	23
3.4. イメージベースのプロビジョニングの要件	25
3.5. ネットワークサービスの設定	25
3.6. SATELLITE SERVER へのドメインの追加	29
3.7. SATELLITE SERVER へのサブネットの追加	30
<b>第4章 DHCP および DNS プロバイダーとしての INFOBLOX の使用</b> .....	<b>32</b>
4.1. 制限事項	32
4.2. 前提条件	32
4.3. CAPSULE SERVER への INFOBLOX の CA 証明書のインストール	32
4.4. DHCP INFOBLOX モジュールのインストール	33
4.5. DNS INFOBLOX モジュールのインストール	34
<b>第5章 プロビジョニング時間を削減するための IPXE の設定</b> .....	<b>35</b>
5.1. 仮想マシンの起動	36
5.2. PXELINUX からの IPXE のチェーンブート	38
<b>第6章 PXE を使用したホストのプロビジョニング</b> .....	<b>41</b>
6.1. ベアメタルプロビジョニングの前提条件	41
6.2. セキュリティートークンの有効期間の設定	42
6.3. 無人プロビジョニングによるホストの作成	42
6.4. UEFI HTTP ブートプロビジョニングを使用したホストの作成	43
6.5. プロビジョニング中の SSH キーのデプロイ	46
<b>第7章 DISCOVERY サービスの設定</b> .....	<b>47</b>
7.1. DISCOVERY サービスのインストール	48
7.2. テンプレート PXELINUX DISCOVERY スニペットのプロビジョニング	48

7.3. 検出されたホストの自動コンテキスト	49
7.4. DISCOVERY テンプレートとスニペットの設定	49
7.5. 検出されたホストからのホストの作成	50
7.6. DISCOVERY ルールの作成	52
7.7. PXE を使用しない DISCOVERY の実装	54
7.8. 無人での使用、カスタマイズ、イメージのリマスター	55
7.9. DISCOVERY イメージの拡張	57
7.10. DISCOVERY のトラブルシューティング	58
<b>第8章 RED HAT IMAGE BUILDER イメージを使用したプロビジョニング</b>	<b>60</b>
<b>第9章 KVM (LIBVIRT) での仮想マシンのプロビジョニング</b>	<b>61</b>
9.1. SATELLITE SERVER の KVM 接続設定	62
9.2. SATELLITE SERVER への KVM 接続の追加	62
9.3. SATELLITE SERVER での KVM イメージの追加	63
9.4. KVM の詳細のコンピュートプロファイルへの追加	64
9.5. KVM でのホストの作成	65
<b>第10章 RED HAT VIRTUALIZATION での仮想マシンのプロビジョニング</b>	<b>68</b>
10.1. SATELLITE SERVER への RED HAT VIRTUALIZATION 接続の追加	69
10.2. RED HAT VIRTUALIZATION での CLOUD-INIT イメージの準備	70
10.3. SATELLITE SERVER への RED HAT VIRTUALIZATION イメージの追加	70
10.4. CLOUD-INIT テンプレートの準備	71
10.5. RED HAT VIRTUALIZATION の詳細のコンピュートプロファイルへの追加	73
10.6. RED HAT VIRTUALIZATION でのホストの作成	74
<b>第11章 VMWARE VSPHERE での仮想マシンのプロビジョニング</b>	<b>77</b>
11.1. VMWARE VSPHERE プロビジョニングの要件	77
11.2. VMWARE VSPHERE ユーザーの作成	77
11.3. SATELLITE SERVER への VMWARE VSPHERE 接続の追加	78
11.4. SATELLITE SERVER への VMWARE VSPHERE イメージの追加	79
11.5. VMWARE VSPHERE の詳細のコンピュートプロファイルへの追加	80
11.6. VMWARE VSPHERE サーバーでのホストの作成	81
11.7. プロビジョニングでの VMWARE VSPHERE CLOUD-INIT と USERDATA テンプレートの使用	83
11.8. コンピュートリソースのキャッシング	87
<b>第12章 CONTAINER-NATIVE VIRTUALIZATION を使用した仮想マシンのプロビジョニング</b>	<b>89</b>
12.1. SATELLITE SERVER への CONTAINER-NATIVE VIRTUALIZATION 接続の追加	90
<b>第13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>92</b>
13.1. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM 接続の追加	92
13.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM イメージの追加	93
13.3. RED HAT OPENSTACK PLATFORM の詳細のコンピュートプロファイルへの追加	94
13.4. RED HAT OPENSTACK PLATFORM でのイメージベースのホストの作成	95
<b>第14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>97</b>
14.1. AMAZON EC2 プロビジョニングの要件	97
14.2. SATELLITE SERVER への AMAZON EC2 接続の追加	97
14.3. コンピュートリソースと HTTP プロキシとの連携	98
14.4. SATELLITE SERVER への AMAZON EC2 イメージの追加	99
14.5. AMAZON EC2 の詳細のコンピュートプロファイルへの追加	100
14.6. AMAZON EC2 でのイメージベースのホストの作成	100
14.7. SSH での AMAZON EC2 インスタンスへの接続	101
14.8. AMAZON WEB SERVICE EC2 環境向け FINISH テンプレートの設定	102
14.9. AMAZON WEB SERVICES と SATELLITE に関する詳細情報	103

---

<b>第15章 GOOGLE COMPUTE ENGINE でのクラウドインスタンスのプロビジョニング</b> .....	<b>104</b>
15.1. SATELLITE SERVER への GOOGLE COMPUTE ENGINE 接続の追加	104
15.2. SATELLITE SERVER への GOOGLE COMPUTE ENGINE イメージの追加	106
15.3. GOOGLE COMPUTE ENGINE の詳細のコンピューティングプロファイルへの追加	106
15.4. GOOGLE COMPUTE ENGINE でのイメージベースのホストの作成	107
<b>第16章 MICROSOFT AZURE RESOURCE MANAGER でのクラウドインスタンスのプロビジョニング</b> .....	<b>110</b>
16.1. MICROSOFT AZURE RESOURCE MANAGER 接続の SATELLITE SERVER への追加	111
16.2. MICROSOFT AZURE RESOURCE MANAGER イメージの SATELLITE SERVER への追加	112
16.3. MICROSOFT AZURE RESOURCE MANAGER の詳細のコンピューティングプロファイルへの追加	113
16.4. MICROSOFT AZURE RESOURCE MANAGER でのイメージベースのホスト作成	115
<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>118</b>
<b>付録B FIPS 準拠ホストのプロビジョニング</b> .....	<b>120</b>
B.1. 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	120
B.2. FIPS 有効化パラメーターの設定	120
B.3. FIPS モードの有効化の確認	121
<b>付録C RED HAT SATELLITE 向けクラウドイメージの構築</b> .....	<b>122</b>
C.1. RED HAT ENTERPRISE LINUX のカスタムイメージの作成	122
C.2. RED HAT ENTERPRISE LINUX 7 イメージの作成	123
C.3. RED HAT ENTERPRISE LINUX 6 イメージの作成	124
C.4. ホスト登録の設定	125
C.5. ホストの登録	126
C.6. KATELLO エージェントのインストール	127
C.7. PUPPET エージェントのインストール	128
C.8. RED HAT ENTERPRISE LINUX 7 イメージの完了	128
C.9. RED HAT ENTERPRISE LINUX 6 イメージの完了	129
C.10. 次のステップ	130





# 第1章 はじめに

## 1.1. プロビジョニングの概要

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 または外部 Capsule Server は、PXE ベースの方法を使用してベアメタルホスト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 アプリケーションライフサイクルに含まれます。

Red Hat Satellite では、さまざまな方法で、ホストをプロビジョニングできます。

### ベアメタルプロビジョニング

Satellite は、主に PXE ブートおよび 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。

### Cloud Providers

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。

## 1.2. ネットワークブートのプロビジョニングワークフロー

PXE ブートは、物理または仮想ホストの起動デバイスの順序として、1番目がネットワーク、2番目がハードドライブに設定されていることが前提です。

プロビジョニングプロセスは、基本的な PXE のワークフローに従います。

1. ホストを作成して、ドメインとサブネットを選択します。Satellite は、サブネットに関連付けられている DHCP Capsule Server、または Satellite の PostgreSQL データベースから、利用可能な IP アドレスを要求します。次に、Satellite は、この IP アドレスを Create Host ウィンドウの **IP address** フィールドに投入します。新しいホストのオプションをすべて入力したら、新しいホストリクエストを送信します。
2. ホストとそのドメインおよびサブネットの設定仕様に応じて、Satellite は次の設定を作成します。
  - サブネットに関連付けられた Capsule Server の DHCP レコード
  - ドメインに関連付けられた Capsule Server の正引き DNS レコード
  - サブネットに関連付けられた DNS Capsule Server の逆引き DNS レコード

- サブネットが関連付けられた TFTP Capsule Server にあるホストの PXELinux、Grub、Grub2、iPXE 設定ファイル
  - 関連する Puppet サーバーの Puppet 証明書
  - 関連付けられた ID サーバーのレルム
3. 新規ホストが DHCP サーバーから DHCP 予約を要求します。
  4. DHCP サーバーは予約要求に応答し、TFTP の **next-server**、**filename** オプションを返します。
  5. ホストは、PXELoader の設定をもとに、TFTP サーバーからブートローダーやメニューを要求します。
  6. ブートローダーは TFTP 経由で返されます。
  7. ブートローダーはプロビジョニングインターフェイスの MAC アドレスを介してホストの設定を取得します。
  8. ブートローダーはオペレーティングシステムのインストーラーカーネル、初期 RAM ディスク、およびブートパラメーターを取得します。
  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を要求します。
  10. Satellite はプロビジョニングテンプレートをレンダリングし、結果をホストに返します。
  11. インストーラーはオペレーティングシステムをインストールします。
    - インストーラーは Red Hat Subscription Manager を使用してホストを Satellite に登録します。
    - インストーラーは **katello-agent** や **puppet** などの管理ツールをインストールします。
    - 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。
  12. PXE 設定ファイルはローカルブートテンプレートに戻ります。
  13. ホストは再起動します。
  14. 新規ホストが DHCP サーバーから DHCP 予約を要求します。
  15. DHCP サーバーは予約要求に応答し、TFTP の **next-server**、**filename** オプションを返します。
  16. ホストは、PXELoader の設定をもとに、TFTP サーバーからブートローダーやメニューを要求します。
  17. ブートローダーは TFTP 経由で返されます。
  18. ブートローダーはプロビジョニングインターフェイスの MAC アドレスを介してホストの設定をフェッチします。
  19. ブートローダーはローカルドライブから起動を開始します。
  20. 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。

## 検出

検出サービスを使用している場合は、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に、ホストのタッチ先の 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て、ホストの USB ポートを使用して転送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

Satellite は仮想マシンを作成して、MAC アドレスを取得し、Satellite にその MAC アドレスを保存します。イメージベースのプロビジョニングを使用する場合は、ホストは、標準の PXE ブートやオペレーティングシステムのインストールを行うわけではありません。コンピュータリソースは、使用するホストのイメージのコピーを作成します。Satellite のイメージ設定に合わせて、**cloud-init** を使用するなど、シードデータを初期設定用に渡すことができます。Satellite は、SSH を使用してホストに接続し、テンプレートを実行してカスタマイズを完了できます。



## 注記

デフォルトでは、プロビジョニングされたプロファイルホストを Satellite から削除しても、外部コンピューティングリソース上の実際の仮想マシンは破棄されません。Satellite でホストエントリを削除するときに仮想マシンを破棄するには、**Administer > Settings > Provisioning** に移動し、**destroy\_vm\_on\_host\_delete** 設定を使用してこの動作を設定します。関連付けられた仮想マシンを破棄せず、後で同じリソース名で新しい仮想マシンを作成しようとする、その仮想マシン名が外部コンピューティングリソースにすでに存在するため、失敗します。すでにプロビジョニングされているホストに使用する標準のホスト登録ワークフローを使用して、既存の仮想マシンを Satellite に登録できます。

## 第2章 プロビジョニングリソースの設定

### 2.1. プロビジョニングコンテキスト

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、セキュリティー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 組織の管理](#) を参照してください。

ロケーションは組織と同様に機能します。相違点は、ロケーションが物理的な設定または地理的な設定をもとにしている点です。ユーザーは階層でロケーションをネスト化できます。ロケーションの詳細は、[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

プロビジョニング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

1. **任意の組織** タブで、組織を選択します。
2. **任意のロケーション** をクリックして、使用する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 手順

- 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。

### 2.3. 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り

ソースを組み合わせます。

Red Hat の CDN からオペレーティングシステムをインポートすると、**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。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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の候補を選択します。
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。詳細は、[Satellite へのインストールメディアの追加](#) 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 手順

- **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"
```

## 2.4. 複数のオペレーティングシステムの詳細更新

この手順を使用して複数のオペレーティングシステムの詳細を更新します。以下の例では、各オペレーティングシステムに、**Kickstart default** と呼ばれるパーティションテーブル、**Kickstart default PXELinux** と呼ばれる設定テンプレート、**Kickstart Default** と呼ばれるプロビジョニングテンプレートを割り当てる方法を紹介しています。

## 手順

1. Satellite Server で以下の Bash スクリプトを実行します。

```
PARTID=$(hammer --csv partition-table list | grep "Kickstart default," | cut -d, -f1)
PXEID=$(hammer --csv template list --per-page=1000 | grep "Kickstart default PXELinux" |
cut -d, -f1)
SATID=$(hammer --csv template list --per-page=1000 | grep "provision" | grep ",Kickstart
default" | cut -d, -f1)

for i in $(hammer --no-headers --csv os list | awk -F, {'print $1'})
do
    hammer partition-table add-operatingsystem --id="{PARTID}" --operatingsystem-id="{i}"
    hammer template add-operatingsystem --id="{PXEID}" --operatingsystem-id="{i}"
    hammer os set-default-template --id="{i}" --config-template-id={PXEID}
    hammer os add-config-template --id="{i}" --config-template-id={SATID}
    hammer os set-default-template --id="{i}" --config-template-id={SATID}
done
```

2. 更新したオペレーティングシステムの情報を表示して、オペレーティングシステムが正しく更新されたことを確認します。

```
# hammer os info --id 1
```

## 2.5. 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 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 [Supported architectures and provisioning scenarios in Satellite 6](#) を参照してください。

## 手順

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. **Submit** をクリックします。

## CLI手順

- **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"
```

## 2.6. 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

1. Satellite Web UI で **ホスト > ハードウェアモデル** に移動して、**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手順

- **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"
```

## 2.7. 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、同期済みのキックスタートリポジトリのセットが含まれており、これは、プロビジョニングしたホストのオペレーティングシステムをインストールする時に使用します。リポジトリの追加に関する詳細は、[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。

この手順を使用して、キックスタートリポジトリを設定します。

### 前提条件

プロビジョニングする前に、**BaseOS** と **Appstream Kickstart** の両方を有効にする必要があります。

### 手順

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。

Red Hat Enterprise Linux 8 の場合は、**Red Hat Enterprise Linux 8 for x86\_64 - AppStream Kickstart x86\_64 8** と **Red Hat Enterprise Linux 8 for x86\_64 - BaseOS Kickstart x86\_64 8** の両方のリポジトリが追加されていることを確認します。

オフライン環境をご利用の場合は、Red Hat Enterprise Linux バイナリー DVD からキックスタートリポジトリをインポートする必要があります。詳細は、[コンテンツ管理ガイドのキックスタートリポジトリのインポート](#) を参照してください。

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。詳細は、[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** のチェックボックスを選択します。

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"
```

## 2.8. SATELLITE へのインストールメディアの追加

インストールメディアは、Satellite Server が外部リポジトリからマシン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ために使用するパッケージのソースです。このパラメーターを使用して、サードパーティーのコンテンツをインストールできます。Red Hat コンテンツは、リポジトリの同期により配信されます。

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。

デフォルトでは Satellite には公式な Linux ディストリビューションのインストールメディアが含まれています。これらのインストールメディアは特定のバージョンのオペレーティングシステムを対象としている点に注意してください。たとえば、**CentOS mirror (7.x)** は CentOS 7 以前に、**CentOS mirror (8.x)** は CentOS 8 以降に使用する必要がある点に注意してください。

インストールメディアを使用して複数の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時のダウンロードパフォーマンスを向上させたい場合は、最も近いミラーまたはローカルコピーを参照するようにインストールメディアの **パス** を変更する必要があります。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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手順

- **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"
```

## 2.9. 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、テンプレート的一种で、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します。パーティションテーブルは、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。Red Hat Satellite 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作成を行い、そのエントリをオペレーティングシステムのエントリに追加することができます。

Web UI の代わりに CLI を使用する場合は、[CLI手順](#) を参照してください。

### 手順

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の作成** をクリックします。
2. **名前** フィールドに、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必要がある場合は、**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。以下に例を示します。

```
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。Satellite 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 手順

1. 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を含むプレーンテキスト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。
2. **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"
```

## 2.10. 動的パーティションの例

次のセクションでは、Anaconda キックスタートテンプレートを使用して、ディスク全体を消去し、自動的にパーティションを作成して1つのパーティションを最大サイズに拡大してから、プロビジョニングプロセスの次の一連のイベントに進むように Anaconda に指示します。

```
zerombr
clearpart --all --initlabel
autopart <%= host_param('autopart_options') %>
```

動的パーティション設定は、インストールプログラムにより実行されます。そのため、独自のルールを記述して、ディスクサイズ、ドライブの数、ベンダー、製造メーカーなど、ノードからのランタイム情報に基づいて、ディスクのパーティション方法を指定できます。

サーバーをプロビジョニングして動的パーティショニングを使用する場合には、以下の例をテンプレートとして追加します。**#Dynamic** エントリーが追加されると、テンプレートの内容が **%pre** シェルスクリプトレットに読み込まれ、`/tmp/diskpart.cfg` が作成され、キックスタートのパーティションのセクションに追加されます。

```
#Dynamic (do not remove this line)

MEMORY=$((`grep MemTotal: /proc/meminfo | sed 's/^MemTotal: */'|sed 's/ .*//'` / 1024))
if [ "$MEMORY" -lt 2048 ]; then
    SWAP_MEMORY=$((MEMORY * 2))
elif [ "$MEMORY" -lt 8192 ]; then
    SWAP_MEMORY=$MEMORY
elif [ "$MEMORY" -lt 65536 ]; then
    SWAP_MEMORY=$((MEMORY / 2))
else
    SWAP_MEMORY=32768
fi

cat <<EOF > /tmp/diskpart.cfg
zerombr yes
clearpart --all --initlabel
part /boot --fstype ext4 --size 200 --asprimary
part swap --size "$SWAP_MEMORY"
part / --fstype ext4 --size 1024 --grow
EOF
```

## 2.11. プロビジョニングテンプレート

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。

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、テンプレートの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。詳細は、**ホストの管理** ガイドの [テンプレートリポジトリの同期](#) を参照してください。

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。さらに前の変更へ戻ることもできます。**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。

- **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せん。
- **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。
- **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。

## 2.12. 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 プロビジョニングの場合は、**PXELinux** テンプレートを選択します。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、SSH 接続を使用して実行するプロビジョニング後の設定スクリプト。finish テンプレートは、user\_data をサポートしない仮想またはクラウド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 イメージと呼ばれる場合もあるので、Foreman Discovery ISO のイメージと混同しないようにしてください。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。

Red Hat Satellite は、終了スクリプトがリターンコード **0** で正常に終了すると、このコードが成功したとして処理し、ホストはビルドモードを終了します。終了スクリプトには、**call back** HTTP 呼び出しを使用するビルドモードを含むものが複数ある点に注意してください。上記のスクリプトは、イメージベースのプロビジョニングには使用されず、Debian、Ubuntu、BSD などのオペレーティングシステムをインストールした後の設定に使用されます。

## user\_data

シードデータとも呼ばれるカスタムデータに対応するプロバイダーのプロビジョニング後の設定スクリプト。user\_data テンプレートを使用してクラウドまたは仮想環境のみで仮想マシンをプロビジョニングできます。このテンプレートでは、Satellite がホストに到達できなくても構いません。クラウドまたは仮想化プラットフォームがデータをイメージに配信します。

プロビジョニングするイメージに、データを読み取るためのソフトウェアがインストールされており、そのソフトウェアが起動時に開始されるように設定されていることを確認します。たとえば、**cloud-init** は YAML 入力が必要で、また、**ignition** は JASON 入力が必要です。

## cloud\_init

VMWare などの一部の環境ではカスタムデータをサポートしないか、独自のデータ形式を使用しておりカスタマイズ時の機能に制限がある場合があります。このような場合には、**foreman** プラグインで cloud-init クライアントを設定し、HTTP または HTTPS で Satellite から直接テンプレートのダウンロードを試行します。この手法はどの環境でも利用できますが、仮想化環境での利用を推奨します。

**cloud\_init** テンプレートを使用する以下の要件を満たしていることを確認してください。

- プロビジョニングするイメージに、データを読み取るためのソフトウェアがインストールされており、そのソフトウェアが起動時に開始されるように設定されていることを確認します。
- プロビジョニングされたホストは、ホストのプロビジョニングインターフェイスの IP と一致する IP アドレスから Satellite に到達できます。  
cloud-init は NAT の前では機能しないことに注意してください。

## Bootdisk

PXE 以外の起動方法に使用するテンプレート

### カーネル実行 (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おらず、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。ですが、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で、お客様に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。

### スクリプト

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。

## 2.13. 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。この手順を使用して、新規のプロビジョニングテンプレートを作成します。

### 手順

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、**テンプレートの作成** をクリックします。
2. **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。
3. 必要に応じて残りのフィールドに入力します。**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、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。

### CLI 手順

1. テンプレートを含むプレーンテキストファイルを作成してから、CLI でテンプレートを作成します。この例では、**~/my-template** ファイルを使用しています。
2. **hammer template create** コマンドを使用してテンプレートを作成し、**--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"
```

## 2.14. プロビジョニングテンプレートのクローン作成

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。この手順を使用して、テンプレートのクローンを作成して、更新をこのクローンに追加します。

## 手順

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. **クローン** をクリックして、テンプレートを複製します。
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を入力します。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。
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** の一覧から、テンプレートのタイプを選択します。(例: **プロビジョニングテンプレート**)
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。

## 2.15. 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を使用して、CPU、メモリー、ストレージなどの仮想マシンハードウェアの詳細を事前定義できます。Red Hat Satellite のデフォルトのインストールには、3つの定義済みプロファイルが含まれています。

- **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、**コンピュープロファイルの作成** をクリックします。
2. **Name** フィールドには、プロファイル名を入力します。
3. **Submit** をクリックします。新しいウィンドウが開き、コンピューティングプロファイルの名前が表示されます。
4. 新しいウィンドウで、各コンピュートリソースの名前をクリックし、このコンピューティングプロファイルに設定する属性を編集します。

### CLI手順

Red Hat Satellite 6.10 には、コンピュープロファイルの CLI コマンドは実装されていません。

## 2.16. 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。

### 手順

1. 暗号化されたパスワードを生成します。

```
# 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. 暗号化パスワードを貼り付け、**保存** をクリックします。

## 2.17. 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
- Red Hat Virtualization

### 前提条件

- Satellite で仮想マシンを作成しておく必要がある。
- 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する。

- Satellite Server に Katello root CA 証明書をインポートする必要がある。ブラウザー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。詳細は、**Red Hat Satellite の管理ガイド**の [Katello ルート 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3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 上に統合された Capsule でネットワークサービスを設定します。

新規ホストは、Capsule Server にアクセスできる必要があります。Capsule Server は、Satellite Server 上に統合された Capsule、または外部 Capsule Server のいずれかになります。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 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推奨します。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これには、以下のものが含まれます。
  - 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イ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 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での Satellite Server の設定](#)を参照してください。

### 3.1. ネットワークリソース

Satellite には、ホストの作成に必要なネットワークリソースが含まれます。Satellite には、以下のネットワークリソースが含まれます。

#### ドメイン

Satellite で管理するホストはすべてドメインに割り当てる必要があります。ドメインを使用して、Satellite は A、AAAA および PTR レコードを管理します。Satellite で DNS サーバーを管理しない場合でも、ドメイン 1 つ以上を作成して、関連付ける必要があります。ドメインは、Satellite がホストする命名規則に含めます。たとえば、**example.com** ドメインで名前が **test123** のホストは、**test123.example.com** という完全修飾ドメイン名になります。

#### サブネット

Satellite が管理するホストはすべてサブネットに割り当てる必要があります。サブネットを使用することで、Satellite は IPv4 予約を管理できます。予約機能が統合されていない場合でも、最低でもサブネット 1 つを作成して関連付ける必要があります。Satellite でサブネットを管理すると、Satellite の外にあるサブネットの DHCP レコードを作成できません。Satellite では、IP Address Management (IPAM) を使用して、以下のオプションのいずれかで IP アドレスを管理できます。

- **DHCP**: DHCP Capsule は、範囲の最初のアドレスから開始し、予約されているアドレスをすべて飛ばして、次に利用可能な IP アドレスを検索することで、IP アドレスの割り当てを管理します。Capsule は IP アドレスを割り当てる前に、ICMP と TCP に Ping を送信して、IP アドレスが使用中かどうかを確認します。ホストの電源がオフの場合または、ファイアウォールの設定で接続が無効な場合には、Satellite は IP アドレスが利用可能であると誤って判断します。このチェックは、電源がオフになっているホストでは機能しないので、**DHCP** オプションは Satellite が制御し、外部で作成されたホストが含まれていないサブネットでのみ利用できます。

Capsule DHCP モジュールでは、提供された IP アドレスを短期間保持し、同時アクセス時の競合を回避するので、IP 範囲の IP アドレスによっては、一時的に未使用のままとなる場合があります。

- **内部 DB:** Satellite は、Satellite データベースからの IP アドレスを順番にすべて除外し、サブネットの範囲から次に利用可能な IP アドレスを検索します。主要なデータソースは、データベースで、DHCP 予約ではありません。複数のホストが並行して作成される場合には、この IPAM は安全ではないので、このような場合には、代わりに **DHCP** または **Random DB IPAM** を使用してください。
- **Random DB:** Satellite は、Satellite データベースから無作為に全 IP アドレスを除外し、サブネットの範囲から次に利用可能な IP アドレスを検索します。主要なデータソースは、データベースで、DHCP 予約ではありません。この IPAM は、IP アドレスは無作為な順に返されるので、同時にホストを作成する場合にも安全に使用でき、競合の可能性を最小限に抑えることができます。
- **EUI-64:** 48 ビットの MAC アドレスを使用して、RFC2373 に準拠した Extended Unique Identifier (EUI) 64 ビット IPv6 アドレスの生成を取得します。
- **外部 IPAM:** Capsule 機能を使用して IPAM を外部システムに委譲します。Satellite は現在、外部の IPAM 実装機能 hw 同梱されていませんが、プラグインが複数、開発中です。
- **None:** 各ホストの IP アドレスは手動で入力する必要があります。オプション DHCP、内部 DB、およびランダム DB は、外部で作成されたレコードとのサブネットで DHCP の競合を引き起こす可能性があります。これらのサブネットは、Satellite でのみ管理する必要があります。

サブネットの追加に関する詳細は、「[Satellite Server へのサブネットの追加](#)」を参照してください。

## DHCP の範囲

Satellite Server では、検出された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。

## 3.2. SATELLITE および DHCP オプション

Satellite は、DHCP Capsule で DHCP の予約を管理します。Satellite は **next-server** および **filename** DHCP オプションを設定します。

### next-server オプション

**next-server** オプションでは、起動する TFTP サーバーの IP アドレスを提供します。このオプションはデフォルトでは設定されておらず、TFTP Capsule ごとに設定する必要があります。**--foreman-proxy-tftp-servername** オプションを指定して **satellite-installer** コマンドを使用して、**/etc/foreman-proxy/settings.d/tftp.yml** ファイルに TFTP サーバーを設定できます。

```
# satellite-installer --foreman-proxy-tftp-servername 1.2.3.4
```

次に各 TFTP Capsule は API を使用してこの設定を報告して、DHCP レコードの作成時に、Satellite が設定情報を取得できるようになります。

PXE ローダーが **none** に設定されている場合には、Satellite は DHCP レコードに **next-server** オプションを追加しません。

**next-server** オプションが未定義の場合には、Satellite は逆引き DNS 検索を使用して、割り当てる TFTP サーバーのアドレスを検索しますが、以下の問題が発生する可能性があります。

- プロビジョニング中の DNS タイムアウト
- 不正な DNS サーバーのクエリー。(例: キャッシングサーバーではなく、権威サーバーなど)
- TFTP サーバーの不正な IP アドレスに関するエラー。(PTR record was invalid など)

上記の問題が発生した場合には、特に PTR レコードの解決など、Satellite と Capsule の両方で DNS 設定を確認してください。

### filename オプション

**filename** オプションには、プロビジョニング時に、ダウンロードして実行するファイルへの完全パスが含まれます。ホストまたはホストグループに選択した PXE ロードーは、使用する **filename** オプションを定義します。PXE ロードーが **none** に設定されている場合には Satellite は DHCP レコードに **filename** オプションを追加しません。PXE ロードーオプションによっては、**filename** が以下のように変更されます。

PXE ロードーオプション	ファイル名のエントリー	注記
PXELinux BIOS	<b>pxelinux.0</b>	
PXELinux UEFI	<b>pxelinux.efi</b>	
iPXE Chain BIOS	<b>undionly.kpxe</b>	
PXEGrub2 UEFI	<b>grub2/grubx64.efi</b>	x64 はアーキテクチャーにより異なる場合があります
iPXE UEFI HTTP	<b>\http://capsule.example.com:8000/httpboot/ipxe-x64.efi</b>	<b>httpboot</b> 機能が必要で、完全な URL として <b>filename</b> をレンダリングします。ここでは <b>capsule.example.com</b> は、Satellite にある Capsule の既知のホスト名に置き換えます。
Grub2 UEFI HTTP	<b>\http://capsule.example.com:8000/httpboot/grub2/grubx64.efi</b>	<b>httpboot</b> 機能が必要で、完全な URL として <b>filename</b> をレンダリングします。ここでは <b>capsule.example.com</b> は、Satellite にある Capsule の既知のホスト名に置き換えます。

## 3.3. SATELLITE で DHCP の問題をトラブルシューティングする手順

Satellite は内部または外部の DHCP Capsule で ISC DHCP サーバーを管理し、DHCP 予約およびリースを表示、作成、削除できます。Satellite は、DHCP の予約およびリースを一覧表示、作成、削除することができます。ただし、時折発生する可能性のある問題が多数あります。

### 同期していない DHCP レコード

DHCP のオーケストレーション中にエラーが発生した場合は、Satellite データベースと DHCP サーバーの DHCP レコードが一致しない場合があります。これを修正するには、Satellite データベースから不足している DHCP レコードを DHCP サーバーに追加してから、以下の手順に従って DHCP サーバーから不要なレコードを削除する必要があります。

1. DHCP サーバーに追加予定の DHCP レコードをプレビューするには、以下のコマンドを実行します。

```
# foreman-rake orchestration:dhcp:add_missing subnet_name=NAME
```

2. 前の手順のプレビューの変更で問題がない場合は、上記のコマンドに **perform=1** 引数を指定して適用します。

```
# foreman-rake orchestration:dhcp:add_missing subnet_name=NAME perform=1
```

3. Satellite と DHCP サーバーの DHCP レコードが同期されている状態を維持するには、DHCP サーバーから不要な DHCP レコードを削除します。Satellite は、すべての管理対象 DHCP サーバーにサードパーティレコードが含まれていないことを前提とします。そのため、この手順では、想定外のレコードを削除する場合があります。DHCP サーバーから削除されるレコードをプレビューするには、以下のコマンドを入力します。

```
# foreman-rake orchestration:dhcp:remove_offending subnet_name=NAME
```

4. 前の手順のプレビューの変更で問題がない場合は、上記のコマンドに **perform=1** 引数を指定して適用します。

```
# foreman-rake orchestration:dhcp:remove_offending subnet_name=NAME perform=1
```

## PXE ローターオプションの変更

PXE のローダーオプションが既存のホスト用に変更された場合には、DHCP の競合が発生します。この回避策として、DHCP エントリーを上書きしてください。

## DHCP ファイルでの不正なパーミッション

オペレーティングシステムを更新すると、**dhcpd** パッケージが更新される場合があります。これが原因で、重要なディレクトリーやファイルのパーミッションがリセットされて、DHCP Capsule が必要な情報を読み取ることができなくなってしまいます。

詳細は、Red Hat ナレッジベースの [DHCP error while provisioning host from Satellite server Error ERF12-6899 ProxyAPI::ProxyException: Unable to set DHCP entry RestClient::ResourceNotFound 404 Resource Not Found](#) を参照してください。

## DHCP Capsule エントリーの変更

Satellite は、DHCP Capsule セットが含まれるサブネットに割り当てられたホストの DHCP レコードのみを管理します。ホストを作成して、DHCP Capsule を消去または変更した場合に、ホストを削除しようとする、削除のアクションに失敗します。

DHCP Capsule を設定せずにホストを作成してから DHCP Capsule を設定しようとする、DHCP の競合が発生します。

## dhcpd.leases ファイルで削除済みのホストエントリー

DHCP リースに対する変更は、**dhcpd.leases** ファイルの最後に追加されます。エントリーがこのファイルに追加されるので、**dhcpd.leases** ファイルに、同時に同じリースのエントリーが 2 つ以上存在す

る可能性があります。同じリースのエントリーが2つ以上ある場合には、ファイルの最後のエントリーが優先されます。リースファイルのグループ、サブグループ、ホストの宣言も、同じ方法で処理されます。リースが削除されると、`{ deleted; }` が宣言に追加されます。

## 3.4. 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。

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、以下の項目を確認してください。

- 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 **cloud-init** スクリプトに対するさまざまなサポートレベルに関する詳細は、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。

## 3.5. 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** コマンドを実行し、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; 8.8.4.4" \
--foreman-proxy-dns-interface "eth1" \
--foreman-proxy-dns-reverse "140.168.192.in-addr.arpa" \
--foreman-proxy-dns-server "127.0.0.1" \
--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 の機能を更新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"
```

### 3.5.1. インストーラーを使用した複数のサブネットまたはドメイン

satellite-installer オプションでは、単一の DHCP サブネットまたは DNS ドメインしか定義できません。複数のサブネットを定義する方法の1つとして、カスタムの設定ファイルを使用します。

追加のサブネットまたはドメインごとに、`/etc/foreman-installer/custom-hiera.yaml` ファイルにエントリーを作成します。

```
dhcp::pools:
  isolated.lan:
    network: 192.168.99.0
    mask: 255.255.255.0
    gateway: 192.168.99.1
    range: 192.168.99.5 192.168.99.49

dns::zones:
  # creates @ SOA $::fqdn root.example.com.
  # creates $::fqdn A $::ipaddress
  example.com: {}

  # creates @ SOA test.example.net. hostmaster.example.com.
  # creates test.example.net A 192.0.2.100
  example.net:
    soa: test.example.net
    soaip: 192.0.2.100
    contact: hostmaster.example.com.

  # creates @ SOA $::fqdn root.example.org.
  # does NOT create an A record
  example.org:
    reverse: true

  # creates @ SOA $::fqdn hostmaster.example.com.
  2.0.192.in-addr.arpa:
    reverse: true
    contact: hostmaster.example.com.
```

`satellite-installer` を実行して、変更を加え、`/etc/dhcp/dhcpd.conf` に適切なエントリーが含まれていることを確認します。サブネットは、Satellite データベースに定義する必要があります。

### 3.5.2. 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。このオプションは、**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イスを設定します。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。このオプションは、**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イスを設定します。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。このオプションは、**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 --help** を実行し、DHCP、DNS、TFTP その他の Satellite Capsule サービスに関するオプションを表示します。

### 3.5.3. 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。

**Red Hat Enterprise Linux 7 で NAT を使用する場合:**



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 で NAT を実行する場合:

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_conntrack_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_conntrack_tftp` を追加します。

```
IPTABLES_MODULES="ip_conntrack_tftp"
```

## 3.6. 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。

### DNS サーバー設定の考慮事項

DNS レコードの作成時に、Satellite は競合する DNS を検索して、ホスト名が使用されていないことを確認します。これにより、以下の DNS サーバーの1つに対する実行がチェックされます。

- システム全体のリゾルバー (管理 > 設定 > ローカルネームサーバーのクエリー が true に設定されている場合)
- ホストに関連付けられたサブネットに定義されているネームサーバー
- ホストに関連付けられたドメイン名から SOA に照会される信頼できる NS レコード

DNS 競合の解決時にタイムアウトが発生した場合は、次の設定を確認してください。

- サブネットネームサーバーは、Satellite Server から到達可能である必要があります。
- ドメイン名には、Satellite Server から入手できる Start of Authority (SOA) レコードが必要です。
- `/etc/resolv.conf` ファイルのシステムリゾルバーには、有効かつ機能する設定が必要です。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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 手順

- **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 上の 統合 Capsule ID です。

## 3.7. 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イスを設定するので、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。インターフェイスを設定するには、Satellite Server には、これらのインターフェイ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。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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. **ネットワークの接頭辞** フィールドには、サブネットのネットワーク接頭辞を入力します。
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。IPAM の詳細は、「[ネットワークリソース](#)」を参照してください。
11. 選択した IPAM メソッドの情報を入力します。**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。
12. **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。
13. **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。
14. **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。
15. **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。
16. **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。
17. **送信** をクリックしてサブネットの情報を保存します。

## CLI 手順

- 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。

## 第4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 6.10 以降で、Infoblox バージョンはサポートされます。

### 4.1. 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** コマンドを使用して、Capsule に Infoblox モジュールをインストールしてビューを設定した後は、ビューを編集できません。

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。詳細は、[3章 ネットワークの設定](#)を参照してください。

Satellite IPAM 機能は Infoblox と統合できません。

### 4.2. 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 の管理ロールを設定されていることを確認してください。

管理ロールは、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。

### 4.3. 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.example.com:443 </dev/null | \
openssl x509 -text >/etc/pki/ca-trust/source/anchors/infoblox.crt
# update-ca-trust extract
```

- **infoblox.example.com** 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:password https://infoblox.example.com/wapi/v2.0/network
```

証明書に問題がない場合の応答例:

```
[
  {
    "_ref":
    "network/ZG5zLm5ldHdvcmskMTkyLjE2OC4yMDluMC8yNC8w:infoblox.example.com/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を使用して証明書をインストールします ([How to install a CA certificate on Red Hat Enterprise Linux 6/7 and later](#))。

## 4.4. 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。

### 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 fixedaddress** 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。

**host** 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 **host** 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、競合を発生させずに、**host** 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 fixedaddress \
--foreman-proxy-dhcp-server infoblox.example.com \
--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、**リフレッシュ** 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。

## 4.5. 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、以下のコマンドを入力して Infoblox モジュールを設定します。

```
# 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.example.com \  
--foreman-proxy-plugin-dns-infoblox-username admin \  
--foreman-proxy-plugin-dns-infoblox-password infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-view default
```

オプションで **--foreman-proxy-plugin-dns-infoblox-dns-view** オプションの値を変更して、デフォルトビュー以外の DNS Infoblox ビューに指定します。

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、**リフレッシュ** 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5章 プロビジョニング時間を削減するための iPXE の設定

以下の制限で PXE を使用できない場合には、Satellite を使用し、BIOS モードの iPXE をチェーンブートして、HTTP プロトコルで起動するように PXELinux を設定することができます。

- 管理対象外の DHCP サーバーのあるネットワーク
- ネットワーク上でブラックリストに登録されているか、ファイアウォールによって制限されている PXE サービス
- 低帯域幅ネットワークなどが原因で、信頼性の低い TFTP UDP ベースのプロトコル

iPXE サポートの詳細については、[Supported architectures for provisioning](#) の記事を参照してください。

### iPXE の概要

iPXE は、オープンソースのネットワークブートファームウェアです。HTTP サーバーからの起動など、追加機能で強化された完全な PXE 実装を提供します。詳細は、[ipxe.org](http://ipxe.org) を参照してください。

Red Hat Satellite で iPXE を使用方法は 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

### セキュリティ情報

Red Hat Enterprise Linux の iPXE バイナリーは、セキュリティ機能を使用せずに構築されます。このため、HTTPS は使用できず、HTTP だけが使用できます。

Red Hat Enterprise Linux の iPXE のセキュリティ関連機能はすべて、サポート対象ではありません。詳細情報は、[Red Hat Enterprise Linux HTTPS support in iPXE](#) を参照してください。

### 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 にホストが存在する。
- プロビジョニングインターフェイスの MAC アドレスが、ホスト設定と一致する。
- ホストのプロビジョニングインターフェイスには、有効な DHCP 予約がある。
- NIC は PXE ブートが可能である。詳細は、[supported hardware on ipxe.org](http://supportedhardwareonipxe.org) のページで、iPXE 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。
- NIC は iPXE と互換性がある。

iPXE 環境を準備するには、すべての Capsule でこの手順を実行する必要があります。

### 手順

1. **tftp** サービスおよび **httpboot** サービスを有効にします。

```
# satellite-installer --foreman-proxy-httpboot true --foreman-proxy-tftp true
```

2. **ipxe-bootimgs** RPM パッケージをインストールしてください。

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

- Linux カーネルヘッダーを含む iPXE ファームウェアを TFTP ディレクトリーにコピーします。

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

- UNDI iPXE ファームウェアを TFTP ディレクトリーにコピーします。

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

3. SELinux ファイルコンテキストを修正します。

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

4. 必要に応じて、Foreman ディスカバリーを設定します。詳細は、[7章 Discovery サービスの設定](#)を参照してください。

- 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。
- **Default PXE global template entry** の行を特定して、**Value** の列で値を **discovery** に変更します。

## 5.1. 仮想マシンの起動

仮想化ハイパーバイザーの一部は、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 なしで仮想マシンを起動できます。

### 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 は、Capsule から iPXE ブートストラップテンプレートをロードします
5. iPXE は、Foreman から URL パラメーターとして MAC を使用して iPXE テンプレートをロードします
6. 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
- Red Hat Virtualization
- RHEV

## iPXE を使うための Satellite Server の設定

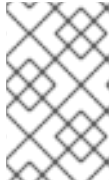
デフォルトのテンプレートを使用して、ホストの iPXE ブーティングを設定できます。テンプレートのデフォルトの値を変更する場合は、テンプレートをクローンし、そのクローンを編集します。

### 手順

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. **Association** 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. **Locations** タブをクリックして、ホストがある場所を追加します。
7. **Organizations** 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. **オペレーティングシステム** タブを選択します。
16. **PXE Loader** を **iPXE Embedded** または **None** に設定します。
17. **テンプレート** タブを選択します。
18. **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。
19. **dhcpd.conf** ファイルを次のように設定します。

```
if exists user-class and option user-class = "iPXE" {  
    filename "http://capsule.example.com:8000/unattended/iPXE?bootstrap=1";  
} # elseif existing statements if non-iPXE environment should be preserved
```

- 分離したネットワークを使用する場合には、Satellite Server の URL ではなく、TCP ポートが **8000** の Capsule Server URL を使用します。



### 注記

--foreman-proxy-http-port installer オプションを使用してポートを変更した場合は、カスタムポートを使用します。アップグレード後は必ず、`/etc/dhcp/dhcpd.conf` ファイルを更新してください。

## 5.2. PXELINUX からの IPXE 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 インターフェイスに内蔵のドライバーを使用するように設定します。iPXE で HTTP を使用するには、組み込みのドライバー (**ipxe.lkrn**) を備えた iPXE ビルドを使用します。Universal Network Device Interface (UNDI) は、TFTP クライアントを実装する最小限の UDP/IP スタックですが、HTTP (**undionly-ipxe.0**) などの他のプロトコルをサポートすることはできません。ネットワークハードウェアの機能と iPXE ドライバーの可用性に応じて、**ipxe.lkrn** または **undionly-ipxe.0** ファイルのいずれかをロードすることを選択できます。

### 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。

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. **PXELinux chain iPXE** を入力して **ipxe.lkrn** を使用するか、BIOS システムの場合は **PXELinux chain iPXE UNDI** を入力して **undionly-ipxe.0** を使用し、**Search** をクリックします。
3. 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。
4. 使用するテンプレートの名前をクリックします。
5. 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。

6. **Association** 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。
7. **Locations** タブをクリックして、ホストがある場所を追加します。
8. **Organizations** 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. **送信** をクリックして変更を保存します。
10. **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。
11. 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。
12. 使用するテンプレートの名前をクリックします。
13. 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。
14. **Association** 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にテンプレートを関連付けます。
15. **Locations** タブをクリックして、ホストがある場所を追加します。
16. **Organizations** タブをクリックして、ホストが属する組織を追加します。
17. **送信** をクリックして変更を保存します。
18. **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。
19. **テンプレート** タブをクリックします。
20. **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。
21. **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。
22. **送信** をクリックして変更を保存します。
23. **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。
24. **オペレーティングシステム** タブを選択します。
25. **PXE Loader** を **PXELinux BIOS** に設定して PXELinux 経由で iPXE をチェーンブートするか、**iPXE Chain BIOS** に設定して **undionly-ipxe.0** を直接ロードします。
26. **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。
27. **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。
28. **dhcpd.conf** ファイルを次のように設定します。

```
if exists user-class and option user-class = "iPXE" {  
    filename "http://capsule.example.com:8000/unattended/iPXE?bootstrap=1";  
} # elseif existing statements if non-iPXE environment should be preserved
```



### 注記

**--foreman-proxy-http-port** インストーラーオプションを使用してポートを変更した場合は、カスタムポートを使用します。アップグレード後は必ず、**/etc/dhcp/dhcpd.conf** ファイルを更新してください。

## 第6章 PXE を使用したホストのプロビジョニング

Red Hat Satellite 6.10 では、ベアメタルインスタン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。詳細は、[7章 Discovery サービスの設定](#) を参照してください。

### Discovery を使用した PXE なしのプロビジョニング

新規ホストは ISO ブートディスクを使用して 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。詳細は、「[PXE を使用しない Discovery の実装](#)」を参照してください。

### BIOS および UEFI サポート

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

BIOS および UEFI インターフェイスはいずれも、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。

サポート対象のワークフローに関する詳細は、[Supported architectures and provisioning scenarios](#) を参照してください。

Satellite のプロビジョニングでは、PXE ローダーオプションが、プロビジョニング時に使用する DHCP の **filename** オプションを定義します。BIOS システムの場合は、**PXELinux BIOS** オプションを使用してプロビジョニングノードを有効化し、TFTP 経由で **pxelinux.0** ファイルをダウンロードします。また、UEFI システムの場合は、**PXEGrub2 UEFI** オプションを使用して、TFTP クライアントを有効にして **grub2/grubx64.efi** ファイルをダウンロードします。

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 ローダ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。

- 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。  
ネットワーク要件の詳細は、[3章 ネットワークの設定](#) を参照してください。

Discovery サービスの詳細は、[7章 Discovery サービスの設定](#) を参照してください。

- ベアメタルホストまたは空の仮想マシン
- Red Hat Enterprise Linux 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。
- 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。

## 6.2. セキュリティートークンの有効期間の設定

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。

トークンの生成が無効になっている場合には、攻撃者はクライアントの IP アドレスを偽装して、Satellite Server から、暗号化された root パスワードなど、キックスタートをダウンロードできます。

## 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。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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. **インターフェイス** タブをクリックして、ホストのインターフェイスで、**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イスに **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、それらを選択してください。
9. **MAC アドレス** フィールドには、ホストのプロビジョニングインターフェイスの 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。

10. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
11. **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。
12. オプション: **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の詳細は、「**プロビジョニングテンプレート**」を参照してください。
13. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
14. **送信** をクリックしてホストの詳細を保存します。  
ネットワークインターフェイスの詳細は、[ネットワークインターフェイスの追加](#) を参照してください。

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。物理ホストを起動して、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の HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

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 にも登録し、必要な設定と管理ツールを Satellite Tools 6.10 のリポジトリーからインストールします。

## CLI 手順

1. **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
```

2. **hammer host interface update** コマンドを使用し、ネットワークインターフェイ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. UEFI HTTP ブートプロビジョニングを使用したホストの作成

UEFI HTTP ブートを使用して、Satellite からホストをプロビジョニングできます。これは、IPv6 ネットワークでホストをプロビジョニングできる唯一の方法です。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 前提条件

- HTTP 起動要件を満たすようにしてください。詳細は、Red Hat Satellite の計画の [HTTP ブートの要件](#) を参照してください。

## 手順

1. プロビジョニングに使用する Capsule で、**grub2-efi** パッケージを最新版に更新します。

```
# satellite-maintain packages install grub2-efi
```

2. **foreman-proxy-http**、**foreman-proxy-httpboot** および **foreman-proxy-tftp** 機能を有効にします。

```
# satellite-installer --scenario satellite \  
--foreman-proxy-httpboot true \  
--foreman-proxy-http true \  
--foreman-proxy-tftp true
```

3. Satellite Web UI で、**ホスト > ホストの作成** に移動します。
4. **名前** フィールドには、ホストの名前を入力します。
5. **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。
6. **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。
7. **インターフェイス** タブをクリックして、ホストのインターフェイスで、**編集** をクリックします。
8. 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。
9. **MAC アドレス** フィールドには、ホストのプロビジョニングインターフェイスの 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。
10. Satellite Server が、ホストの最初のインターフェイスに **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、それらを選択してください。
11. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
12. **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。
13. **PXE Loader** の一覧から **Grub2 UEFI HTTP** を選択します。
14. オプション: **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。
15. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
16. **送信** をクリックしてホストの詳細を保存します。



ネットワークインターフェイスの詳細は、[ネットワークインターフェイスの追加](#) を参照してください。

17. ネットワークから UEFI モードで起動するようにホストを設定します。
18. ホストを起動します。
19. 起動メニューから、**Kickstart default PXEGrub2** を選択します。

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、UEFI 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。物理ホストを起動して、ブートモードを UEFI HTTP に設定すると、ホストは定義済みの DHCP サービスを検出し、キックスターツリーで Capsule の HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

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 にも登録し、必要な設定と管理ツールを Satellite Tools 6.10 のリポジトリーからインストールします。

### CLI 手順

1. プロビジョニングに使用する Capsule で、**grub2-efi** パッケージを最新版に更新します。

```
# satellite-maintain packages install grub2-efi
```

2. **foreman-proxy-http**、**foreman-proxy-httpboot** および **foreman-proxy-tftp true** 機能を有効にします。

```
# satellite-installer --scenario satellite \
--foreman-proxy-httpboot true \
--foreman-proxy-http true \
--foreman-proxy-tftp true
```

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

```
# hammer host create --name "My_Host" \
--organization "My_Organization" \
--location "My_Location" \
--hostgroup "My_Host_Group" \
--mac "aa:aa:aa:aa:aa:aa" \
--build true \
--enabled true \
--managed true \
--pxe-loader "Grub2 UEFI HTTP"
```

4. **hammer host interface update** コマンドを使用し、ネットワークインターフェイ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 "My_Host" \
--managed true \
--primary true \
--provision true
```

5. ネットワークから UEFI モードで起動するようにホストを設定します。
6. ホストを起動します。

7. 起動メニューから、**Kickstart default PXEGrub2** を選択します。

これで、ホストのエントリ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、UEFI 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。物理ホストを起動して、ブートモードを UEFI HTTP に設定すると、ホストは定義済みの DHCP サービスを検出し、キックスタートツリーで Capsule の HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

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 にも登録し、必要な設定と管理ツールを Satellite Tools 6.10 のリポジトリーからインストールします。

## 6.5. 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 の管理のユーザーの SSH キー管理](#) 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 [Red Hat Satellite でのホストの作成](#) を参照してください。

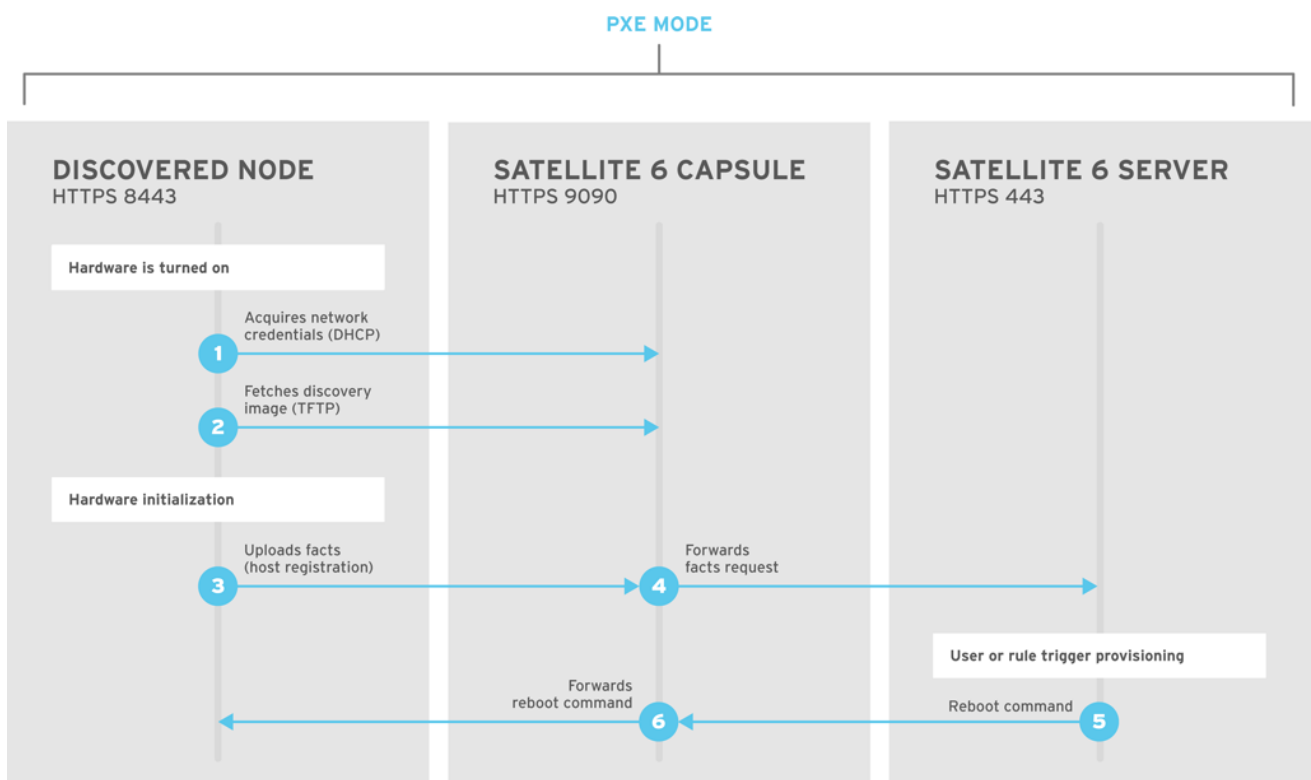
**Owned by** ユーザーの SSH キーは、プロビジョニングプロセス中に **create\_users** スニペットが実行されると、自動的に追加されます。**Owned by** は、個人のユーザーやユーザーグループに設定することができます。**Owned by** をユーザーグループに設定すると、そのユーザーグループ内の全ユーザーの SSH キーが自動的に追加されます。

## 第7章 DISCOVERY サービスの設定

Red Hat Satellite は、Satellite インベントリに含まれていないネットワーク上に存在するホストを検出できます。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。

空のベアメタルホストを起動する場合には、起動メニューに **local** と **discovery** の2つのオプションが含まれます。**discovery** を選択して、Discovery イメージを起動すると、数分後に Discovery イメージの起動が完了して、ステータス画面が表示されます。

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は、デフォルトで、ローカルハードドライブから起動するように設定されています。デフォルトの設定を変更するには、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。デフォルトの PXE グローバルテンプレートエントリーの列を特定し **値** の行に **discovery** と入力します。



Discovery イメージの提供に Satellite Server を使用するには、以下の RPM パッケージをインストールします。

- **tfm-rubygem-foreman\_discovery**
- **foreman-discovery-image**
- **tfm-rubygem-smart\_proxy\_discovery**

**tfm-rubygem-foreman\_discovery** パッケージには、検出されたノード、接続、必要なデータベース構造、API を処理する Satellite プラグインが含まれます。

**tfm-rubygem-smart\_proxy\_discovery** パッケージは、Discovery サービスのプロキシとして機能するように Capsule Server (Satellite Server の統合 Capsule など) を設定します。

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。

## 7.1. DISCOVERY サービスのインストール

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

### 手順

1. Capsule Server で以下のコマンドを入力します。

```
# satellite-maintain packages install foreman-discovery-image tfm-rubygem-smart_proxy_discovery
```

2. satellite-maintain サービスを再起動します。

```
# satellite-maintain service restart
```

3. Satellite Web UI で、**インフラストラクチャー > Capsule** に移動します。
4. Capsule Server をクリックして、**アクション** 一覧から **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、Discovery サービスを提供するために適切な Capsule Server が選択されている必要があります。

これを確認するには、**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、**機能のリフレッシュ** 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 タブに移動し、使用する **Discovery Proxy** を選択します。使用するサブネットごとにこれを実行します。

## 7.2. 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 > プロビジョニングテンプレート** 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 **pxegrub2\_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。

### 7.3. 検出されたホストの自動コンテキスト

Satellite Server は以下のルールの順番に従って、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. 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。
2. **discovery\_organization** または **discovery\_location** ファクト値が設定されている場合には、検出ホストはこれらのファクト値を組織およびロケーションとして使用します。このファクト値を設定するには、**管理 > 設定 > 検出済み** に移動して、これらのファクトを **デフォルトの組織** と **デフォルトのロケーション** フィールドに追加します。検出ホストのサブネットが、ファクトで設定した組織とロケーションに所属しているようにします。所属していない場合には、Satellite はセキュリティ上の理由で設定を拒否します。
3. 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。

**foreman\_organization** および **foreman\_location** ファクトは、検出ホストにコンテキストを割り当てる値として、有効ではなくなりました。ただし、これらのファクトを使用して、Puppet 実行用のホストはまだ設定できます。設定方法は、**管理 > 設定 > Puppet** セクションに移動し、**foreman\_organization** と **foreman\_location** ファクトを **デフォルトの組織** と **デフォルトのロケーション** フィールドに追加します。

### 7.4. DISCOVERY テンプレートとスニペットの設定

Discovery サービスを使用するには、プロビジョニングオプションを指定して、デフォルトサービスとして Discovery を設定し、使用するテンプレートを設定する必要があります。

#### 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。

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のカスタマイズ

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。

テンプレートまたはテンプレートに含まれるスニペットを変更した場合には、Satellite Server のデフォルトの PXE テンプレートにその変更内容を伝搬する必要があります。

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、**PXE のデフォルトのビルド** をクリックします。

これで、Satellite Server 上にあるデフォルトの PXE テンプレートが更新されます。

## 追加の設定

### 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 を使用する場合は、**proxy.type** 引数を **proxy** に設定することを確認します。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するか、適切なサブネットの共通 DNS エイリアスを使用してください。

## 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
```

## 7.5. 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。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 前提条件

- Satellite でドメインとサブネットを設定する。ネットワーク要件の詳細は、[3章 ネットワークの設定](#) を参照してください。
- Satellite で Discovery サービスを設定する。詳細は、「[Discovery サービスのインストール](#)」を参照してください。
- ベアメタルホストまたは空の仮想マシン
- Red Hat Enterprise Linux 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。
- 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。

## 手順

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 の結果をもとに 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イスに **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、それらを選択してください。
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** の **解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。ホストは、以下のプロビジョニングテンプレートを解決する必要があります。
  - **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の詳細は、「[プロビジョニングテンプレート](#)」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手順

1. プロビジョニングに使用する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
```

このコマンドで、検出ホストの一覧からホストを削除し、プロビジョニング設定を使用してホストのエントリを作成します。Discovery イメージは自動的にホストをリセットし、PXE にブートできるようにします。ホストは、Satellite Server の統合 Capsule 上の DHCP サービスを検出して、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。残りのプロセスは、「[無人プロビジョニングによるホストの作成](#)」に記載されている、通常の PXE ワークフローと同じです。

## 7.6. 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、Satellite 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。

Web UI の代わりに CLI を使用する場合は、[CLI手順](#) を参照してください。

### 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、スクリプトを使用するか、後で設定管理を設定して **kickstart** スクリプトレットで NIC を設定できます。

### 手順

1. Satellite Web UI で **設定 > ルールの検出** に移動し、**ルールの作成** を選択します。
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。ホスト名は、ホスト固有の値に `@host` 属性を、乱数に `rand` マクロを、または値を増やすために `sequence_hostgroup_param_next` マクロを使用できます。テンプレートのプロビジョニングに関する詳細は、「[プロビジョニングテンプレート](#)」および API ドキュメントを参照してください。
  - `myhost-<%= sequence_hostgroup_param_next("EL7/MyHostgroup", 10, "discovery_host") %>`
  - `myhost-<%= rand(99999) %>`
  - `abc-<%= @host.facts['bios_vendor'] %>-<%= rand(99999) %>`
  - `xyz-<%= @host.hostgroup.name %>`
  - `srv-<%= @host.discovery_rule.name %>`
  - `server-<%= @host.ip.gsub(':', '-') + '-' + @host.hostgroup.subnet.name %>`  
 ホスト名のパターンの作成時には、作成す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ができず、アンダースコアやドットを含めることができません。適切な方法として、`facter` で提供される固有の情報 (MAC アドレス、BIOS、またはシリアル ID など) を使用することができます。
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 手順

1. **hammer discovery-rule create** コマンドを使用してルールを作成します。

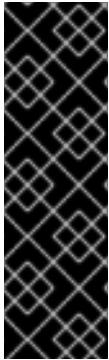
```
# hammer discovery-rule create --name "Hypervisor" \
--search "cpu_count > 8" --hostgroup "My_Host_Group" \
--hostname "hypervisor-<%= rand(99999) %>" \
--hosts-limit 5 --priority 5 --enabled true
```

2. **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"
```

## 7.7. PXE を使用しない DISCOVERY の実装

Satellite には PXE なしの Discovery サービスがあり、PXE ベースのサービス (DHCP や TFTP) なしに操作できます。これは、Satellite Server の Discovery イメージを使用することで、実現できます。切断ノードのインストールをスケジュールしたら、**kexec** コマンドを使用して、ノードを再起動せずに、OS インストーラーで Linux カーネルを再読み込みします。

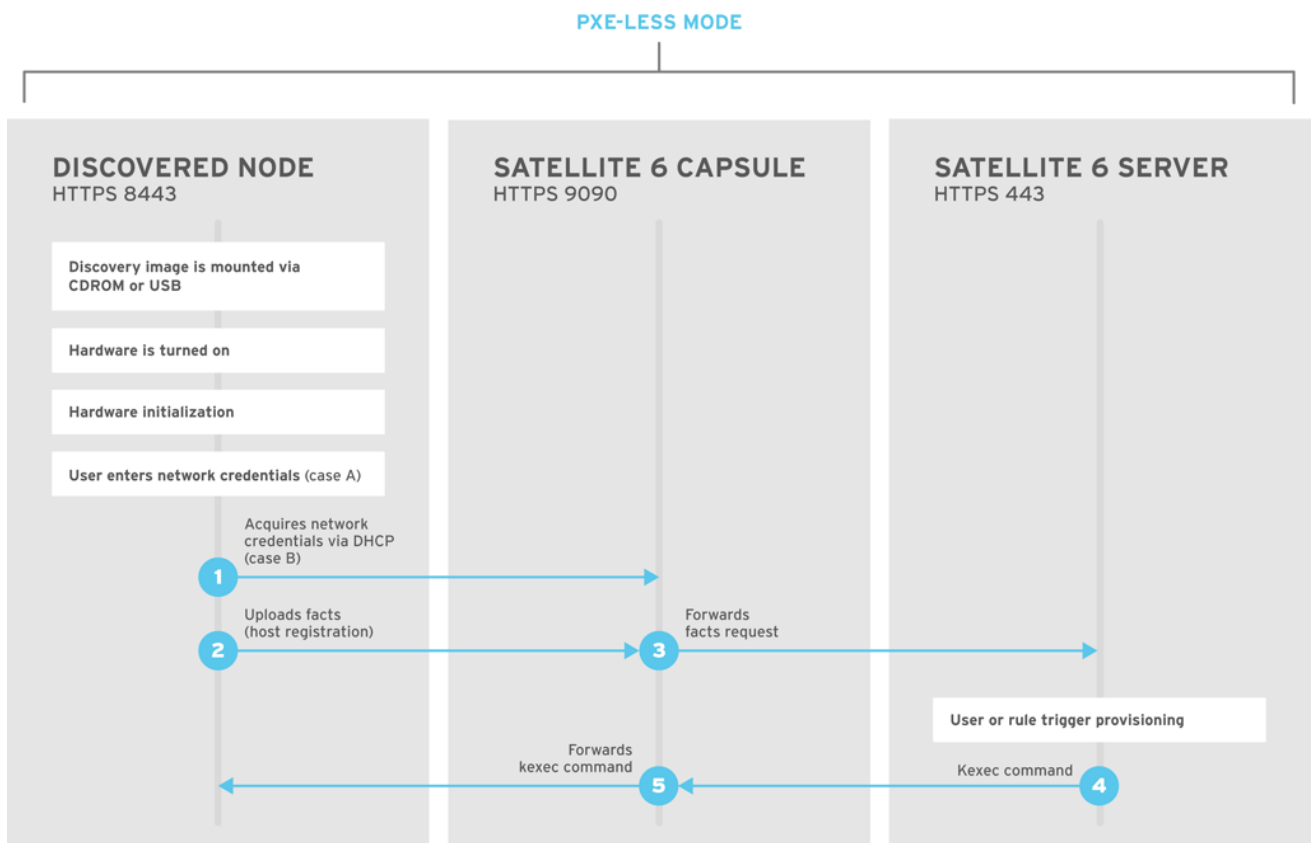


### 重要

Discovery **kexec** は、テクノロジープレビュー機能のみとなっています。テクノロジープレビュー機能は、Red Hat の実稼働サービスレベルアグリーメント (SLA) でサポートされておらず、機能的に完全でない可能性があります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。Red Hat テクノロジープレビュー機能のサポート範囲に関する情報は、<https://access.redhat.com/ja/support/offerings/techpreview> を参照してください。

### 既知の問題

プロセス実行時にコンソールがフリーズする可能性があります。ハードウェアによっては、グラフィカルハードウェアの問題が発生する場合があります。



SATELLITE6\_376339\_1115

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

1. 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
```

2. 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イスが含まれます。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イスの設定オプションも要求します。

3. これらの詳細を入力した後に、**次へ** を選択します。
4. **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イ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イスを自動的に設定しようとします。
5. プライマリーインターフェイスの設定後に、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
```

6. **Connection type** を **Proxy** に設定し、**次へ** を選択します。
7. オプション: Discovery イメージは、Facter ツールが Satellite Server に送り返す **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。

検出されたホストのプロビジョニング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。

## 7.8. 無人での使用、カスタマイズ、イメージのリマスター

カスタマイズされた Discovery ISO を作成して、起動後にイメージ設定プロセスを自動化できます。Discovery イメージは、オペレーティングシステムの Linux カーネルを使用して、カーネルのパラメーターを指定し、Discovery サービスを設定します。このカーネルパラメーターには、以下のエントリが含まれます。

### proxy.url

Discovery サービスを提供する Capsule Server または Satellite 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。この場合には、Capsule Server ではなく Satellite Server との通信が直接行われます。

## fdi.pxmac

プライマリーインターフェイスの 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イスです。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イスを選択するよう求められます。

## fdi.pxip, fdi.pxgw, fdi.pxdns

プライマリーネットワークインターフェイスの 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イスを設定します。

## 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 に設定した場合には、イメージは準自動化モードを使用します。このモードでは、一連のダイアログオプションで選択内容を確認できます。1 に設定した場合、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。

## fdi.initnet

デフォルトでは、このイメージは全ネットワークインターフェイス (値 **all**) を初期化します。この設定を **bootif** に指定すると、ネットワークブートに使用したネットワークインターフェイスのみが初期化されます。

## fdi.rootpw

デフォルトでは **root** アカウントはロックされています。このオプションを使用して Root パスワードを設定します。クリアテキストのパスワードも、暗号化パスワードも両方入力できます。

## fdi.ssh

デフォルトでは SSH サービスは無効です。これは **1** または **true** に設定して、SSH アクセスを有効にします。

## fdi.ipv4.method

デフォルトでは、NetworkManager IPv4 メソッドの設定は **auto** に設定されます。このオプションが優先されるので、IPv4 スタックを無効にするには **ignore** に設定します。このオプションが機能するのは、DHCP モードの場合のみです。

## fdi.ipv6.method

デフォルトでは、NetworkManager IPv6 メソッドの設定は **auto** に設定されます。このオプションが優先されるので、IPv6 スタックを無効にするには **ignore** に設定します。このオプションが機能するのは、DHCP モードの場合のみです。

## fdi.zips

起動中にダウンロードおよび起動する拡張子を持つファイル名。詳細は、[「Discovery イメージの拡張」](#) を参照してください。

## fdi.zipserver

拡張子のダウンロードに使用する TFTP サーバー。詳細は、[「Discovery イメージの拡張」](#) を参照してください。

## fdi.countdown

初回の検出試行後に text-user インターフェイスがリフレッシュされるまで待機する秒数。デフォルト値は 45 秒です。ステータスページで IP アドレスを **N/A** として報告する場合は、この値を増やします。

#### fdi.dhcp\_timeout

NetworkManager DHCP のタイムアウト。デフォルト値は 300 秒です。

#### fdi.vlan.primary

プライマリーインターフェイスに設定する VLAN タグ付け ID。

### discovery-remaster ツールを使用した OS イメージのリマスター

Satellite Server は、**foreman-discovery-image** パッケージで **discovery-remaster** ツールも提供します。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。以下に例を示します。

```
# 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"
```

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。

検出されたホストのプロビジョニング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。

## 7.9. DISCOVERY イメージの拡張

カスタムファクト、ソフトウェア、またはデバイスドライバーを使って Satellite Discovery イメージを拡張することができます。イメージで使用できるように追加コードが含まれる圧縮されたアーカイブファイルを提供することもできます。

### 手順

- 最初に、以下のディレクトリ構造を作成します。

```
.
├── autostart.d
│   └── 01_zip.sh
├── bin
│   └── ntpdate
├── facts
│   └── test.rb
└── lib
```

```
├── libcrypto.so.1.0.0
│   └── ruby
│       └── test.rb
```

- **autostart.d** ディレクトリーには、ホストが Satellite に登録される前の起動時に POSIX の順にイメージにより実行されるスクリプトが含まれています。
  - **bin** ディレクトリーは、**\$PATH** 変数に追加され、このディレクトリーにバイナリーファイルを配置して、**autostart** スクリプトで使用します。
  - **facts** ディレクトリーは **FACTERLIB** 変数に追加され、カスタムファクトを設定して Satellite に送信できるようになります。
  - **/bin** のバイナリーファイルが正しく実行できるように、**lib** ディレクトリーは **LD\_LIBRARY\_PATH** 変数に、**lib/ruby** は **RUBYLIB** 変数に追加されます。
2. ディレクトリー構造の作成後に、以下のコマンドで **.zip** アーカイブにパッケージ化します。

```
zip -r my_extension.zip .
```

3. Discovery イメージで使用される拡張子が認識されるように、検出イメージと共に zip ファイルを TFTP サーバーに配置してから、PXELinux テンプレートの **APPEND** 行を、パスが TFTP ルートに相対する **fdi.zips** オプションで更新します。たとえば、**\$TFTP/zip1.zip** および **\$TFTP/boot/zip2.zip** に 2 つのアーカイブがある場合は、以下の構文を使用します。

```
fdi.zips=zip1.zip,boot/zip2.zip
```

既存の環境変数 (**PATH**、**LD\_LIBRARY\_PATH**、**RUBYLIB** および **FACTERLIB**) に、新規ディレクティブとオプションを追加できます。スクリプトで明示的にパスを指定する場合には、**.zip** ファイルのコンテンツが、イメージの **/opt/extension** ディレクトリーに展開されます。

複数の **.zip** ファイルを作成できますが、Discovery イメージと同じ場所に展開される点に注意してください。**.zip** ファイルから後で展開されたファイルは、同じファイル名を使用している場合には、以前のバージョンを上書きします。

## 7.10. DISCOVERY のトラブルシューティング

マシンが、Satellite Web UI の **ホスト > 検出されたホスト** に表示されない場合は、以下の設定領域を調べてエラーを切り分けます。

- **ホスト > プロビジョニングテンプレート** に移動し、**PXE デフォルトのビルド** ボタンを使用してデフォルトの PXELinux テンプレートを再デプロイします。
- TFTP Capsule Server で **pxelinux.cfg/default** 設定ファイルを確認します。
- ホスト、Capsule Server、および Satellite Server 間で適切なネットワーク接続があることを確認します。
- 使用している PXELinux テンプレートに含まれている PXE 検出スニペットを確認します。スニペットの名前は **pxelinux\_discovery**、**pxegrub\_discovery**、または **pxegrub2\_discovery** です。PXE 検出スニペットの **proxy.url** オプションと **proxy.type** オプションを検証してください。

- 検出されたノードで DNS が適切に機能していることを確認するか、使用している PXE Linux テンプレートにある PXE 検出スニペットの **proxy.url** オプションにある IP アドレスを使用します。
- DHCP サーバーが IP アドレスを起動したイメージに適切に送信していることを確認します。
- 検出されたホスト (または仮想マシン) に 1200 MB 以上のメモリーがあることを確認します。メモリーが 1200 MB より少なくなると、イメージがインメモリーで抽出されるので、各種のカーネルパニックエラーがランダムに発生する可能性があります。

重要なシステムファクトを収集するには、**discovery-debug** コマンドを使用します。これにより、システムログ、ネットワーク設定、ファクトの一覧などの情報が標準出力に出力されます。通常のユースケースでは、追加の調査のために、**scp** コマンドでこの出力をリダイレクトしてコピーします。

検出されたホストの最初の仮想コンソールは **systemd** ログのために予約されます。特に役立つシステムログには、以下のようにタグが付けられます。

- **discover-host**: 最初のファクトのアップロード
- **foreman-discovery**: ファクトの更新、リモート再起動のコマンド
- **nm-prepare**: NetworkManager を事前に定義する起動スクリプト
- **NetworkManager**: ネットワークの情報

TTY2 以上を使用して、検出されたホストにログインします。root アカウントおよび SSH アクセスはデフォルトで無効にされますが、以下のカーネルコマンドラインのオプションを使って、デフォルト PXELinux テンプレートの APPEND 行で、SSH の有効化および root パスワードの設定ができます。

```
fdi.ssh=1 fdi.rootpw=My_Password
```

## 第8章 RED HAT IMAGE BUILDER イメージを使用したプロビジョニング

Satellite で、Red Hat Web コンソールを統合して、アクションの実行と、ホストの監視を行うことができます。Red Hat の Web コンソールを使用して、Red Hat Image Builder にアクセスし、イメージをビルドしてから HTTP サーバーにアップロードし、このイメージを使用してホストをプロビジョニングします。イメージのプロビジョニング用に Satellite を設定すると、Anaconda インストーラーがディスクをパーティションに分割し、イメージをダウンロードしてマウントし、ファイルをホストにコピーします。推奨されるイメージタイプは TAR です。

Red Hat Web コンソールと Satellite の統合に関する詳細は、[ホストの管理](#) ガイドの [Red Hat Web コンソールを使用したホストの管理と監視](#) を参照してください。

### 前提条件

1. Red Hat Image Builder を使用して作成される既存の TAR イメージです。

### 手順

Satellite で、Anaconda Kickstart で Red Hat Image Builder イメージを使用するには、次の手順を実行します。

1. Satellite で、カスタムの製品を作成し、カスタムのファイルリポジトリをこの製品に追加して、イメージをリポジトリにアップロードします。詳細は、[コンテンツ管理ガイド](#) の [個別の ISO イメージとファイルのインポート](#) を参照してください。
2. Satellite Web UI で、[設定](#) > [ホストグループ](#) に移動し、使用するホストグループを選択します。
3. [パラメーター](#) タブをクリックしてから、[パラメーターの追加](#) をクリックします。
4. [名前](#) フィールドで、`kickstart_liveimg` を入力します。
5. [タイプ](#) の一覧から `string` を選択します。
6. [値](#) のフィールドに、イメージの保存先を参照する相対パスまたは絶対パスを `custom/product/repository/image_name` の形式で入力します。
7. [送信](#) をクリックして変更を保存します。

ベアメタルプロビジョニングや、コンピュートリソースを使用したプロビジョニングに、このイメージを使用できます。ベアメタルプロビジョニングの詳細は、[6章 PXE を使用したホストのプロビジョニング](#) を参照してください。異なるコンピュートリソースを使用したプロビジョニングの詳細は、使用するコンピュートリソースに関する章を参照してください。



## 第9章 KVM (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 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。

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

### 前提条件

- Red Hat Enterprise Linux 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。
- 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 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、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。Capsule Server のネットワークサービス設定の詳細は、[ネットワークの設定](#) を参照します。
- KVM 仮想化ツールを実行する Red Hat Enterprise Linux サーバー (libvirt daemon)。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。Satellite 経由で管理できるのは、ディレクトリーのプールストレージタイプのみです。
- オプション: この手順の例では、KVM の root ユーザーを使用します。KVM サーバーで root 以外のユーザーを使用する場合には、KVM サーバーの **libvirt** グループにユーザーを追加します。

```
useradd -a -G libvirt non_root_user
```

- 以下のロールが割り当てられた Satellite ユーザーアカウント:
  - **Edit hosts**
  - **View hosts**  
詳細は、[Red Hat Satellite の管理ガイド](#) の [ユーザーへのロールの割り当て](#) を参照してください。
- 以下のパーミッションが指定された Satellite のカスタムロール
  - **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、**Red Hat Satellite の管理** ガイドの [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 の管理** 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。

## 手順の概要

1. [「Satellite Server の KVM 接続設定」](#)
2. [「Satellite Server への KVM 接続の追加」](#)
3. オプション: [「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する場合にはこの手順を使用します。
4. [「KVM の詳細のコンピュートプロファイルへの追加」](#)
5. [「KVM でのホストの作成」](#)

## 9.1. SATELLITE SERVER の KVM 接続設定

KVM 接続を追加する前に、**foreman** ユーザーの SSH キーペアを作成して、Satellite Server と KVM 間の安全な接続を確保します。

### 手順

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。

```
# satellite-maintain packages 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'
```

## 9.2. SATELLITE SERVER への KVM 接続の追加

KVM を Satellite のコンピュートリソースとして追加するには、次の手順を使用します。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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** に移動して、**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、新規コンピュートリソースの名前を入力します。
3. **プロバイダー** リストから **Libvirt** を選択します。
4. **説明** フィールドには、コンピュート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。KVM サーバーで実行した以下のコマンドの出力から、ゲストの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 手順

- コンピュートリソースを作成するには、**hammer compute-resource create** コマンドを入力します。

```
# 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"
```

## 9.3. 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には、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

Satellite 6 で管理できるのは、ディレクトリープールストレージタイプのみである点に注意してください。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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、KVM 接続の名前をクリックします。
2. **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. **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/libvirt/images/TestImage.qcow2
```

9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
10. **送信** をクリックしてイメージの詳細を保存します。

## CLI 手順

- **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 "KVM Image" \  
--compute-resource "My_KVM_Server" \  
--operatingsystem "RedHat version" \  
--architecture "x86_64" \  
--username root \  
--user-data false \  
--uuid "/var/lib/libvirt/images/KVMImage.qcow2" \  

```

## 9.4. KVM の詳細のコンピュータプロファイルへの追加

この手順を使用して、KVM ハードウェア設定をコンピューティングプロファイルに追加します。このコンピュータプロファイルを使用して KVM でホストを作成すると、これらの設定が自動的に入力されます。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. KVM コンピュートリソースの名前をクリックします。
4. **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。
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を実行する場合には、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イス** リストから、ホストのネットワークインターフェイ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イスは、複数作成することができますが、少なくとも1つのインターフェイスが 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。
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。
9. **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 手順

1. 以下のコマンドを実行してコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name "Libvirt CP"
```

2. コンピュートプロファイルの値を追加するには、以下のコマンドを入力します。

```
# hammer compute-profile values create --compute-profile "Libvirt CP" \
--compute-resource "My_KVM_Server" \
--interface
"compute_type=network,compute_model=virtio,compute_network=examplenetwork" \
--volume "pool_name=default,capacity=20G,format_type=qcow2" \
--compute-attributes "cpus=1,memory=1073741824"
```

## 9.5. KVM 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路または、既存のイメージをもとに、ホストを作成できます。

- ネットワーク接続経路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にトリガーされます。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。

- 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は、KVM サーバーが新規ボリュームのベースとして既存のイメージを使用し、仮想マシンを作成するようトリガーします。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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

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. **インターフェイス** タブをクリックして、ホストのインターフェイスで **編集** をクリックします。
8. 特に以下の項目など、フィールドに自動的に入力されていることを確認します。
  - **ホスト** タブの **名前** は **DNS 名** になります。
  - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** フィールドは空白です。KVM サーバーが MAC アドレスをホストに割り当てます。
  - ホストの最初のインターフェイスで **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。選択されていない場合は、それらを選択してください。
  - KVM 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。
9. **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。
10. **Provisioning Method** のページで、使用するホストを選択します。
  - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
  - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
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、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ます。必要に応じてこれらの設定を変更します。
13. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
14. **送信** をクリックしてホストエントリーを保存します。

## CLI手順

- ネットワークベースのプロビジョニングを使用するには、**--provision-method build** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \
--name "kvm-host1" \
--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=exam
plenetwork" \
--compute-attributes="cpus=1,memory=1073741824" \
--volume="pool_name=default,capacity=20G,format_type=qcow2" \
--root-password "password"
```

- イメージベースのプロビジョニングを使用するには、**--provision-method image** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \
--name "kvm-host2" \
--organization "My_Organization" \
--location "New York" \
--hostgroup "Base" \
--compute-resource "My_KVM_Server" \
--provision-method image \
--image "KVM Image" \
--enabled true \
--managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=network,compute_network=exampl
enetwork" \
--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、**hammer host create --help** コマンドを入力します。

## 第10章 RED HAT VIRTUALIZATION での仮想マシンのプロビジョニング

Red Hat Virtualization は、エンタープライズレベルのサーバーおよびデスクトップ仮想化プラットフォームです。Red Hat Satellite では、Red Hat Virtualization の REST API 経由で仮想化機能を管理できます。これには、仮想マシンの作成および電源状態の制御が含まれます。

Red Hat Virtualization のプロビジョニングを使用して、ネットワーク接続経由か、既存のイメージをもとに仮想マシンを作成できます。

**cloud-init** を使用して、プロビジョニングする仮想マシンを設定できます。**cloud-init** を使用すると、管理対象の DHCP と TFTP など、ネットワークで特別な設定を回避し、仮想マシンのインストールを完了できます。この手法では、Satellite は、SSH を使用して終了スクリプトを実行し、プロビジョニングした仮想マシンに接続する必要はありません。

### 前提条件

- Red Hat Enterprise Linux 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。
- 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。
- 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、[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の、**blank** 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、[Virtual Machine Management Guideの Templates](#) を参照してください。
- Satellite Server との通信に使用する Red Hat Virtualization で管理者に相当するユーザー。この通信には、**admin@internal** ユーザーを使用しないでください。代わりに、以下のパーミッションで新しい Red Hat Virtualization ユーザーを作成します。
  - 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 > 作成
  - ネットワーク > vNIC プロファイルの設定 > プロパティの編集
  - ネットワーク > vNIC プロファイル > 削除
  - ネットワーク > vNIC プロファイル > vNIC プロファイルの仮想マシンへの割り当て
  - ネットワーク > vNIC プロファイル > 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 > 操作のプロビジョニング > インポート/エクスポート
  - VM > 操作のプロビジョニング > 作成
  - VM > 操作のプロビジョニング > 削除
  - VM > 操作のプロビジョニング > インポート/エクスポート
  - VM > 操作のプロビジョニング > ストレージの編集



- ディスク > 操作のプロビジョニング > 作成
- ディスク > ディスクプロファイル > ディスクプロファイルのアップロード  
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 Administration Guide](#)の [Administering User Tasks From the Administration Portal](#) を参照してください。

## 手順の概要

1. 「[Satellite Server への Red Hat Virtualization 接続の追加](#)」
2. オプション: 「[Red Hat Virtualization での Cloud-init イメージの準備](#)」 **cloud-init** を使用してイメージベースの仮想マシンを設定するには、この手順を使用します。
3. オプション: 「[Satellite Server への Red Hat Virtualization 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する場合にはこの手順を使用します。
4. オプション: 「[cloud-init テンプレートの準備](#)」 **cloud-init** を使用してイメージベースの仮想マシンを設定するには、この手順を使用します。
5. 「[Red Hat Virtualization の詳細のコンピュートプロファイルへの追加](#)」
6. 「[Red Hat Virtualization でのホストの作成](#)」

## 10.1. SATELLITE SERVER への RED HAT VIRTUALIZATION 接続の追加

この手順を使用して、Red Hat Virtualization を Satellite のコンピュートリソースとして追加します。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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](#) に移動して、[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、新規コンピュート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、コンピュートリソースの説明を入力します。
5. **URL** フィールドに、Red Hat Virtualization Manager の API の接続 URL を <https://rhv.example.com/ovirt-engine/api/v4> の形式で入力します。
6. **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。
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、[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。
9. [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。
10. **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。

11. **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。
12. **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。
13. **組織** タブをクリックして、使用する組織を選択します。
14. **送信** をクリックしてコンピュータリソースを保存します。

## CLI 手順

- **--provider** で **Ovirt** を、**--datacenter** で使用するデータセンター名を指定して、**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 rhv.example.com" \
--url "https://rhv.example.com/ovirt-engine/api" \
--user "Satellite_User" --password "My_Password" \
--locations "New York" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

## 10.2. RED HAT VIRTUALIZATION での CLOUD-INIT イメージの準備

プロビジョニング時に **cloud-init** を使用するには、Red Hat Virtualization にインストールした **cloud-init** でイメージを準備してから、イメージを Satellite にインポートしてプロビジョニングに使用してください。

### 手順

1. Red Hat Virtualization で、仮想マシンを作成して、Satellite でイメージベースのプロビジョニングに使用します。
2. 仮想マシンで **cloud-init** をインストールします。

```
yum install cloud-init
```

3. **/etc/cloud/cloud.cfg** ファイルに、以下の情報を追加します。

```
datasource_list: ["NoCloud", "ConfigDrive"]
```

4. Red Hat Virtualization で、この仮想マシンからイメージを作成します。

このイメージを Satellite に追加するには、**User Data** チェックボックスを選択します。

## 10.3. SATELLITE SERVER への RED HAT VIRTUALIZATION 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には、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 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、Red Hat Virtualization 接続の名前をクリックします。
2. **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. **イメージ** リストの Red Hat Virtualization コンピュートリソースからイメージを選択します。
9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
10. **送信** をクリックしてイメージの詳細を保存します。

## CLI 手順

- **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 "RHV_Image" \
--compute-resource "My_RHV" \
--operatingsystem "RedHat version" \
--architecture "x86_64" \
--username root \
--uuid "9788910c-4030-4ae0-bad7-603375dd72b1" \
```

## 10.4. CLOUD-INIT テンプレートの準備

## 手順

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、**テンプレートの作成** をクリックします。
2. **名前** フィールドには、テンプレートの名前を入力します。
3. **Editor** フィールドで、以下のテンプレートの情報を入力します。

```
<%#
kind: user_data
name: Cloud-init
-%>
#cloud-config
hostname: <%= @host.shortname %>
```

```

<%=# Allow user to specify additional SSH key as host paramter -%>
<% if @host.params['sshkey'].present? ||
@host.params['remote_execution_ssh_keys'].present? -%>
ssh_authorized_keys:
<% if @host.params['sshkey'].present? -%>
  - <%= @host.params['sshkey'] %>
<% end -%>
<% if @host.params['remote_execution_ssh_keys'].present? -%>
<% @host.params['remote_execution_ssh_keys'].each do |key| -%>
  - <%= key %>
<% end -%>
<% end -%>
<% end -%>
runcmd:
  - |
    #!/bin/bash
<%= indent 4 do
  snippet 'subscription_manager_registration'
end %>
<% if @host.info['parameters']['realm'] && @host.realm && @host.realm.realm_type == 'Red
Hat Identity Management' -%>
  <%= indent 4 do
    snippet 'idm_register'
  end %>
<% end -%>
<% unless @host.operatingsystem.atomic? -%>
  # update all the base packages from the updates repository
  yum -t -y -e 0 update
<% end -%>
<%
  # safemode renderer does not support unary negation
  non_atomic = @host.operatingsystem.atomic? ? false : true
  pm_set = @host.puppetmaster.empty? ? false : true
  puppet_enabled = non_atomic && (pm_set || @host.params['force-puppet'])
%>
<% if puppet_enabled %>
  yum install -y puppet
  cat > /etc/puppet/puppet.conf << EOF
  <%= indent 4 do
    snippet 'puppet.conf'
  end %>
  EOF
  # Setup puppet to run on system reboot
  /sbin/chkconfig --level 345 puppet on

  /usr/bin/puppet agent --config /etc/puppet/puppet.conf --onetime --tags no_such_tag <%=
@host.puppetmaster.blank? ? " : "--server #{@host.puppetmaster}" %> --no-daemonize
  /sbin/service puppet start
<% end -%>
phone_home:
  url: <%= foreman_url('built') %>
  post: []
  tries: 10pp

```

4. **タイプ** タブをクリックして、**タイプ** リストから**ユーザーデータのテンプレート** を選択します。
5. **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、このテンプレートに関連付け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、**ロケーション** 一覧から、テンプレートに関連付けるロケーションを選択します。
7. **組織** タブをクリックして、**組織** 一覧から、テンプレートに関連付ける組織を選択します。
8. **Submit** をクリックします。
9. **ホスト > オペレーティングシステム** に移動し、テンプレートに関連付ける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、**ユーザーデータテンプレート** リストから、**新規テンプレート** 名を選択します。
11. **Submit** をクリックします。

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

この手順を使用して、Red Hat Virtualization ハードウェア設定をコンピューティングプロファイルに追加します。このコンピュープロファイルを使用して KVM でホストを作成すると、これらの設定が自動的に入力されます。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

1. Satellite Web UI で、**インフラストラクチャー > コンピュープロファイル** に移動します。
2. コンピュープロファイルウィンドウで、既存のコンピュープロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Red Hat Virtualization コンピュートリソースの名前をクリックします。
4. **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。
5. **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。
6. **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。
7. **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。
8. **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。
9. **ネットワークインターフェイス** エリアには、ホストのネットワークインターフェイ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イスは、複数作成することができますが、少なくとも1つのインターフェイスが 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。ネットワークインターフェイスごとに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イ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。
10. **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。
  11. **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 手順

1. 以下のコマンドを実行してコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name "Red Hat Virtualization CP"
```

2. コンピュートプロファイルの値を設定するには、以下のコマンドを入力します。

```
# hammer compute-profile values create --compute-profile "Red Hat Virtualization CP" \
--compute-resource "My_RHV" \
--interface
"compute_interface=Interface_Type,compute_name=eth0,compute_network=satnetwork" \
--volume "size_gb=20G,storage_domain=Data,bootable=true" \
--compute-attributes "cluster=Default,cores=1,memory=1073741824,start=true"
```

## 10.6. RED HAT VIRTUALIZATION でのホストの作成

Satellite では Red Hat Virtualization のプロビジョニングを使用して、ネットワーク接続経由か、既存のイメージをもとに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 仮想ネットワーク上にある 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 サーバーがトリガーされ、仮想マシンを作成して起動します。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。
- 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、Red Hat Virtualization サーバーが新規ボリュームのベースとして既存のイメージを使用し、仮想マシンを作成するようトリガーします。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## DHCP の競合

ネットワークベースのプロビジョニングでは、Red Hat Virtualization 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

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. **インターフェイス** タブをクリックして、ホストのインターフェイスで **編集** をクリックします。
8. 特に以下の項目など、フィールドに自動的に入力されていることを確認します。
  - **ホスト** タブの **名前** は **DNS 名** になります。
  - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。
  - **MAC アドレス** フィールドは空白です。Red Hat Virtualization サーバーが MAC アドレスをホストに割り当てます。
  - ホストの最初のインターフェイスで **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。選択されていない場合は、それらを選択してください。
  - 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。
9. **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。
10. **Provisioning Method** のページで、使用するホストを選択します。
  - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
  - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
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、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ます。必要に応じてこれらの設定を変更します。
13. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。

14. **送信** をクリックしてホストエントリーを保存します。

### CLI 手順

- ネットワークベースのプロビジョニングを使用するには、**--provision-method build** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \  
--name "RHV-vm1" \  
--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を使用するには、**--provision-method image** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \  
--name "RHV-vm2" \  
--organization "My_Organization" \  
--location "New York" \  
--hostgroup "Base" \  
--compute-resource "My_RHV" \  
--provision-method image \  
--image "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、**hammer host create --help** コマンドを入力します。



## 第11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。

### 11.1. 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。

- 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、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、[3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)。
- Red Hat Enterprise Linux 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。
- 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. 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でください。その代わりに、以下のパーミッションを持つユーザーを作成してください。

VMware vCenter Server バージョン 6.10 の場合は、以下のパーミッションを設定します。

- All Privileges (すべての権限) → Datastore (データストア) → Allocate Space (領域の割り当て)、Browse datastore (データストアの参照)、Update Virtual Machine files (仮想マシンファイルの更新)、Low level file operations (ローレベルのファイル操作)
- 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) → Change Config (All) (設定の変更 (すべて))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Interaction (All) (対話 (すべて))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Edit Inventory (インベントリ編集)
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Provisioning (All) (プロビジョニング (すべて))

同じ手順が VMware vCenter Server バージョン 7.0 にも適用されることに注意してください。

VMware vCenter Server バージョン 6.5 の場合には、次のパーミッションを設定します。

- All Privileges (すべての権限) → Datastore (データストア) → Allocate Space (領域の割り当て)、Browse datastore (データストアの参照)、Update Virtual Machine files (仮想マシンファイルの更新)、Low level file operations (ローレベルのファイル操作)
- 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 (All) (対話 (すべて))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Inventory (All) (インベントリー (All))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Provisioning (All) (プロビジョニング (すべて))

### 11.3. 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。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

Satellite が TCP ポート 443 を介して vCenter と通信できるように、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

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。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タ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、「[コンピュータ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 手順

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

```
# hammer compute-resource create --name "My_vSphere" \
--provider "Vmware" \
--description "vSphere server at vsphere.example.com" \
--server "vsphere.example.com" --user "My_User" \
--password "My_Password" --locations "My_Location" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

## 11.4. 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。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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** に移動して、コンピュータ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 手順

- **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"
```

## 11.5. 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。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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、**コンピュータプロファイル** のウィンドウで **コンピュータプロファイル** 名前をクリックしてから **vSphere 接続** 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イス** リストから、ホストのネットワークインターフェイ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イスは、複数作成することができますが、少なくとも1つのインターフェイスが 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。
14. Eager Zero シックプロビジョニングを使用する必要がある場合は、**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トプロファイルを保存します。

## CLI 手順

1. 以下のコマンドを実行してコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name "VMWare CP"
```

2. コンピュートプロファイルの値を追加するには、以下のコマンドを入力します。

```
# hammer compute-profile values create --compute-profile "VMWare CP" \
--compute-resource "My_vSphere" \
--interface "compute_type=VirtualE1000,compute_network=mynetwork" \
--volume "size_gb=20G,datastore=Data,name=myharddisk,thin=true" \
--compute-attributes
"cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=true"
```

## 11.6. 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。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。

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す。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 手順

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、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. **インターフェイス** タブをクリックして、ホストのインターフェイスで **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イスで **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. インターフェイスウィンドウには、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. 必要なプロビジョニング方法を選択します。
  - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
  - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
  - ブートディスクベースのプロビジョニングの場合、**ブートディスクベース** をクリックします。
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、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ます。要件に合わせてこれらの設定を変更します。
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。存在しない場合には、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 手順

- **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=my
network" \
--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"
```

- **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_VMware" --provision-method image \
--image "Test VMware Image" --enabled true --managed true \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=my
network" \
--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、**hammer host create --help** コマンドを入力します。

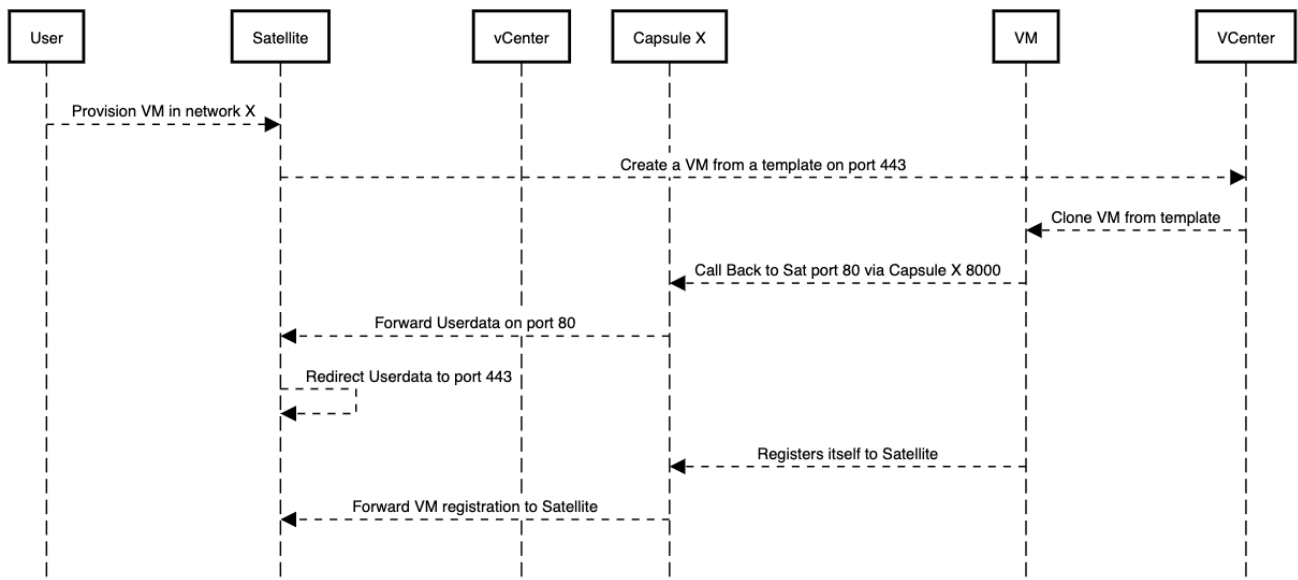
## 11.7. プロビジョニングでの VMWARE VSPHERE CLOUD-INIT と USERDATA テンプレートの使用

**Cloud-init** と **userdata** テンプレートと VMware を併用して、ユーザーデータを新規の仮想マシンに挿入し、VMware をさらにカスタマイズして、VMware がホストする仮想マシンを使用して Satellite にコールバックできるようになります。

同じ手順を使用して、Satellite 内で、ワークフローに変更をいくつか加えて、VMware コンピュータリソースを設定できます。

### VMware cloud-init プロビジョニングの概要

## VMware Template Provisioning



Satellite で VMware プロビジョニング用にコンピュータリソースとイメージを設定する場合に、プロビジョニングのイベントは以下の順番に発生します。

- ユーザーが Satellite Web UI、API、または Hammer を使用して1つ以上の仮想マシンをプロビジョニングする。
- Satellite が VMware vCenter を呼び出して仮想マシンテンプレートのクローンを作成する。
- Satellite **userdata** のプロビジョニングテンプレートがカスタマイズされたアイデンティティ情報を追加する。
- プロビジョニングが完了すると、**cloud-init** プロビジョニングテンプレートが仮想マシンに対して、**cloud-init** の実行時に Capsule にコールバックするように指示を出す。
- VMware vCenter がこのテンプレートのクローンを仮想マシンに作成する。
- VMware vCenter がホスト名、IP、DNS など、仮想マシンの ID のカスタマイズを適用する。
- 仮想マシンがビルドされて、**cloud-init** が呼びされ、ポート **80** で Satellite にコールバックしてから、**443** にリダイレクトする。

### ポートとファイアウォールの要件

**cloud-init** サービスがあるので、仮想マシンを Capsule に登録している場合でさえも、仮想マシンは常に Satellite に対してコールバックします。必要な接続ができるように、ポートとファイアウォールを設定してください。

ポートとファイアウォールの要件に関する詳細は、[オンラインネットワークからの Satellite Server のインストールのポートとファイアウォールの要件](#) および [Capsule Server のインストールのポートとファイアウォールの要件](#) を参照してください。

### userdata および Cloud-init テンプレートとオペレーティングシステムの関連付け

1. Satellite Web UI で、**ホスト > オペレーティングシステム** に移動し、プロビジョニングに使用するオペレーティングシステムを選択します。
2. **テンプレート** タブをクリックします。



3. **Cloud-init** テンプレート リストから **Cloudinit デフォルト** を選択します。
4. **User data** テンプレート リストから **UserData open-vm-tools** を選択します。
5. **送信** をクリックして変更を保存します。

### cloud-init テンプレートを使用するためのイメージの準備

イメージを準備するには、先に仮想マシンに必要な設定を行ってからでないと、Satellite で使用するイメージとして保存できません。

プロビジョニングに **cloud-init** テンプレートを使用するには、**cloud-init** をインストールして、有効化し、Satellite Server に対してコールバックを行うように、仮想マシンを設定する必要があります。

セキュリティ上の理由から CA 証明書をインストールして、全通信に HTTPS を使用するようになる必要があります。この手順には、不要な情報がプロビジョニングに使用するイメージに転送されないように、仮想マシンをクリーンアップする手順が含まれます。

**cloud-init** が含まれるイメージの場合は、**cloud-init** がデフォルトで無効になっているので、**cloud-init** が Satellite と通信できるように以下の手順を実行する必要があります。

1. イメージの作成に使用する仮想マシンで、**cloud-init**、**open-vm-tools** および **perl** をインストールします。

```
# yum -y install cloud-init open-vm-tools perl
```

2. **cloud-init** によるネットワーク設定を無効にします。

```
# cat << EOM > /etc/cloud/cloud.cfg.d/01_network.cfg
network:
  config: disabled
EOM
```

3. Satellite からデータをフェッチするように **cloud-init** を設定します。

```
# cat << EOM > /etc/cloud/cloud.cfg.d/10_datasource.cfg
datasource_list: [NoCloud]
datasource:
  NoCloud:
    seedfrom: https://satellite.example.com/userdata/
EOM
```

4. **cloud-init** で使用するモジュールを設定します。

```
# cat << EOM > /etc/cloud/cloud.cfg
cloud_init_modules:
  - bootcmd

cloud_config_modules:
  - runcmd

cloud_final_modules:
  - scripts-per-once
  - scripts-per-boot
  - scripts-per-instance
  - scripts-user
```

```
- phone-home

system_info:
  distro: rhel
  paths:
    cloud_dir: /var/lib/cloud
    templates_dir: /etc/cloud/templates
  ssh_svcname: sshd
EOM
```

5. イメージの CA 証明書を有効にします。

```
# update-ca-trust enable
```

6. Satellite Server から **katello-server-ca.crt** ファイルをダウンロードします。

```
# wget -O /etc/pki/ca-trust/source/anchors/cloud-init-ca.crt
http://satellite.example.com/pub/katello-server-ca.crt
```

7. 証明書の記録を更新するには、次のコマンドを入力します。

```
# update-ca-trust extract
```

8. 次のコマンドを使用して、イメージを消去します。

```
# systemctl stop rsyslog
# systemctl stop auditd
# package-cleanup --oldkernels --count=1
# yum clean all
```

9. ログスペースの削減、古いログの削除、ログの省略には、以下のコマンドを使用します。

```
# logrotate -f /etc/logrotate.conf
# rm -f /var/log/*-???????? /var/log/*.gz
# rm -f /var/log/dmesg.old
# rm -rf /var/log/anaconda
# cat /dev/null > /var/log/audit/audit.log
# cat /dev/null > /var/log/wtmp
# cat /dev/null > /var/log/lastlog
# cat /dev/null > /var/log/grubby
```

10. **udev** ハードウェアルールを削除します。

```
# rm -f /etc/udev/rules.d/70*
```

11. 既存のネットワーク設定に関連する **ifcfg** スクリプトを削除します。

```
# rm -f /etc/sysconfig/network-scripts/ifcfg-ens*
# rm -f /etc/sysconfig/network-scripts/ifcfg-eth*
```

12. SSH ホストキーを削除します。

```
# rm -f /etc/ssh/SSH_keys
```

13. root ユーザーの SSH 履歴を削除します。

```
# rm -rf ~root/.ssh/known_hosts
```

14. root ユーザーの Shell 履歴を削除します。

```
# rm -f ~root/.bash_history
# unset HISTFILE
```

この仮想マシンからイメージを作成できるようになりました。

Satellite にイメージを追加するには、「[Satellite Server への VMware vSphere イメージの追加](#)」のセクションを使用してください。

### ユーザーデータテンプレートを転送するための Capsule の設定

Capsule テンプレート機能を使用して Satellite をデプロイする場合に、X-Forwarded-For HTTP ヘッダーで転送されたホストの IP アドレスを認識して正しいテンプレートペイロードを提供するように Satellite を設定する必要があります。

セキュリティ上の理由から Satellite はローカルホストからのみ、この HTTP ヘッダーを認識します。個別の Capsule ごとに、ホストの IP アドレスを認識するための正規表現を設定する必要があります。Web UI で **管理 > 設定 > プロビジョニング** に移動して、**リモートアドレス** の設定を変更します。CLI では、以下のコマンドを入力して設定できます。

```
# hammer settings set --name remote_addr --value '(localhost(4|6|4to6)?|192.168.122.(1|2|3))'
```

## 11.8. コンピュートリソースのキャッシング

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

### 11.8.1. コンピュートリソースのキャッシングの有効化

コンピュートリソースのキャッシング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。

### 11.8.2. コンピュートリソースのキャッシュのリフレッシュ

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

#### 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 手順

この API の呼び出しを使用して、コンピュータ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
```

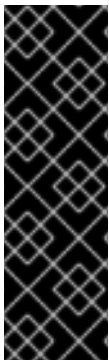
**hammer compute-resource list** コマンドを使用して、コンピュータリソースのキャッシュをリフレッシュする VMware サーバーの ID を判断します。

## 第12章 CONTAINER-NATIVE VIRTUALIZATION を使用した仮想マシンのプロビジョニング

Container-native Virtualization は、Kubernetes を採用済みまたは採用予定ではあるものの、簡単にコンテナ化できない既存の仮想マシンワークロードもある開発チームのニーズに対応します。この技術を使用すると、開発プラットフォームを統一して、開発者はアプリケーションコンテナ内にあるアプリケーションや、共有環境にある仮想マシンをビルド、変更、デプロイできます。これらの機能は、オープンなハイブリッドクラウドで、迅速にアプリケーションの最新化を図るサポートをします。

Red Hat Satellite では、Container-native Virtualization 向けのコンピュートリソースを作成し、Satellite を使用して Kubernetes 仮想マシンをプロビジョニングおよび管理できます。

今回のリリースでは、テンプレートのプロビジョニングはサポートされていません。



### 重要

Container-native Virtualization のコンピュートリソースは、テクノロジープレビュー機能のみとなっています。テクノロジープレビュー機能は、Red Hat の実稼働サービスレベルアグリーメント (SLA) でサポートされておらず、機能的に完全でない可能性があります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビューの機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、[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。

### 前提条件

- Red Hat Enterprise Linux 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。
- 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。
- Container-native Virtualization ユーザーを用意し、OpenShift Container Platform 仮想クラスターの **cluster-admin** パーミッションを割り当てる。詳細は、OpenShift Container Platform ドキュメントの [認証](#) ガイドの [RBAC を使用したパーミッションの定義および適用](#) を参照してください。
- コンテナネイティブ仮想化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、[ネットワークの設定](#) を参照します。
- 以下のロールが割り当てられた Satellite ユーザーアカウント：
  - **Edit hosts**
  - **View hosts**  
詳細は、[Red Hat Satellite の管理ガイド](#) の [ユーザーへのロールの割り当て](#) を参照してください。
- 以下のパーミッションが指定された Satellite のカスタムロール
  - **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、Red Hat Satellite の管理ガイドの [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 の管理ガイドの [ロールへのパーミッションの追加](#) を参照してください。

## 12.1. SATELLITE SERVER への CONTAINER-NATIVE VIRTUALIZATION 接続の追加

この手順を使用して、Container-native Virtualization を Satellite のコンピュートリソースとして追加します。

### 手順

1. 以下の `satellite-installer` コマンドを入力して、Satellite の Container-native Virtualization プラグインを有効にします。

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

2. HTTP と HTTPS 認証に使用するベアータークンを生成します。Container-native Virtualization サーバーで、トークンを含むシークレットを表示します。

```
# kubectl get secrets
```

3. シークレットのトークンをリストします。

```
# kubectl get secrets YOUR_SECRET -o jsonpath='{.data.token}' | base64 -d | xargs
```

このトークンを書き留めてください。この手順の後半で使用します。

4. Satellite Web UI で **インフラストラクチャー > コンピュートリソース** に移動して、**コンピュートリソースの作成** をクリックします。
5. **名前** フィールドには、新規コンピュートリソースの名前を入力します。
6. **プロバイダー** の一覧から **KubeVirt** を選択します。
7. **説明** フィールドには、コンピュートリソースの説明を入力します。
8. **ホスト名** フィールドで、使用する Container-native Virtualization サーバーのアドレスを入力します。
9. **API ポート** フィールドに、Satellite から Container-native Virtualization へのプロビジョニング要求に使用するポート番号を入力します。
10. **Namespace** フィールドで、使用する Container-native Virtualization の仮想クラスターのユーザー名を入力します。

11. トークン フィールドに、HTTP および HTTPS 認証向けのベアータークンを入力します。
12. オプション: X509 認証局 フィールドに、API サーバー呼び出しのクライアントの証明書認証を有効にする証明書を入力します。

## 第13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、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利用できます。Red Hat Satellite 6 は、Red Hat OpenStack Platforms REST API と対話し、クラウドインスタンスを作成して、電源管理の状態を制御することができます。

### 前提条件

- Red Hat Enterprise Linux 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。
- 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。
- OpenStack 環境でネットワークを管理する Capsule Server。詳細は、[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 Instances and Images Guide](#) を参照してください。

### 手順の概要

1. [「Satellite Server への Red Hat OpenStack Platform 接続の追加」](#)
2. [「Satellite Server への Red Hat OpenStack Platform イメージの追加」](#)
3. [「Red Hat OpenStack Platform の詳細のコンピュータプロファイルへの追加」](#)
4. [「Red Hat OpenStack Platform でのイメージベースのホストの作成」](#)

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

この手順を使用して、Red Hat OpenStack Platform を Satellite のコンピュートリソースとして追加します。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

#### 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** に移動して、**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、新規コンピュートリソースの名前を入力します。
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、コンピュート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/v3.0/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 手順

- コンピュートリソースを作成するには、**hammer compute-resource create** コマンドを入力します。

```
# hammer compute-resource create --name "My_OpenStack" \
--provider "OpenStack" \
--description "My OpenStack environment at openstack.example.com" \
--url "http://openstack.example.com:5000/v3.0/tokens" --user "My_Username" \
--password "My_Password" --tenant "openstack" --locations "New York" \
--organizations "My_Organization"
```

## 13.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM 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には、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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. **イメージ** リストの Red Hat OpenStack コンピュートリソースからイメージを選択します。
9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
10. **送信** をクリックしてイメージの詳細を保存します。

## CLI 手順

- **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 "OpenStack Image" \  
--compute-resource "My_OpenStack_Platform" \  
--operatingsystem "RedHat version" \  
--architecture "x86_64" \  
--username root \  
--user-data true \  
--uuid "/path/to/OpenstackImage.qcow2"
```

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

この手順を使用して、Red Hat OpenStack Platform ハードウェア設定をコンピューティングプロファイルに追加します。このコンピューティングプロファイルを使用して Red Hat OpenStack Platform でホストを作成すると、これらの設定が自動的に入力されます。

### 手順

1. Satellite Web UI で、**インフラストラクチャー > コンピュータープロファイル** に移動します。
2. コンピュータープロファイルウィンドウで、既存のコンピュータープロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Red Hat OpenStack Platform コンピュートリソースの名前をクリックします。
4. **フレーバー** リストから、ホストに使用する Red Hat OpenStack Platform のハードウェアのプロファイルを選択します。
5. **アベイラビリティゾーン** リストから Red Hat 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。
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. **テナント** リストから、Red Hat 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。
8. **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。
9. **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。

10. **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。
11. **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。
12. **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。
13. **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手順

Red Hat Satellite 6.10 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。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。

## 13.4. 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 プロビジョニングを使用して、既存のイメージからホストを作成できます。新規ホストのエントリーは、Red Hat OpenStack Platform サーバーが新規ボリュームのベースとして既存のイメージを使用し、インスタンスを作成するようトリガーします。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

## 手順

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 OpenStack Platform 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. **ライフサイクル環境** 一覧から、環境を選択します。
8. **インターフェイス** タブをクリックして、ホストのインターフェイスで **編集** をクリックします。
9. 特に以下の項目など、フィールドに自動的に入力されていることを確認します。
  - **ホスト** タブの **名前** は **DNS 名** になります。
  - **MAC アドレス** フィールドは空白です。Red Hat OpenStack Platform は、プロビジョニング中に MAC アドレスをホストに割り当てます。
  - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。

- ホストの最初のインターフェイスで **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。選択されていない場合は、それらを選択してください。
10. **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。
  11. 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。
  12. **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。
  13. **仮想マシン** タブをクリックして、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ます。必要に応じてこれらの設定を変更します。
  14. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
  15. **送信** をクリックしてホストエントリーを保存します。

## CLI手順

- **--provision-method image** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \  
--name "openstack-host1" \  
--organization "My_Organization" \  
--location "New York" \  
--hostgroup "Base" \  
--compute-resource "My_OpenStack_Platform" \  
--provision-method image \  
--image "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、**hammer host create --help** コマンドを入力します。

## 第14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。Satellite 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。

### 14.1. 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。

- 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)。
- Red Hat Enterprise Linux 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。
- 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。

### 14.2. 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。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

#### 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の設定を使用します。Satellite Server の時間が正しく同期されていることを確認します。さらに、`ntpd` または `chronyd` などの NTP サービスが Satellite Server で正しく実行されていることを確認します。Amazon Web Services に正しい時間を指定できないと、認証に失敗する可能性があります。

Satellite での時間の同期に関する詳細は、[オンラインネットワークからの Satellite Server のインストールの chronyd とシステムクロックの同期](#) を参照してください。

#### 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. オプション: **HTTP プロキシ** リストから、HTTP プロキシを選択して、外部の API サービスに接続します。HTTP プロキシを Satellite に追加してから、リストからプロキシを選択する必要があります。詳細は、[「コンピュートリソースと HTTP プロキシとの連携」](#) 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. オプション: **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。
12. 新規コンピュートリソースを選択して、**SSH キー** タブをクリックし、**ダウンロード** をクリックして、SSH 認証に使用するために SSH キーのコピーを保存します。[BZ1793138](#) が解決するまで、Amazon EC2 コンピュートリソースのコピーの作成直後に SSH キーのコピーをダウンロードします。後で SSH キーが必要な場合には、[「SSH での Amazon EC2 インスタンスへの接続」](#) の手順を実行してください。

## CLI 手順

- **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"
```

## 14.3. コンピュートリソースと HTTP プロキシとの連携

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 プロキシを作成して、その HTTP 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 HTTP 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別の HTTP プロキシを追加した場合には、**管理 > 設定** で定義した HTTP プロキシが優先されます。

### 手順

1. Satellite Web UI で、**インフラストラクチャー > HTTP プロキシ** に移動して、**新規 HTTP プロキシ** を選択します。
2. **名前** フィールドで、HTTP プロキシの名前を入力します。
3. **URL** フィールドに、ポート番号を含む HTTP プロキシの URL を入力します。
4. オプション: HTTP プロキシに認証が必要な場合には、HTTP 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 HTTP 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. **Submit** をクリックします。

## 14.4. 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ロケーションが含まれます。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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. オプション: 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場合には、**ユーザーデータ** チェックボックスを選択します。ユーザーデータを有効にすると、Finish スクリプトは自動的に無効になります。これは、逆の場合にも当てはまります。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. オプション: **IAM ロール** リストから、イメージの作成に使用する Amazon のセキュリティロールを入力します。
11. **送信** をクリックしてイメージの詳細を保存します。

### CLI 手順

- **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"
```

## 14.5. 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 > コンピュータプロファイル** 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. **アベイラビリティゾーン** 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タプロファイルを保存します。

### CLI 手順

コンピュータプロファイルの CLI コマンドは、Red Hat Satellite 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。

## 14.6. 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ホストを作成します。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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. **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. **コンピュータプロファイル** 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. **インターフェイス** タブをクリックしてから、ホストのインターフェイスの **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、ホストの最初のインターフェイスの IP アドレスと **Managed**、**Primary**、および **Provision** のオプションを自動的に選択します。
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入力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、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、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のイメージを使用し、インスタンスを作成するようトリガーします。

### CLI 手順

- **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に対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

## 14.7. 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。

### 手順

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 を書き留めます。

```
# 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
```

## 14.8. 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。

SSH で finish テンプレートを使用する場合は、Satellite は EC2 環境内に存在して、適切なセキュリティグループに存在する必要があります。現在、Satellite は Capsule Server を使用しないで直接 SSH finish プロビジョニングを実行します。Satellite Server が EC2 内にはない場合は、EC2 仮想マシンは到達可能な必要な外部 IP ではなく内部 IP を報告します。

### 手順

1. Red Hat 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** 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-proxy/.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。

## 14.9. 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15章 GOOGLE COMPUTE ENGINE でのクラウドインスタンスのプロビジョニング

Red Hat Satellite は、新規仮想マシンの作成、電源管理の状態の制御など、Google Compute Engine (GCE) と対話できます。GCE ホストの作成には、Satellite では Red Hat がサポートするゴールデンイメージのみを使用できます。

### 前提条件

- Red Hat Enterprise Linux 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。
- 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。
- GCE プロジェクトで、必要な IAM Compute ロールでサービスアカウントを設定します。詳細は、GCE ドキュメントの [Compute Engine IAM roles](#) を参照してください。
- GCE プロジェクト全体のメタデータで、**enable-oslogin** を **FALSE** に設定します。詳細は、GCE ドキュメントの [Enabling or disabling OS Login](#) を参照してください。
- オプション: GCE ホストで Puppet を使用するには、[管理 > 設定 > Puppet](#) に移動して、**証明書での UUID の使用** 設定を有効にして、Puppet が一貫性のある Puppet 証明書 ID を使用するよう設定します。
- ニーズに合わせて **finish** または **user\_data** プロビジョニングテンプレートと使用するオペレーティングシステムを関連付けます。プロビジョニングテンプレートの詳細は、[プロビジョニングテンプレート](#) を参照してください。

### 手順の概要

1. [「Satellite Server への Google Compute Engine 接続の追加」](#)
2. [「Satellite Server への Google Compute Engine イメージの追加」](#)
3. [「Google Compute Engine の詳細のコンピューティングプロファイルへの追加」](#)
4. [「Google Compute Engine でのイメージベースのホストの作成」](#)

## 15.1. SATELLITE SERVER への GOOGLE COMPUTE ENGINE 接続の追加

この手順を使用して、Google Compute Engine (GCE) を Satellite のコンピュートリソースとして追加します。Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

1. GCE では、JSON 形式でサービスアカウントキーを生成し、このファイルを Satellite Server の `/usr/share/foreman/` ディレクトリにアップロードします。
2. Satellite Server で、サービスアカウントキーの所有者を **foreman** ユーザーに変更します。

```
# chown foreman /usr/share/foreman/gce_key.json
```

3. サービスアカウントキーのパーミッションを設定して、ファイルを読み取れるようにします。

```
# chmod 0600 /usr/share/foreman/gce_key.json
```

4. サービスアカウントキーの SELinux コンテキストを復元します。

```
# restorecon -vv /usr/share/foreman/gce_key.json
```

5. 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作成** をクリックします。
6. **名前** フィールドには、コンピュートリソースの名前を入力します。
7. **プロバイダー** の一覧から **Google** を選択します。
8. オプション:**説明** フィールドに、リソースの説明を入力します。
9. **Google プロジェクト ID** フィールドに、プロジェクト ID を入力します。
10. **クライアント E-メール** フィールドに、クライアントのメールアドレスを入力します。
11. **証明書パス** フィールドに、サービスアカウントキーへのパスを入力します。(例: `/usr/share/foreman/gce_key.json` など)
12. **ゾーンのロード** をクリックして GCE 環境からゾーン一覧を生成します。
13. **ゾーン** 一覧から使用する GCE ゾーンを選択します。
14. **Submit** をクリックします。

## CLI手順

1. GCE では、JSON 形式でサービスアカウントキーを生成し、このファイルを Satellite Server の `/usr/share/foreman/` ディレクトリーにアップロードします。
2. Satellite Server で、サービスアカウントキーの所有者を **foreman** ユーザーに変更します。

```
# chown foreman /usr/share/foreman/gce_key.json
```

3. サービスアカウントキーのパーミッションを設定して、ファイルを読み取れるようにします。

```
# chmod 0600 /usr/share/foreman/gce_key.json
```

4. サービスアカウントキーの SELinux コンテキストを復元します。

```
# restorecon -vv /usr/share/foreman/gce_key.json
```

5. **hammer compute-resource create** コマンドを使用して GCE コンピュートリソースを Satellite に追加します。

```
# hammer compute-resource create --name 'gce_cr' \
  --provider 'gce' \
  --project 'gce_project_id' \
  --key-path 'gce_key.json' \
  --zone 'us-west1-b' \
  --email 'gce_email'
```

## 15.2. SATELLITE SERVER への GOOGLE COMPUTE ENGINE 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には、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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、Google Chrome Engine 接続の名前をクリックします。
2. **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。**root** ユーザーは SSH キーを使用して GCE インスタンスに接続できないので、**root** 以外のユーザーを指定してください。ユーザー名は文字で始まり、小文字と数字で設定されている必要があります。
7. **イメージ** リストの Google Chrome Engine コンピュートリソースからイメージを選択します。
8. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
9. **送信** をクリックしてイメージの詳細を保存します。

### CLI 手順

- **hammer compute-resource image create** コマンドでイメージを作成します。**root** ユーザーは SSH キーを使用して GCE インスタンスに接続できないので、**--username** オプションを使用して、**root** 以外のユーザーを指定します。ユーザー名は文字で始まり、小文字と数字で設定されている必要があります。

```
# hammer compute-resource image create \  
--name 'gce_image_name' \  
--compute-resource 'gce_cr' \  
--operatingsystem-id 1 \  
--architecture-id 1 \  
--uuid '3780108136525169178' \  
--username 'admin'
```

## 15.3. GOOGLE COMPUTE ENGINE の詳細のコンピューティングプロファイルへの追加

この手順を使用して、GCE ハードウェア設定をコンピューティングプロファイルに追加します。このコンピューティングプロファイルを使用して GCE でホストを作成すると、これらの設定が自動的に入力されます。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. GCE コンピュートリソースの名前をクリックします。
4. **マシンタイプ** 一覧からプロビジョニングに使用するマシンタイプを選択します。
5. **イメージ** リストから、プロビジョニングに使用するイメージを選択します。
6. **ネットワーク** リストから、プロビジョニングに使用する GCE ネットワークを選択します。
7. オプション: **一時的な外部 IP の関連付け** チェックボックスを選択して、動的で一時的な IP アドレスを割り当て、Satellite がホストと通信に使用できるようにします。ホストを再起動すると、このパブリック IP アドレスは変わります。永続的な IP アドレスが必要な場合には、GCE で静的なパブリック IP アドレスを予約し、ホストにアタッチします。
8. **サイズ (GB)** フィールドに、ホストで作成するストレージのサイズを入力します。
9. **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 手順

1. GCE コンピュートリソースで使用するコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name gce_profile
```

2. GCE の情報をコンピュートプロファイルに追加します。

```
# hammer compute-profile values create --compute-profile gce_profile \  
--compute-resource 'gce_cr' \  
--volume "size_gb=20" \  
--compute-attributes "machine_type=f1-micro,associate_external_ip=true,network=default"
```

## 15.4. GOOGLE COMPUTE ENGINE でのイメージベースのホストの作成

Satellite では、Google Chrome Engine プロビジョニングを使用して、既存のイメージからホストを作成できます。新規ホストのエントリは、Google Chrome Engine サーバーが新規ボリュームのベースとして既存のイメージを使用し、インスタンスを作成するようトリガーします。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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、Google Compute Engine 接続を選択します。
6. **コンピュータプロファイル** リストから、仮想マシン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. **ライフサイクル環境** 一覧から、環境を選択します。
8. **インターフェイス** タブをクリックして、ホストのインターフェイスで **編集** をクリックします。
9. 特に以下の項目など、フィールドに自動的に入力されていることを確認します。
  - **ホスト** タブの **名前** は **DNS 名** になります。
  - **MAC アドレス** フィールドは空白です。Google Compute Engine サーバーは MAC アドレスをホストに割り当てます。
  - 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。
  - **ドメイン** フィールドに必要なドメインの情報が入力されていること。
  - ホストの最初のインターフェイスで **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。選択されていない場合は、それらを選択してください。
10. **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。
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、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ます。必要に応じてこれらの設定を変更します。
13. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
14. **送信** をクリックしてホストエントリーを保存します。

## CLI手順

- **--provision-method image** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \  
--name "GCE_VM" \  
--organization "Your_Organization" \  
--location "Your_Location" \  
--compute-resource gce_cr_name
```



```
--compute-profile "gce_profile_name" \  
--provision-method 'image' \  
--image gce_image_name \  
--root-password "your_root_password" \  
--interface "type=interface,domain_id=1,managed=true,primary=true,provision=true" \  
--puppet-environment-id 1 \  
--puppet-ca-proxy-id 1 \  
--puppet-proxy-id 1 \  
--architecture x86_64 \  
--operatingsystem "operating_system_name"
```

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、**hammer host create --help** コマンドを入力します。

## 第16章 MICROSOFT AZURE RESOURCE MANAGER でのクラウドインスタンスのプロビジョニング

Red Hat Satellite は、新規仮想マシンの作成、電源管理の状態の制御など、Microsoft Azure Resource Manager と対話できます。Azure ホストの作成では、イメージベースのプロビジョニングのみがサポートされます。これには、Marketplace イメージ、カスタムイメージ、共有イメージギャラリーを使用したプロビジョニングを含みます。

Azure Resource Manager の概念の詳細は、[Azure Resource Manager documentation](#) を参照してください。

### 前提条件

- Red Hat Enterprise Linux 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。
- 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。
- Azure Active Directory アプリケーション作成に適したパーミッションがあることを確認します。詳細は、[Microsoft identity platform \(Azure Active Directory for developers\)](#) ドキュメントの [Check Azure AD permissions](#) を参照してください。
- Azure Active Directory アプリケーションとサービスのプリンシパルを作成して設定し、アプリケーションまたは **client ID**、ディレクトリーまたは **tenant ID**、およびクライアントシークレットを取得する必要があります。詳細は、[Microsoft identity platform \(Azure Active Directory for developers\)](#) ドキュメントの [Use the portal to create an Azure AD application and service principal that can access resources](#) を参照してください。
- オプション: Azure ホストで Puppet を使用するには、**管理 > 設定 > Puppet** に移動して、**証明書での UUID の使用** 設定を有効にして、Puppet が一貫性のある Puppet 証明書 ID を使用するように設定します。
- ニーズに合わせて **finish** または **user\_data** プロビジョニングテンプレートと使用するオペレーティングシステムを関連付けます。プロビジョニングテンプレートの詳細は、[プロビジョニングテンプレート](#) を参照してください。
- オプション: 仮想マシンで静的なプライベート IP アドレスを使用する場合には、**ネットワークアドレス** フィールドが Azure サブネットのアドレスと一致するように、Satellite のサブネットを作成してください。
- RHEL BYOS イメージを作成する前に、Azure CLI またはポータルでイメージの条件に同意して、イメージを使用してサブスクリプションの仮想マシンを作成し、管理できるようにします。

### 手順の概要

1. [「Microsoft Azure Resource Manager 接続の Satellite Server への追加」](#)
2. [「Microsoft Azure Resource Manager イメージの Satellite Server への追加」](#)
3. [「Microsoft Azure Resource Manager の詳細のコンピューティングプロファイルへの追加」](#)
4. [「Microsoft Azure Resource Manager でのイメージベースのホスト作成」](#)

## 16.1. MICROSOFT AZURE RESOURCE MANAGER 接続の SATELLITE SERVER への追加

以下の手順を使用して Satellite で Microsoft Azure Resource Manager をコンピュートリソースとして追加します。使用する Azure Resource Manager のリージョンごとに、別のコンピュートリソースを追加する必要がある点に留意してください。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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** に移動して、**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、コンピュート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 **Azure リソースマネージャー** を選択します。
4. オプション: **説明** フィールドに、リソースの説明を入力します。
5. デフォルトでは、**Cloud** は Public/Standard に設定されています。Azure Government Cloud は、次のリージョンをサポートしています。
  - 米国政府
  - 中国
  - ドイツ
6. **Client ID** フィールドでアプリケーションか、**クライアント ID** を入力します。
7. **クライアントシークレット** フィールドに、クライアントシークレットを入力します。
8. **サブスクリプション ID** フィールドに、サブスクリプション ID を入力します。
9. **Tenant ID** フィールドで、ディレクトリーまたは **テナント ID** を入力します。
10. **リージョンの読み込み** をクリックします。これにより、Azure Resource Manager への接続が成功し、サブスクリプションで利用可能なリージョンが読み込まれていることをテストします。
11. **Azure リージョン** リストから、使用する Azure リージョンを選択します。
12. **Submit** をクリックします。

### CLI 手順

- **hammer compute-resource create** コマンドを使用して Satellite に Azure コンピュートリソースを追加します。--region オプションの値には、小文字を使用する必要があり、特殊文字を含めることはできません。

```
# hammer compute-resource create --name azure_cr_name \  
--provider azurearm \  
--tenant tenant-id \  
--app-ident client-id \  

```

```
--secret-key client-secret \  
--sub-id subscription-id \  
--region 'region'
```



### 重要

Azure Government Cloud を使用している場合は、**--cloud** パラメーターを渡す必要があります。**cloud** パラメーターの値は次のとおりです。

Azure Government Cloud の名前	hammer --cloud の値
米国政府	azureusgovernment
中国	azurechina
ドイツ	azuregermancloud

## 16.2. MICROSOFT AZURE RESOURCE MANAGER イメージの SATELLITE SERVER への追加

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には、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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、Microsoft Azure Resource Manager 接続の名前をクリックします。
2. **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。**root** ユーザーは使用できません。
7. オプション: **パスワード** フィールドに認証に使用するパスワードを入力します。
8. **Azure Image Name** フィールドでイメージ名を **prefix://UUID** 形式で入力します。
  - カスタムのイメージの場合は **custom** のプリフィックスを使用します。(例: **custom://image-name**)
  - 共有ギャラリーのイメージの場合には **gallery** のプリフィックスを使用します。(例: **gallery://image-name**)

- パブリックまたは RHEL Bring Your Own Subscription (BYOS) イメージの場合には、**marketplace** のプリフィックスを使用します。(例: `marketplace://OpenLogicCentOS:7.5:latest`)  
詳細は、[Find Linux VM images in the Azure Marketplace with the Azure CLI](#) を参照してください。
9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
  10. **送信** をクリックしてイメージの詳細を保存します。

## CLI 手順

- **hammer compute-resource image create** コマンドでイメージを作成します。イメージに入力するユーザー名は、このイメージでホストを作成する時に使用したのと同じでなければなりません。**--password** オプションは、イメージの作成には任意です。**root** ユーザーは使用できません。

```
# hammer compute-resource image create \
--name Azure_image_name \
--compute-resource azure_cr_name \
--uuid 'marketplace://RedHat:RHEL:7-RAW:latest' \
--username 'azure_username' \
--user-data no
```

## 16.3. MICROSOFT AZURE RESOURCE MANAGER の詳細のコンピューティングプロファイルへの追加

この手順を使用して、Azure ハードウェア設定をコンピューティングプロファイルに追加します。このコンピュートプロファイルを使用して Azure でホストを作成すると、これらの設定が自動的に入力されます。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 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** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Azure コンピュートリソースの名前をクリックします。
4. **リソースグループ** リストから、プロビジョニングするリソースグループを選択します。
5. **VM サイズ** リストからプロビジョニングする仮想マシンのサイズを選択します。
6. **プラットフォーム** リストから **Linux** を選択します。
7. **Username** フィールドに、認証に使用するユーザー名を入力します。コンピュートプロファイルに入力するユーザー名は、イメージの作成時に使用したのと同じものでなければなりません。
8. ユーザーを認証するには、以下のオプションのいずれかを使用します。

- パスワードを使用した認証は、**パスワード** フィールドにパスワードを入力します。
  - SSH キーを使用した認証は、**SSH Key** フィールドに SSH キーを入力します。
9. オプション: 仮想マシンに、プレミアム仮想マシンディスクを使用させる場合には、**プレミアム OS ディスク** のチェックボックスを選択します。
  10. **OS ディスクのキャッシュ** リスからディスクのキャッシュ設定を選択します。
  11. オプション: **カスタムスクリプトコマンド** フィールドに、仮想マシンのプロビジョニング時に仮想マシンで実行するコマンドを入力します。
  12. オプション: プロビジョニングの完了時にカスタムスクリプトを実行する場合には、**コンマ区切りのファイルの URI** フィールドに、使用するコンマ区切りのファイルの URI を入力します。Red Hat Satellite はファイルを `/var/lib/waagent/custom-script/download/0/` ディレクトリーにダウンロードし、スクリプトの実行には `sudo` の特権が必要であるため、スクリプトには、最初に **sudo** を含める必要があります。
  13. オプション: **NVIDIA driver / CUDA** チェックボックスを選択することにより、**NVIDIA Driver** を追加できます。詳細は、次の Microsoft Azure のドキュメントを参照してください。
    - [NVIDIA GPU Driver Extension for Linux](#)
    - [NVIDIA GPU Driver Extension for Windows](#)
  14. オプション: VM に追加ボリュームを作成する場合には、**Add Volume** ボタンをクリックして **Size** を GB 単位で入力し、**Data Disk Caching** メソッドを選択します。
    - これらのディスクの最大数は、選択した仮想マシンのサイズによって異なることに注意してください。Microsoft Azure 仮想マシンのストレージ要件の詳細は、[Microsoft Azure ドキュメント](#) を参照してください。
  15. **インターフェイスの追加** をクリックします。



### 重要

インターフェイスの最大数は、選択した仮想マシンサイズにより異なります。詳細は、上記の Microsoft Azure のドキュメントリンクを参照してください。

1. **Azure サブネット** リストからプロビジョニングする Azure サブネットを選択します。
2. **パブリック IP** リストからパブリック IP 設定を選択します。
3. オプション: 静的なプライベート IP を仮想マシンに使用させるには、**静的なプライベート IP** のチェックボックスを選択します。
4. **Submit** をクリックします。

### CLI 手順

1. Azure Resource Manager のコンピュートリソースで使用するコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name compute_profile_name
```

2. コンピュートプロファイルに Azure の情報を追加します。**ユーザー名** の設定では、イメージへのアクセスに使用する SSH ユーザー名を入力します。コンピュートプロファイルに入力するユーザー名は、イメージの作成時に使用したのと同じものでなければなりません。

```
# hammer compute-profile values create \  
--compute-profile "compute_profile_name" \  
--compute-resource azure_cr_name \  
--compute-  
attributes="resource_group=resource_group,vm_size=Standard_B1s,username=azure_us  
er,password=azure_password,platform=Linux,script_command=touch /var/tmp/text.txt" \  
--  
interface="compute_public_ip=Dynamic,compute_network=mysubnetID,compute_private_ip=fal  
se"  
--volume="disk_size_gb=5,data_disk_caching=None"
```

オプション: プロビジョニング後に仮想マシンでスクリプトを実行する場合には、以下の設定を指定します。

- 直接スクリプトを入力するには、**script\_command** 設定で、プロビジョニングした仮想マシンで実行するコマンドを入力します。
- **script\_uris** 設定を使用して URI からスクリプトを実行するには、使用するコンマ区切りのファイルの URI を入力します。Red Hat Satellite はファイルを `/var/lib/qaagent/custom-script/download/0/` ディレクトリーにダウンロードし、スクリプトの実行には `sudo` の特権が必要であるため、スクリプトには、最初に **sudo** を含める必要があります。

## 16.4. MICROSOFT AZURE RESOURCE MANAGER でのイメージベースのホスト作成

Satellite では、Microsoft Azure Resource Manager プロビジョニングを使用して、既存のイメージからホストを作成できます。新規ホストのエントリーは、Microsoft Azure Resource Manager サーバーが新規ボリュームのベースとして既存のイメージを使用し、インスタンスを作成するようトリガーします。

Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

### 手順

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、Microsoft Azure Resource Manager 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. **ライフサイクル環境** 一覧から、環境を選択します。
8. **インターフェイス** タブをクリックして、ホストのインターフェイスで **編集** をクリックします。

9. 特に以下の項目など、フィールドに自動的に入力されていることを確認します。
  - **ホスト タブの 名前** は **DNS 名** になります。
  - **MAC アドレス** フィールドは空白です。Microsoft Azure Resource Manager サーバーは、プロビジョニング中に MAC アドレスをホストに割り当てます。
  - **Azure サブネット** フィールドに必要な Azure サブネットの情報が入力されていること。
  - ホストの最初のインターフェイスで **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。選択されていない場合は、それらを選択してください。
10. オプション: 仮想マシンで **IPv4 Subnet** リストから静的なプライベート IP アドレスを使用する場合には、Azure サブネットのアドレスと一致するように、**ネットワークアドレス** フィールドで、Satellite のサブネットを選択してください。**IPv4 アドレス** フィールドに、Azure サブネットの範囲内の IPv4 アドレスを入力します。
11. **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。
12. **Provisioning Method** には **Image Based** が選択されているようにします。
13. **イメージ** リストから、プロビジョニングに使用する Azure Resource Manager イメージを選択します。
14. **Root パスワード** フィールドで、認証に使用する root パスワードを入力します。
15. **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。
16. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュー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ます。必要に応じてこれらの設定を変更します。
17. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
18. **送信** をクリックしてホストエントリーを保存します。

## CLI手順

- **--provision-method image** を指定して **hammer host create** コマンドでホストを作成します。以下の例の値は、お使いの環境に合った値に置き換えます。

```
# hammer host create \  
--name="Azure_VM" \  
--organization "Your_Organization" \  
--location "Your_Location" \  
--compute-resource azure_cr_name \  
--compute-profile "compute_profile_name" \  
--provision-method 'image' \  
--image Azure_image_name \  
--domain domain_name \  
--architecture x86_64 \  
--operatingsystem "operating_system_name"
```



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、**hammer host create --help** コマンドを入力します。

## 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 (**content-init.sh**) を作成し、以下を組み込みます。

```
#!/bin/bash

MANIFEST=$1

# Import the content from Red Hat CDN
hammer organization create --name "ACME" --label "ACME" \
--description "Our example organization for managing content."
hammer subscription upload --file ~/$MANIFEST --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (RPMs)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (Kickstart)" \
--releasever "7Server" --basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" --organization "ACME"
hammer repository-set enable \
--name "Red Hat Satellite Tools 6.10 (for RHEL 7 Server) (RPMs)" \
--basearch "x86_64" --product "Red Hat Enterprise Linux Server" \
--organization "ACME"
hammer product synchronize --name "Red Hat Enterprise Linux Server" \
--organization "ACME"

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

# Create and publish our Content View
hammer content-view create --name "Base" \
--description "Base operating system" \
--repositories "Red Hat Enterprise Linux 7 Server RPMs x86_64 7Server,Red Hat Satellite Tools 6.10
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"
```

■

スクリプトに実行権限を設定します。

```
# chmod +x 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。

```
# ./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 FIPS 準拠ホストのプロビジョニング

Satellite 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。

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、以下のタスクを実行します。

- オペレーティングシステムのプロビジョニングのパスワードハッシュアルゴリズムを変更する。
- ホストグループを作成し、ホストグループのパラメーターを設定して FIPS を有効にする。

ホストグループの作成に関する情報は、[ホストの管理](#) ガイドの [ホストグループの作成](#) を参照してください。

プロビジョニングしたホストに、FIPS に準拠する設定が適用されている。これらの設定が有効であることを確認するには、「[FIPS モードの有効化の確認](#)」の手順を実行します。

### B.1. 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

FIPS に準拠するホストをプロビジョニングするには、プロビジョニングで使用す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オペレーティングシステムごとに、適用する必要があります。

1. オペレーティングシステム ID を特定します。

```
$ hammer os list
```

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 Operating_System \  
--password-hash SHA256
```

3. **TITLE** 列と同じ値を使用して、オペレーティングシステムごとにこのコマンドを繰り返します。

```
$ hammer os update --title "RedHat version_number" \  
--password-hash SHA256
```

値をコンマ区切りで指定することはできない点に注意してください。

### B.2. FIPS 有効化パラメーターの設定

FIPS 準拠のホストをプロビジョニングするには、ホストグループを作成して、ホストグループのパラメーター **fips\_enabled** を **true** に設定する必要があります。このパラメーターが **true** に設定されていない場合や、ない場合には、FIPS 固有の変更がシステムに適用されません。ホストのプロビジョニング時または、ホストグループを使用する場合に、このパラメーターを使用してください。

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。

```
$ hammer hostgroup set-parameter --name fips_enabled \  
--value 'true' \  
--hostgroup prod_servers
```

詳細は、**hammer hostgroup set-parameter --help** コマンドの出力を参照してください。

### B.3. 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. **root** または管理者レベルのアカウントで、ホストにログインしてください。
2. 以下のコマンドを入力します。

```
$ cat /proc/sys/crypto/fips_enabled
```

値が **1** の場合は、FIPS モードが有効であることが分かります。

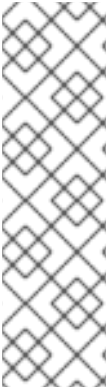
## 付録C RED HAT SATELLITE 向けクラウドイメージの構築

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。

- [最新の RHEL 7 KVM ゲストイメージ](#)
- [最新の RHEL 6 KVM ゲストイメージ](#)

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。

## C.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、イメージを作成する。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、この手順を実行する。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。

## C.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。以下の例を目安にしてください。

```
[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イスを **ON** に設定し、システムの開始時にインターフェイ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イスが起動し、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。

### C.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されます。





## 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。

## C.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る。
  - 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る (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する。
  - 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する。
  - 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
```

## C.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する。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** ユーティリティのバージョンが 1.10 以上であることを確認する。パッケージは標準の Red Hat Enterprise Linux リポジトリで入手可能です。

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。

## C.6. KATELLO エージェントのインストール

### 注記

Katello エージェントは非推奨で、今後の Satellite のバージョンで削除されます。ワークロードを移行して、リモート実行機能を使用してクライアントをリモートで更新します。詳細は、**ホストの管理** ガイドの [Katello エージェントのインストール](#) を参照してください。

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 **goferd** パッケージによって異なります。

### 前提条件

Satellite Tools 6.10 リポジトリは、必要なパッケージを提供するため、有効にして Satellite Server と同期し、ホストが利用できるようにする必要があります。Satellite Tools 6.10 の有効化に関する情報は、**ホストの管理** ガイドの [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
```

## C.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
```

## C.8. RED HAT ENTERPRISE LINUX 7 イメージの完了

1. システムを更新します。

```
# {package-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
```

- 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
```

- 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
```

- インスタンスの電源をオフにします。

```
# poweroff
```

- 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/
```

- 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
```

- 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。

## C.9. RED HAT ENTERPRISE LINUX 6 イメージの完了

- システムを更新します。

```
# {package-update}
```

- cloud-init** パッケージをインストールします。

```
# yum install cloud-utils-growpart cloud-init
```

- `/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イスが `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。

## C.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。
- 後で使用できるように、保存場所にイメージを移動します。

