



Red Hat Satellite 6.2

プロビジョニングガイド

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

エディション 1.0

Red Hat Satellite 6.2 プロビジョニングガイド

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

法律上の通知

Copyright © 2018 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, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

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

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

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

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

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

目次

第1章 はじめに	5
1.1. アプリケーションライフサイクルでのシステムのプロビジョニング	5
1.2. プロビジョニングタイプの定義	5
1.3. シナリオの定義	6
1.4. 章の概要	6
第2章 プロビジョニングコンテキストの設定	8
2.1. プロビジョニングコンテキストの定義	8
2.2. 組織の作成	9
2.3. ロケーションの作成	10
2.4. コンテキストの設定	12
2.5. 章の概要	12
第3章 プロビジョニングリソースの設定	13
3.1. インストールメディアの作成	13
3.2. パーティションテーブルの作成	14
3.3. プロビジョニングテンプレートの作成	15
3.4. オペレーティングシステムの作成	17
3.5. コンピュートプロファイルの作成	18
3.6. アクティベーションキーの作成	18
3.7. 章の概要	20
第4章 ネットワークの設定	21
4.1. ネットワークサービスの設定	21
4.2. ドメインの SATELLITE SERVER への追加	24
4.3. サブネットの SATELLITE SERVER への追加	25
4.4. 章の概要	26
第5章 プロビジョニングワークフローの概要	28
5.1. プロビジョニングワークフローの定義	28
5.2. SATELLITE SERVER でのホストの作成	29
5.3. SATELLITE SERVER でのホストグループの作成	32
5.4. 章の概要	34
第6章 ベアメタルホストのプロビジョニング	36
6.1. ベアメタルプロビジョニングの要件定義	36
6.2. 無人プロビジョニングによる新規ホストの作成	36
6.3. RED HAT SATELLITE 検出サービスの設定	38
6.4. 検出されたホストからの新規ホストの作成	41
6.5. 検出ルールの作成	42
6.6. PXE を使用しないプロビジョニングによる新規ホストの作成	44
6.7. PXE を使用しない検出の実装	46
6.8. 章の概要	49
第7章 KVM サーバー (LIBVIRT) での仮想マシンのプロビジョニング	50
7.1. KVM プロビジョニングの要件定義	50
7.2. SATELLITE SERVER での KVM 接続の設定	50
7.3. KVM 接続の SATELLITE SERVER への追加	51
7.4. SATELLITE SERVER での KVM イメージの追加	51
7.5. KVM の詳細のコンピュートプロファイルへの追加	52
7.6. KVM サーバーでのネットワークベースのホストの作成	53
7.7. KVM サーバーでのイメージベースのホストの作成	55
7.8. 章の概要	56

第8章 RED HAT ENTERPRISE VIRTUALIZATION での仮想マシンのプロビジョニング	58
8.1. RED HAT ENTERPRISE VIRTUALIZATION のプロビジョニングの要件定義	58
8.2. RED HAT ENTERPRISE VIRTUALIZATION ユーザーの作成	58
8.3. RED HAT ENTERPRISE VIRTUALIZATION 接続の SATELLITE SERVER への追加	59
8.4. SATELLITE SERVER での RED HAT ENTERPRISE VIRTUALIZATION イメージの追加	60
8.5. RED HAT ENTERPRISE VIRTUALIZATION の詳細のコンピュータプロファイルへの追加	61
8.6. RED HAT ENTERPRISE VIRTUALIZATION サーバーでのネットワークベースのホストの作成	62
8.7. RED HAT ENTERPRISE VIRTUALIZATION サーバーでのイメージベースのホストの作成	64
8.8. 章の概要	66
第9章 VMWARE VSPHERE での仮想マシンのプロビジョニング	67
9.1. VMWARE VSPHERE プロビジョニングの要件定義	67
9.2. VMWARE VSPHERE ユーザーの作成	67
9.3. VMWARE VSPHERE 接続の SATELLITE SERVER への追加	67
9.4. SATELLITE SERVER での VMWARE VSPHERE イメージの追加	68
9.5. VMWARE VSPHERE の詳細のコンピュータプロファイルへの追加	69
9.6. VMWARE VSPHERE サーバーでのネットワークベースのホストの作成	70
9.7. VMWARE VSPHERE サーバーでのイメージベースのホストの作成	72
9.8. 章の概要	74
第10章 RED HAT OPENSTACK PLATFORM でのクラウドインスタンスのプロビジョニング	75
10.1. RED HAT OPENSTACK PLATFORM のプロビジョニングの要件定義	75
10.2. RED HAT OPENSTACK PLATFORM 接続の SATELLITE SERVER への追加	75
10.3. SATELLITE SERVER での RED HAT OPENSTACK PLATFORM イメージの追加	76
10.4. RED HAT OPENSTACK PLATFORM の詳細のコンピュータプロファイルへの追加	77
10.5. RED HAT OPENSTACK PLATFORM でのイメージベースのホストの作成	77
10.6. 章の概要	79
第11章 AMAZON EC2 でのクラウドインスタンスのプロビジョニング	80
11.1. AMAZON EC2 プロビジョニングの要件定義	80
11.2. AMAZON EC2 接続の SATELLITE SERVER への追加	80
11.3. SATELLITE SERVER での AMAZON EC2 イメージの追加	81
11.4. AMAZON EC2 の詳細のコンピュータプロファイルへの追加	82
11.5. AMAZON EC2 でのイメージベースのホストの作成	82
11.6. 章の概要	84
第12章 コンテナのプロビジョニング	85
12.1. コンテナプロビジョニングの要件定義	85
12.2. RED HAT ENTERPRISE LINUX ATOMIC HOST の設定	85
12.3. ATOMIC HOST 接続の SATELLITE SERVER への追加	86
12.4. 外部レジストリーの SATELLITE SERVER への追加	86
12.5. SATELLITE SERVER でのコンテナの作成	87
12.6. 章の概要	89
第13章 プロビジョニングの最終設定	90
13.1. シナリオ目標の完了	90
13.2. 他のアプリケーションの統合	90
13.3. 他のドキュメントの参照	90
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	92
付録B HAMMER CLI の追加のホストパラメーター	94
B.1. 共通のインターフェースパラメーター	94
B.2. EC2 パラメーター	95
B.3. LIBVIRT パラメーター	96

B.4. RED HAT OPENSTACK PLATFORM パラメーター	97
B.5. RED HAT ENTERPRISE VIRTUALIZATION パラメーター	97
B.6. VMWARE INTERFACE パラメーター	98
付録C FIPS 準拠ホストのプロビジョニング	100
C.1. 関連するオペレーティングシステム、ロケーションおよび組織の特定	100
C.2. FIPS プロビジョニングテンプレートの作成および有効化	101
C.3. プロビジョニングのパスワードハッシュアルゴリズムの変更	104
C.4. PUPPET の FIPS 準拠メッセージアルゴリズムへの切り替え	105
C.5. FIPS 有効化パラメーターの設定	105
C.6. FIPS モードの有効化の確認	105

第1章 はじめに

Red Hat Satellite 6 におけるプロビジョニングとは、コンピューティングリソースのプールから、新たに設定されるホストを要求する機能です。システム管理者は、異なる種類のコンピューティングリソースを **Satellite Server** に割り当て、複数の異なる方法を使用してそれらのプールと通信します。たとえば、**Satellite Server** は PXE ベースと PXE を使用しない方法の両方を使用してベアメタルシステムをプロビジョニングできます。同様に **Satellite Server** は、API 経由で特定のプロバイダーからクラウドインスタンスをプロビジョニングできます。これらのプロビジョニング方法は Red Hat Satellite 6 アプリケーションライフサイクルの一部として組み込まれています。そのため、これは新規システムの作成だけでなく、それらを管理し、Red Hat コンテンツで最新の状態を維持する必要のあるユーザーにとって便利な機能です。

1.1. アプリケーションライフサイクルでのシステムのプロビジョニング

アプリケーションライフサイクルは、特定のシステムとそのソフトウェアを特定の段階でプロビジョニングする方法を定義します。単純なケースとして、アプリケーションライフサイクルには以下のような 2 つのステージが含まれる場合があります。

- Development (開発)
- Production (本番)

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

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

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

1.2. プロビジョニングタイプの定義

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

ベアメタルプロビジョニング

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

クラウドプロバイダー

Satellite Server はプライベート (Red Hat OpenStack Platform) およびパブリック (Amazon EC2) クラウドプロバイダーに接続します。これにより、クラウド環境で保存されたイメージから新規イン

スタンスをプロビジョニングする方法を実行できます。この方法には、使用するハードウェアのプロファイル(またはフレーバー)を定義する機能も含まれます。

仮想化インフラストラクチャー

Satellite Server は、Red Hat Enterprise 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か、またはベアメタルプロバイダーと同じ PXE ベースのブート方法を使用して仮想マシンをプロビジョニングする方法を実行できます。

Linux コンテナ

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

1.3. シナリオの定義

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

ACME の Satellite Server は Red Hat のコンテンツ配信ネットワークのコンテンツおよびその他のソースと同期しています。これと同様に、お使いになる Satellite Server にもプロビジョニングの実行前に同期されたコンテンツが含まれている必要があります。そのため、まず [Red Hat Satellite 6 コンテンツ管理ガイド](#) のシナリオに実行してから本書を参照することをお勧めします。



注記

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

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

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

EOF
```



重要

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

1.4. 章の概要

本章では、Red Hat Satellite 6 のコンテキストでプロビジョニングの概念を説明し、Red Hat Satellite 6 アプリケーションライフサイクルにおけるプロビジョニングの位置付けについても説明しました。ま

た、**Red Hat Satellite 6**で使用される各種のプロビジョニングタイプについても簡単に説明しました。本書の後の章では、基づくプロビジョニングの複数のシナリオについて説明します。

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

第2章 プロビジョニングコンテキストの設定

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

2.1. プロビジョニングコンテキストの定義

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

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

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

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

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

このシナリオでは、単純なプロビジョニングコンテキストを使用しています。[Red Hat Satellite 6 コンテンツ管理ガイド](#) には、本書のシナリオで使用する **ACME** の組織を作成する方法が説明されています。またロケーションの作成方法についても説明されています。

2.2. 組織の作成

以下の手順は、組織の作成方法を示しています。この手順については、[Red Hat Satellite 6 コンテンツ管理ガイド](#)にも記載されており、同ガイドの組織の作成シナリオに従う場合は以下の手順に従う必要はありません。組織に変更を加える必要があり、その際には組織のプロパティを編集する必要があることに注意してください。

Web UI を使用する場合

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

新規組織 をクリックします。

以下の 3 つのセクションから構成される作成ウィザードが表示されます。

組織の作成

組織の基本的な詳細情報を指定します。これには以下が含まれます。

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

ホストの選択

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

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

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

プロパティの編集

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

重要

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

- **Capsule (スマートプロキシ):** Satellite Server と同じホスト名を使用する Satellite Server の統合 Capsule。
- **メディア:** [Red Hat Satellite 6 コンテンツ管理ガイド](#) のコンテンツと同期した Red Hat Enterprise Linux 7.2 キックスタートツリー。
- **プロビジョニングテンプレート:** 選択されたすべてのテンプレート。
- **パーティションテーブル:** 選択されたすべてのテーブル。
- **ドメイン:** この組織が使用し、管理するドメイン。
- **環境: production (本番)** および [Red Hat Satellite 6 コンテンツ管理ガイド](#) で作成された環境を含むすべての Puppet 環境。

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

組織の作成後に、送信 をクリックします。

CLI を使用する場合

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

これにより、組織のサンプルが作成されます。

2.3. ロケーションの作成

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

Web UI を使用する場合

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

新規ロケーション をクリックします。

以下の 3 つのセクションから構成される作成ウィザードが表示されます。

組織の作成

ロケーションの基本的な詳細情報を指定します。これには、以下が含まれます。

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

ホストの選択

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

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

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

プロパティの編集

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

重要

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

- **Capsule (スマートプロキシ):** Satellite Server と同じホスト名を使用する Satellite Server の統合 Capsule。
- **メディア:** Red Hat Satellite 6 コンテンツ管理ガイド のコンテンツと同期した Red Hat Enterprise Linux 7.2 キックスタートツリー。
- **プロビジョニングテンプレート:** 選択されたすべてのテンプレート。
- **パーティションテーブル:** 選択されたすべてのテーブル。
- **ドメイン:** このロケーションが使用し、管理するドメイン。
- **環境:** production (本番) および Red Hat Satellite 6 コンテンツ管理ガイド で作成された環境を含むすべての Puppet 環境。
- **組織:** ACME の組織。

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

ロケーションの作成後に、送信 をクリックします。

CLI を使用する場合

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

これにより、ロケーションのサンプルが作成されます。

2.4. コンテキストの設定

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

Web UI を使用する場合

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



注記

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

CLI を使用する場合

CLI を使用している場合は、コマンドの末尾にオプションとして **--organization** または **--organization-id** および **--location** または **--location-id** のいずれかを含めます。以下は例になります。

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

これにより、CLI を使って対話のコンテキストが設定されます。

2.5. 章の概要

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

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

第3章 プロビジョニングリソースの設定

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

サポートされるアーキテクチャー

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

3.1. インストールメディアの作成

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

Red Hat の CDN からキックスタートツリーをインポートすると、インストールメディアページに新規エントリーが作成されます。このプロセスは、『Red Hat Satellite 6 コンテンツ管理ガイド』の「[同期する Red Hat リポジトリの選択](#)」セクションで説明されています。リポジトリからの同期済みインストールメディアの場合、これを手動で定義する必要はありません。インストールメディアは、ホストまたはホストグループの作成時に **オペレーティングシステム** タブの **同期したコンテンツ** で利用できます。

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

Web UI を使用する場合

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

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

HTTP パスの例:

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

NFS パスの例:

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



注記

Capsule Server の同期したコンテンツは HTTP パスを常に使用します。Capsule Server で管理されたコンテンツは NFS パスをサポートしません。

- **オペレーティングシステムの種類** メディアのディストリビューションまたはファミリー。たとえば、Red Hat Enterprise Linux、CentOS、および Fedora は **Red Hat** ファミリーに属します。

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

送信 をクリックしてインストールメディアを保存します。

CLI を使用する場合

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

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

3.2. パーティションテーブルの作成

パーティションテーブルは、Satellite Server が新規ホストで利用可能なディスクを設定する方法を定義する一連のディレクティブです。Red Hat Satellite 6 には、**Kickstart default** などの、使用する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すると、必要なパーティション設定スキームを設定したり、新規パーティションテーブルのエントリを作成したり、そのエントリを Red Hat Enterprise Linux オペレーティングシステムのエントリに指定したりできます。

Web UI を使用する場合

ホスト > **パーティションテーブル** に移動し、**新規パーティションテーブル** をクリックします。UI にはパーティションテーブルの詳細を入力できる一連のフィールドがあります。

- **名前:** パーティションテーブルを表す名前。
- **デフォルト:** テンプレートが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かどうかを定義します。
- **スニペット:** テンプレートが他のパーティションテーブルレイアウトの再利用可能なスニペットかどうかを定義します。
- **オペレーティングシステムの種類** パーティションレイアウトのディストリビューションまたはファミリー。たとえば、Red Hat Enterprise Linux、CentOS、および Fedora は **Red Hat** ファミリーに属します。
- **テンプレートエディター:** ディスクパーティションのレイアウトを入力するテキスト領域。以下は例になります。

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

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



注記

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

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

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

送信 をクリックしてパーティションテーブルを保存します。

CLI を使用する場合

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

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

3.3. プロビジョニングテンプレートの作成

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

- **provision:** プロビジョニングプロセスを定義する主なテンプレート (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**、**iPXE**、**PXEGrub:** PXE ベースのネットワークの起動テンプレート。
- **finish:** 主要なプロビジョニングプロセスの完了後の設定後スクリプト。これは SSH タスクとして実行されます。
- **Bootdisk:** PXE を使用しないブート方法のためのテンプレート。
- **kexec:** PXE を使用しないブート方法のためのカーネル実行テンプレート。
- **user_data: cloud-init** スクリプトなどのユーザーデータを受け入れるプロバイダー用の設定後スクリプト。
- **script:** デフォルトで使用されないが、カスタムタスクに役立つ任意のスクリプト。
- **ZTP:** Zero Touch Provisioning テンプレート。
- **POAP:** PowerOn Auto Provisioning テンプレート。

Red Hat Satellite には数多くのテンプレートのサンプルが含まれます。 **ホスト > プロビジョニングテン**

プレートに移動するとそれらを表示できます。それらのいずれかのクローンを作成したり、調整したり、独自のテンプレートを作成したりできます。テンプレートには **Embedded Ruby (ERB)** 構文を使用できます。詳細は、『[ホスト設定ガイド](#)』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できます。ダウンロードを可能にするには、まずデバッグ証明書を作成する必要があります。『[サーバー管理ガイド](#)』の「[組織のデバッグ証明書の作成](#)」を参照してください。

Web UI を使用する場合

ホスト > プロビジョニングテンプレートに移動し、**新規テンプレート** をクリックします。UI にはプロビジョニングテンプレートの詳細を入力できる一連のフィールドがあります。または、サンプルのテンプレートのいずれかを選択し、**クローン** をクリックしてこれを複製してから事前に設定された詳細を変更することができます。

- **テンプレート タブ:**
 - **名前:** プロビジョニングテンプレートのテキスト形式の名前。
 - **デフォルト:** テンプレートが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かどうかを定義します。
 - **テンプレートエディター:** プロビジョニングテンプレートの本文を入力するテキスト領域。テンプレートファイルブラウザを使用してテンプレートファイルをアップロードすることもできます。
 - **監査コメント:** プロビジョニングテンプレートへの変更の概要フィールドです。
- **タイプ タブ:**
 - **スニペット:** プロビジョニングテンプレートをスニペットとして指定します。スニペットはスタンドアロンのプロビジョニングテンプレートではありませんが、他のプロビジョニングテンプレートに挿入できるプロビジョニングテンプレートの一部を構成します。
 - **タイプ:** **provision** などのテンプレートのタイプを定義します。
- **関連付け タブ:**
 - **適用可能なオペレーティングシステムセクションのすべての項目の一覧から、オペレーティングシステムエントリーの名前をクリックしてそのオペレーティングシステムエントリーを選択された項目の一覧に移動し、プロビジョニングテンプレートとそのオペレーティングシステムエントリーで使用できるようにします。**
 - またオプションとして、**組み合わせの追加** をクリックして **ホストグループ** の一覧からホストグループを1つ選択するか、または **環境** の一覧から環境を1つ選択すると、指定したホストグループと環境の組み合わせにプロビジョニングテンプレートを使用できるようになります。

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

送信 をクリックしてプロビジョニングテンプレートを保存します。

CLI を使用する場合

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

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

3.4. オペレーティングシステムの作成

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

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

Web UI を使用する場合

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

- **オペレーティングシステムタブ:**
 - **名前:** オペレーティングシステムエントリーを表すテキスト形式の名前。
 - **メジャーバージョン:** オペレーティングシステムのメジャーバージョンに対応する番号。
 - **マイナーバージョン:** オペレーティングシステムのマイナーバージョンに対応する番号。
 - **説明:** オペレーティングシステムの説明を入力するテキストフィールド。
 - **ファミリー:** 新規オペレーティングシステムの分類に使用するオペレーティングシステムのファミリー。
 - **root パスワードのハッシュ:** root パスワードのエンコーディング方法を定義します。
 - **アーキテクチャー:** オペレーティングシステムが使用するアーキテクチャーを選択します。ホスト > アーキテクチャー メニューで追加のアーキテクチャーを作成します。
- **パーティションテーブルタブ:**
 - このオペレーティングシステムに適用できる可能性のあるパーティションテーブルを選択します。
- **インストールメディアタブ:**
 - このオペレーティングシステムに適用されるインストールメディアを選択します。キックスタートツリーはリポジトリの同期時に自動的に利用可能になるはずですが、[インストールメディアの作成](#) を参照してください。
- **テンプレートタブ:**
 - オペレーティングシステムに適用される各タイプのテンプレートを選択します。

送信 をクリックしてプロビジョニングテンプレートを保存します。

CLI を使用する場合

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

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

以下の点に注意してください。

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

3.5. コンピュートプロファイルの作成

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

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

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

Web UI を使用する場合

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

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

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。

3.6. アクティベーションキーの作成

新規ホストを作成する前に、アクティベーションキーを作成することをお勧めします。このアクティベーションキーは、プロビジョニングのシナリオでシステムの登録時に使用されます。本書のシナリオでは、Red Hat Satellite 6 コンテンツ管理ガイド のサブスクリプションおよびリポジトリを割り当てるためにアクティベーションキーのサンプルを作成します。

Web UI を使用する場合

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

- **名前:** アクティベーションキーの名前。システムの登録プロセスでこの名前を使用します。example と入力します。
- **コンテンツホストの制限** このアクティベーションキーで Satellite Server が登録を許可するシステム数を定義します。無制限のコンテンツホスト を選択します。

- **説明:** アクティベーションキーのテキスト形式の説明。 **Example activation key** と入力します。
- **環境:** 使用する環境。 **Production** を選択します。
- **コンテンツビュー:** 環境内で使用するコンテンツビュー (およびリポジトリ)。 **Base** を選択します。

保存 をクリックすると、アクティベーションキーの詳細画面が表示されます。

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



注記

Auto-Attach オプションは有効にされています。これにより、登録時にこれらの製品の割り当てが自動的に実行され、必要なリポジトリが有効になります。アクティベーションキーの **Auto-Attach** が無効になっている場合は、システムはサブスクリプションやコンテンツを割り当てず、**Satellite Server** に登録されるのみとなります。

製品コンテンツページに移動します。アクティベーションキーの製品に関連付けられている全リポジトリが表示されます。デフォルトでは、**Satellite Server** は以下のみを有効にしています。

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

シナリオには以下のデフォルトが設定されているはずです。

Red Hat Enterprise Linux Server:

- Red Hat Enterprise Linux 7 Server (Kickstart) - Enabled: **No (Default)**
- Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs) - Enabled: **No (Default)**
- Red Hat Enterprise Linux 7 Server (RPMs) - Enabled: **Yes (Default)**

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

- Red Hat Satellite Tools 6.2 (for RHEL 7 Server) (RPMs) - Enabled: **Override to Yes**

保存 をクリックします。

CLI を使用する場合

アクティベーションキーを作成します。

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

サブスクリプション ID 一覧を取得します。



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

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

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

アクティベーションキーに関連付けられている製品コンテンツを一覧表示します。

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

Red Hat Satellite Tools 6.2 リポジトリのデフォルトの自動有効化ステータスを上書きします。デフォルトでは無効にされています。以下のコマンドでこれを有効にします。

```
# hammer activation-key content-override --name "example" --content-label rhel-7-server-satellite-tools-6.2-rpms --value 1 --organization "ACME"
```

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

3.7. 章の概要

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

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

第4章 ネットワークの設定

それぞれのプロビジョニングタイプにはネットワーク設定が必要です。新規ホストが **Satellite Server** の統合 **Capsule** または外部 **Capsule Server** のいずれかにアクセスできるようにする必要があります。**Satellite Server** または **Capsule Server** の設定を行うには、以下の2つの基本的な要件を満たしている必要があります。

- 統合 **Capsule** または **Capsule Server** のネットワークサービスの設定。これには以下が含まれます。
 - コンテンツ配信サービス
 - ネットワークサービス (DHCP、DNS、および TFTP)
 - Puppet 設定
- **Satellite Server** でのネットワークリソースデータの定義。これは新規ホストでネットワークインターフェースを設定するのに役立ちます。

本章では、**Satellite Server** の統合 **Capsule** でネットワークサービスを設定する方法を重点的に説明します。これらの方法は、特定のネットワークを管理するスタンドアロン **Capsule Server** の設定にも同様に当てはまります。

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

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

Satellite Server では、検出/管理対象外サービスとプロビジョニングサービスの両方に同じ DHCP 範囲を定義することはできますが、それぞれのサービスに同じサブネット内の別々の範囲を使用することが推奨されます。

4.1. ネットワークサービスの設定

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

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

```
# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 52:54:00:33:e3:1c brd ff:ff:ff:ff:ff:ff
    inet 192.168.125.35/24 brd 192.168.125.255 scope global dynamic ens3
        valid_lft 3042sec preferred_lft 3042sec
    inet6 fe80::5054:ff:fe33:e31c/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
    link/ether 52:54:00:fd:24:ae brd ff:ff:ff:ff:ff:ff
    inet 192.168.140.2/24 brd 192.168.140.255 scope global ens8
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fed:24ae/64 scope link
        valid_lft forever preferred_lft forever
```

Satellite Server は、Red Hat の CDN などの外部接続に **eth0** を使用します。ACME は **eth1** インターフェースから、**192.168.140.0/24** サブネットを使用したホストのプライベートプロビジョニングネットワークに接続しようとします。ここでは、Satellite Server の統合 Capsule がこのネットワーク上で新規ホストの DHCP、DNS、および TFTP サーバーとして機能させることが目標になります。



注記

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

この例では、**satellite-installer** スクリプトはそれらのサービスを設定するために以下のオプションを使用します。

DHCP オプション

--foreman-proxy-dhcp

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

--foreman-proxy-dhcp-gateway

DHCP プールのゲートウェイを定義します。この例では、これを ACME のプライベートネットワークの外部ゲートウェイのアドレスである **192.168.140.1** に設定します。

--foreman-proxy-dhcp-interface

要求をリッスンするために DHCP サービスのインターフェースを設定します。この例では、**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。この例では、**eth1** の Satellite Server のアドレス **192.168.140.1** に設定します。

--foreman-proxy-dhcp-range

検出および管理対象外サービスのスペースで区切られた DHCP プール範囲を定義します。この例では、**192.168.140.10 192.168.140.110** に設定します。これにより、100 アドレスを含むプールが提供されます。

--foreman-proxy-dhcp-server

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

DNS オプション**--foreman-proxy-dns**

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

--foreman-proxy-dns-forwarders

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

--foreman-proxy-dns-interface

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

--foreman-proxy-dns-reverse

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

--foreman-proxy-dns-server

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

--foreman-proxy-dns-zone

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

TFTP オプション**--foreman-proxy-tftp**

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

**注記**

satellite-installer --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。

以下は、設定コマンドの例になります。

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

satellite-installer スクリプトはこれらの設定オプションを適用し、必要なネットワークサービスをセットアップします。設定が完了したら、**hammer proxy info** コマンドを使用して選択した **Capsule Server** でこれらのサービスを確認します。この例では、**Satellite Server** の統合 **Capsule** のドメイン名として **satellite.example.com** を使用しています。

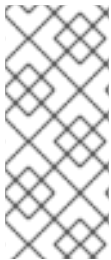
```
# hammer proxy info --name "satellite.example.com"
```

出力には、DNS、DHCP、および TFTP などの有効にされた機能の一覧が表示されます。

```
Features:
  Pulp
  TFTP
  DNS
  DHCP
  Puppet
  Puppet CA
  Dynflow
  SSH
```

4.2. ドメインの **SATELLITE SERVER** への追加

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



注記

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

Web UI を使用する場合

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

- **ドメイン タブ:**
 - **Name (名前):** ドメイン名。この例では **example.com** を使用しています。
 - **Description (フルネーム):** ドメインのテキスト形式の説明。この例では **ACME's example domain** を使用しています。
 - **DNS Capsule (DNS プロキシ):** DNS の割り当てに使用する **Capsule**。この例では、**Satellite Server** の統合 **Capsule** を使用しています。
- **ロケーション タブ:**
 - このドメインを使用する場所を選択します。たとえば、**New York** などのロケーションを選択します。

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

CLI を使用する場合

以下のコマンドでドメインを作成します。

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



注記

この例では `--dns-id` オプションに `1` を指定しています。1 は **Satellite Server** の統合 **Capsule** の ID です。

4.3. サブネットの **SATELLITE SERVER** への追加

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

Web UI を使用する場合

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

- **サブネット** タブ:
 - **名前**: サブネットのテキスト形式の名前。例: **ACME's Internal Network**
 - **ネットワークアドレス**: サブネットのネットワークアドレス。例: **192.168.140.0**
 - **ネットワークマスク**: サブネットのネットワークマスク。例: **255.255.255.0**
 - **ゲートウェイアドレス**: サブネットの外部ゲートウェイ。例: **192.168.140.1**
 - **プライマリー DNS サーバー**: サブネットのプライマリー DNS。例: **192.168.140.2**
 - **セカンダリー DNS サーバー**: サブネットのセカンダリー DNS。例: **8.8.8.8**
 - **IPAM**: IP アドレス管理 (IPAM) に使用するメソッド:
 - **DHCP**: サブネットには DHCP サーバーが含まれます。
 - **内部 DB**: サブネットには DHCP サーバーは含まれませんが、**Satellite** で IP アドレスの割り当てを管理し、内部データベースに IP アドレスを記録できるようにすることが想定されます。
 - **なし**: IP アドレス管理がありません。
この例では、**Satellite Server** は DHCP サーバーとして機能するために **DHCP** を使用します。

- **開始アドレス (Start of IP range):** プロビジョニングサービスの IP 割り当て範囲の開始を定義します。例: **192.168.140.111**
 - **終了アドレス (End of IP range):** プロビジョニングサービスの IP 割り当ての終了を定義します。例: **192.168.140.250**
 - **VLAN ID:** ブロードキャストを分離するためにサブネットの VLAN ID 番号を定義します。この例では VLAN を使用しないため、このフィールドを空白にします。
 - **ブートモード:** このネットワークのネットワークインターフェースのデフォルトブートモードを定義します。静的に割り当てられた IP アドレスに **静的** を使用し、IP アドレスを動的に割り当てるには **DHCP** を使用します。
- **リモート実行 タブ:**
 - リモート実行を制御する **capsule** を選択します。この例では **Satellite Server** 自体がこれに該当します。
 - **ドメイン タブ:**
 - このサブネットに適用されるドメインを選択します。
 - **プロキシ (Capsules) タブ:**
 - DHCP、TFTP、およびリバース DNS サービスを含む、サブネットの各サービスに適用される **capsule** を選択します。この例では、それぞれのサービスに **Satellite Server** の統合 **Capsule** を使用します。
 - **ロケーション タブ:**
 - この **capsule** を使用するロケーションを選択します。たとえば、**New York** というロケーションを選択します。
 - **組織 タブ:**
 - この **capsule** を使用する組織を選択します。たとえば、**ACME** を選択します。

送信 をクリックしてサブネットの情報を保存します。

CLI を使用する場合

以下のコマンドでサブネットを作成します。

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



注記

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

4.4. 章の概要

本章では、**Satellite Server** の統合 **Capsule** で特定のネットワークサービスを設定する方法や、**Satellite Server** が制御するドメインおよびサブネットの詳細をマップする方法を説明しました。これで、新規ホストのネットワークが提供され、また **PXE** ブートやネットワーク設定などの主なサービスがホストに提供されます。

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

第5章 プロビジョニングワークフローの概要

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

5.1. プロビジョニングワークフローの定義

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

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

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

- **検出:** 検出サービスを使用している場合、**Satellite Server** は新規ホストの **MAC** アドレスを自動的に検出し、要求の送信後にホストを再起動します。**Satellite** でホストを再起動できるように、ホストが割り当てられる **Capusule** から **TCP** ポート **8443** にアクセスできなければならないことに注意してください。
- **PXE なしのプロビジョニング:** 新規ホストの要求の送信後に、**Satellite Server** からダウンロードしたブートディスクで特定のホストを起動する必要があります。
- **コンピュートリソース:** コンピュートリソースは新規ホストの仮想マシンを作成し、**MAC** アドレスを **Satellite Server** に返します。また、イメージベースのプロビジョニングを使用している場合、ホストは標準の **PXE** ブートおよびオペレーティングシステムのインストール方法に従いません。その代わりに、コンピュートリソースは新規ホストが使用する選択したイメージのコピーを作成します。
- **コンテナ:** コンテナのプロビジョニングプロセスはこのワークフロープロセスに従いません。

5.2. SATELLITE SERVER でのホストの作成

ホストのプロビジョニングを設定するには、**Satellite Server** でホストのエントリーの作成することからスタートします。これは **Web UI** または **Hammer CLI** で実行できます。ここでは、ホストのプロビジョニングに必要な基本的な要素を説明します。これらは、後の章で扱われる特定のプロビジョニング方法に関連した参照情報になります。

Web UI を使用する場合

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

ホスト タブで、ホストとその配置についての主な詳細を定義します。

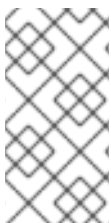
- **名前:** ホストの名前。
- **組織:** このホストを所有する組織。
- **ロケーション:** このホストのロケーション。
- **ホストグループ:** このホストのテンプレートとして使用するホストグループを定義します。
- **デプロイ先:** ベアメタルのホストまたはコンピュートリソースかなど、ホストのデプロイメントのタイプを定義します。
- **ライフサイクル環境:** アプリケーションライフサイクルにおけるホストのステージを定義します。
- **コンテンツビュー:** リポジトリに使用するコンテンツビューを定義します。
- **Puppet 環境:** ホストを含む **Puppet** 環境を定義します。通常、これは事前に選択されたコンテンツビューおよびライフサイクル環境を使用して定義されます。
- **コンテンツソース:** コンテンツビューからコンテンツをプロビジョニングするために使用する **Capsule** サーバー。
- **Puppet CA:** エージェントの認定に使用する **Capsule** サーバー。
- **Puppet マスター:** エージェント通信のマスターサーバーとして使用する **Capsule** サーバー。

- **Openscap Proxy:** OpenSCAP プロキシとして使用するサーバー。

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

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

- **Type (タイプ):** 使用するインターフェースのタイプ。これには、基本的なイーサネット接続 (**イーサネット**) だけでなく、ベースボード管理コントローラー (**BMC**)、ボンド (**Bond**)、およびブリッジ (**Bridge**) が含まれます。これにより、ホストの複雑なネットワーク設定を作成することができます。
- **MAC (MAC address):** インターフェースの **MAC** アドレス。これを使用して、ネットワークの詳細を特定のインターフェースにマップできます。さらに、プロビジョニングインターフェースの **MAC** アドレスは、**PXE** ブート時にベアメタルホストを特定するために使用されます。
- **デバイス ID:** **eth0**、**ens8**、**bond0**、および **br0** などのインターフェース ID。
- **DNS 名:** ホストのドメイン名。通常、これには**ホスト** タブのホスト名が自動的に設定されます。
- **ドメイン:** ホストのプロビジョニングに使用するドメイン。これと**DNS 名** の組み合わせにより、ホストの完全修飾ドメイン名 (**FQDN**) が作成されます。
- **サブネット:** このインターフェースに接続するネットワーク。
- **IP (IP address):** このインターフェースの **IP** アドレス。選択される**サブネット** とそのオプションにより、このフィールドにはデータが自動的に設定されます。
- インターフェースのセレクションには以下が含まれます。
 - **管理 (Managed):** このインターフェースはプロビジョニング時に **DHCP** および **DNS** で管理されます。
 - **Primary:** これはメインのインターフェースを定義し、インターフェースの詳細からホストの **FQDN** を構築します。
 - **Provision:** このインターフェースは **PXE** ブートサービスに使用されます。
 - **リモート実行:** このインターフェースはリモート実行機能に使用されます。



注記

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

オペレーティングシステム タブでは、ホストにインストールするオペレーティングシステムおよび関連分野を定義します。ホストの**アーキテクチャー** を選択してから、そのアーキテクチャーに関連する**オペレーティングシステム** を選択します。フォームには、選択したオペレーティングシステムに基づ

く追加のオプションが提供されます。

- **ビルド (Build mode):** ホストをプロビジョニングし、オペレーティングシステムを開始するかどうかを定義します。このオプションは、すべてのプロビジョニングタスクで必要になります。すでに存在するプロビジョニングされたホストのエントリーを作成する場合にのみこのオプションを無効にする必要があります。
- **メディア (Media Selection):** 同期したキックスタートのリポジトリのみから、またはすべてのリポジトリから選択するかどうかを定義します。ホストのプロビジョニングに使用するインストールメディアのタイプを選択します。同期したキックスタートリポジトリの場合は **Synced Content** を選択し、**ホスト > インストールメディア > 新規メディア** で手動で追加されたインストールメディアなど、他のインストールメディアから選択する場合は **All Media** を選択します。
- **メディア:** オペレーティングシステムのインストールメディアを定義します。通常、これはキックスタートツリーですが、ローカルにマウントされた ISO イメージにすることもできます。
- **パーティションテーブル:** ルートディスクのレイアウトに使用するパーティションテーブルを定義します。このフォームで **カスタムパーティションテーブル** を直接定義することもできます。
- **ルートパス (Root password):** オペレーティングシステム上の root ユーザーのパスワード。
- **プロビジョニングテンプレート:** ホストをプロビジョニングするために選択されたテンプレートが表示されます。**解決** をクリックすると、**Satellite Server** がテンプレートをプロビジョニングプロセスの特定の機能に割り当てる方法を確認できます (PXE、プロビジョニング、ユーザーデータなど)。



注記

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

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



注記

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

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

ホストのエントリーを保存するには、**送信** をクリックします。

CLI を使用する場合

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

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

--interface オプションを使用して特定のインターフェース設定を行います。詳細は、[付録B Hammer CLI の追加のホストパラメーター](#)を参照してください。**hammer host interface create** コマンドで特定のネットワークインターフェース設定を定義することもできます。**--host** または **--host-id** オプションを使用してインターフェースを受信するホストを選択します。以下は例になります。

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

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

5.3. SATELLITE SERVER でのホストグループの作成

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

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

さらに、ホストグループの階層を作成することもできます。たとえば、オペレーティングシステムを定義するベースレベルのホストグループと、ベースレベルのホストグループを継承する 2 つの子ホストグループを設定するかもしれません。

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

この例では、プロビジョニングされたすべてのホストは **Base** ホストグループの継承により、Red Hat

Enterprise Linux 7.2 をオペレーティングシステムとして使用します。2つの web サーバーホストは **Webserver** ホストグループからの設定を継承します。これには、**httpd Puppet** クラスおよび **Base** ホストグループの設定が含まれます。同様に、2つのストレージサーバーは **Storage** ホストからの設定を継承します。これには、**nfs Puppet** クラスおよび **Base** ホストグループの設定が含まれます。カスタムホストは **Base** ホストグループからの設定のみを継承します。

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

Web UI を使用する場合

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

- **ホスト タブ:**
 - **名前:** ホストグループの名前。このシナリオの場合、**Base** と入力します。
 - **ライフサイクル環境** アプリケーションライフサイクルにおけるホストのステージを定義します。[Red Hat Satellite 6 コンテンツ管理ガイド](#) で作成された **Production (本番)** 環境を選択します。
 - **コンテンツビュー:** リポジトリに使用するコンテンツビューを定義します。[Red Hat Satellite 6 コンテンツ管理ガイド](#) で作成された **Base (ベース)** ビューを選択します。
 - **Puppet 環境:** ホストを含む Puppet 環境を定義します。通常、これは事前に選択されたコンテンツビューおよびライフサイクル環境を使用して定義されます。この例では、Puppet モジュールを含まない **production (本番)** 環境を選択します。
 - **コンテンツソース:** コンテンツビューからコンテンツを提供するために使用する Capsule サーバー。Satellite Server の統合 Capsule を選択します。
 - **Puppet CA:** エージェントの認定に使用する Capsule サーバー。Satellite Server の統合 Capsule を選択します。
 - **Puppet マスター:** エージェント通信のマスターサーバーとして使用する Capsule サーバー。Satellite Server の統合 Capsule を選択します。
 - **Openscap Proxy:** OpenSCAP プロキシとして使用するサーバー。これは空白のままにします。
- **Puppet クラス タブ**では、プロビジョニング後にホストに適用する Puppet クラスを選択します。このシナリオでは Puppet クラスを使用しないため、このタブは省略します。
- **ネットワーク タブ:**
 - **ドメイン:** ホストのプロビジョニングに使用するドメイン。これと**DNS 名**の組み合わせにより、ホストの完全修飾ドメイン名 (FQDN) が作成されます。ACME の **example.com** ドメインを選択します。
 - **サブネット:** このインターフェースに接続するネットワーク。**ACME の内部ネットワーク** を選択します。
 - **レルム:** ホストの認証レルムを定義します。このシナリオではレルムを使用しないのでこのフィールドを空白のままにします。
- **オペレーティングシステム タブ:**

- **アーキテクチャー:** ホストのアーキテクチャー。 **x86_64** を選択します。
- **オペレーティングシステム:** インストールするベースオペレーティングシステム。『**Red Hat Satellite 6 コンテンツ管理ガイド**』の「**Red Hat リポジトリの同期**」セクションで説明されているように、Red Hat Enterprise Linux 7.2 のエントリは同期ステップの実行後に表示されるはずです。
- **メディア (Media Selection):** 同期したキックスタートリポジトリのみから、またはすべてのリポジトリから選択するかどうかを定義します。このホストグループのプロビジョニングに使用するインストールメディアのタイプを選択します。同期したキックスタートリポジトリの場合は **Synced Content** を選択し、**ホスト > インストールメディア > 新規メディア** で手動で追加されたインストールメディアなどの他のインストールメディアから選択する場合は **All Media** を選択します。
 - **メディア:** オペレーティングシステムのインストールメディア。Red Hat Enterprise Linux 7.2 からキックスターツリーを選択します。これは、『**Red Hat Satellite 6 コンテンツ管理ガイド**』の「**Red Hat リポジトリの同期**」セクションに説明されているように、同期ステップの実行後に表示されるはずです。
 - **パーティションテーブル:** ルートディスクのレイアウトに使用するパーティションテーブルを定義します。 **デフォルトキックスター** を選択します。
 - **ルートパス (Root password):** オペレーティングシステム上の root ユーザーのパスワード。 root パスワードを入力します。
- **パラメーター** タブでは、プロビジョニングプロセスと **Puppet** 設定の両方に変数データを設定します。このセクションは空白にします。
- **ロケーション** タブで、ホストグループのロケーションを設定します。この例では、**New York** を選択します。
- **組織** タブで、ホストグループを使用できる組織を設定します。この例では、**ACME** を選択します。
- **アクティベーションキー** タブでは、**サンプル**のアクティベーションキーを選択します。これにより、登録に使用するアクティベーションキーを定義する新規のパラメーター (**kt_activation_keys**) が各ホストに追加されます。

送信 をクリックしてホストグループを保存します。

CLI を使用する場合

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

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

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

5.4. 章の概要

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

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

第6章 ベアメタルホストのプロビジョニング

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

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



注記

以前のバージョンの Red Hat Satellite では、プロビジョニング時にホストグループベースのテンプレートのレンダリング機能を使用でき、この機能により、ユーザーは単一ホストではなくホストグループのテンプレートをレンダリングできました。しかしながら、この機能は、ホストのレコードまたは監査証跡がないことなどの制限により Red Hat Satellite 6.2 ではサポートされていません。そのため、同様の機能を提供する検出機能を使用することをお勧めします。

6.1. ベアメタルプロビジョニングの要件定義

ベアメタルプロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- ベアメタルホストのネットワークを管理する Capsule Server。無人プロビジョニングおよび検出ベースのプロビジョニングの場合、Satellite Server には PXE サーバーの設定が必要です。詳細は、[4章 ネットワークの設定](#) を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。
- テスト目的で使用する空のベアメタルホスト。

6.2. 無人プロビジョニングによる新規ホストの作成

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

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

Web UI を使用する場合

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

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

送信 をクリックします。

CLI を使用する場合

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

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

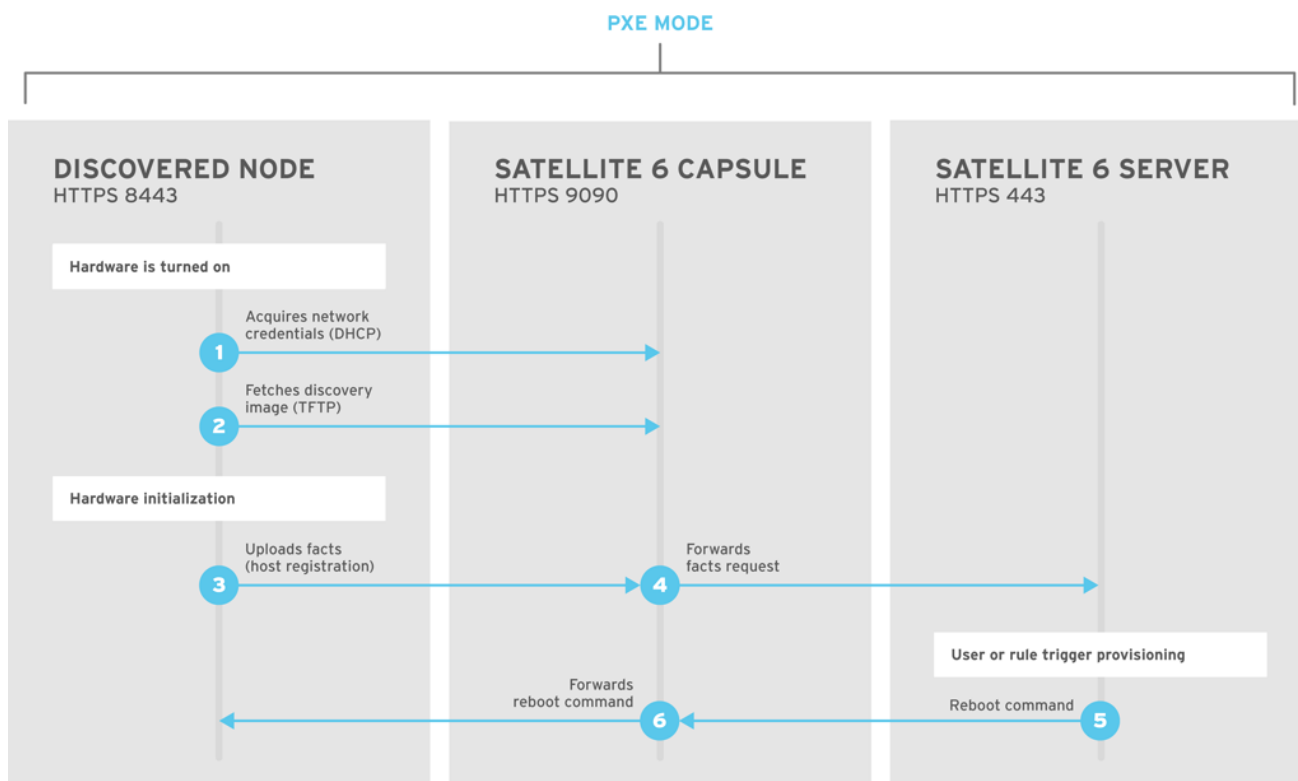
`hammer host interface update` コマンドを使用し、ネットワークインターフェースのオプションが設定されていることを確認します。以下は例になります。

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

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

6.3. RED HAT SATELLITE 検出サービスの設定

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



SATELLITE6_376339_1115

インストール

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

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

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

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

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

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

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

Capsule Server での検出サービスの有効化

以下の手順を実行し、**Capsule Server** で検出サービスを有効にします。

1. 必要な **Capsule Server** で以下のコマンドを実行します。

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

```
# katello-service restart
```

2. **Satellite web UI** にログインし、**インフラストラクチャー > Capsule (スマートプロキシ)** に移動します。
3. 必要な **Capsule Server** をクリックし、更新ボタンをクリックします。**Capsule Server** は実行済みのコマンドに対応しており、設定されているサービスには **discovery** が一覧表示されているのを確認できるでしょう。これは、検出サービスが実行されていることを意味します。

プロビジョニングテンプレート

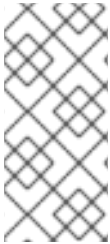
ホスト > プロビジョニングテンプレート セクションの **PXELinux global default** テンプレートには、検出サービスのエントリーが含まれています。

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

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

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

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



注記

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

```
ONTIMEOUT discovery
```

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

サブネット

検出可能なホストを含むすべてのサブネットには、検出サービスを提供するために適切な **Capsule Server** が選択されている必要があります。これを実行するには、**インフラストラクチャー > Capsules (スマートプロキシ)** に移動し、必要な **Capsule Server** に検出機能が一覧表示されているかどうかを確認します。一覧表示がない場合には、**Refresh features (機能の更新)** をクリックすると、それがすぐに表示されます。

インフラストラクチャー > サブネット に移動した後に、サブネットを選択し、**Capsules** タブをクリックしてから必要な **Discovery Proxy (検出プロキシ)** を選択します。該当するそれぞれのサブネットにこれを実行します。

Testing (テスト)

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

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

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

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

注記

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

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

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

6.4. 検出されたホストからの新規ホストの作成

検出されたホストのプロビジョニングについても、同一の基本的なプロビジョニングプロセスを従って実行します。ホストの識別にあたり、ホストの **MAC アドレス** を手動で入力するのではなく、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識別する点が相違点になります。

Web UI を使用する場合

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

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

- 詳細を確認します。
 - **Satellite Server** は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずですが、選択されていない場合は、それらを選択してください。
 - **オペレーティングシステムタブ:**
 - すべてのフィールドには値が自動的に含まれるはずですが、オペレーティングシステムの各要素を確認してください。
 - **プロビジョニングテンプレートの解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。これには、以下が含まれます。
 - **PXELinuxテンプレート:Kickstart default PXELinux**
 - **provisionテンプレート:Satellite Kickstart Default**
- プロビジョニングテンプレートの関連付けについての詳細は、「[プロビジョニングテンプレートの作成](#)」を参照してください。
- **パラメータータブ:**
 - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

プロビジョニングに使用する検出済みホストを特定します。

```
# hammer discovery list
```

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

```
# hammer host create --name "macaaaaaaaaaaaaa" --new-name "baremetal-test2"
--organization "ACME" --location "New York" --hostgroup "Base" --build
true --enabled true --managed true
```

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

6.5. 検出ルールの作成

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の方法として、**Red Hat Satellite 6** は検出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自らを特定のホストグループに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必要があるかもしれません。同様に、ハードディスクの多い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必要があるかもしれません。

Web UI を使用する場合

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

- **名前:** ルールを表すテキスト形式の名前。例:**Hypervisor**
- **Search:** 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に使用するルールを定義します。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**
- **ホストグループ:** 使用するホストグループを定義します。
- **Hostname:** 複数ホストのホスト名を決定するパターンを定義します。これはプロビジョニングテンプレートが使用するのと同じ **ERB** 構文を使用します。ホスト名には、ホスト固有の値に **@host** 属性を使用したり、乱数に **rand** 関数を使用したりできます。
 - **myhost-<%= rand(99999) %>**
 - **abc-<%= @host.facts['bios_vendor'] + '-' + rand(99999) %>**
 - **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を作成する際に、作成されるホスト名が固有の名前であることを確認してください。ホスト名は数字で始めることができません。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。

- **ホストの制限:** ルールを使ってプロビジョニングできるホストの最大数。無制限に設定するには **0** を使用します。
- **優先度:** ルール間の優先度を定義します。値が低いルールほど優先度が高くなります。
- **有効化:** ルールが現時点で有効かどうかを定義します。

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

送信 をクリックしてルールを保存します。

ホスト > 検出されたホスト に移動し、以下のいずれかを選択します。

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

CLI を使用する場合

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

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

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

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

6.6. PXE を使用しないプロビジョニングによる新規ホストの作成

一部のハードウェアは PXE ブートインターフェースを提供しません。ただし、PXE ブートなしに Red Hat Satellite 6 で新規ホストをプロビジョニングすることはできます。これには、ホストが使用できるブート ISO を生成することが関係します。この ISO により、ホストは **Satellite Server** に接続し、インストールメディアを起動し、オペレーティングシステムをインストールできます。ホスト用にブート ISO を作成するプロセスでは、同様の基本的なプロビジョニングプロセスに従います。ISO ファイルを新規ホストのエントリーから生成する点が相違点になります。

Web UI を使用する場合

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

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

- プロビジョニングテンプレートの **解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。これには、以下が含まれます。
 - **bootdisk** テンプレート: **Boot disk iPXE - host**
 - **kexec** テンプレート: **Discovery Red Hat kexec**
 - **provision** テンプレート: **Satellite Kickstart Default**

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

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

送信 をクリックします。

これにより、ホストのエントリーが作成され、ホストの詳細ページが表示されます。ページ右上のオプションには **ブートディスク** メニューが表示され、以下の情報が提供されます。

- **ホストイメージ**: 特定ホストのブート ISO。これには、**Satellite Server** でインストールメディアにアクセスするために必要なブートファイルのみが含まれます。
- **完全ホストイメージ**: 特定ホストのカーネルおよび初期 RAM ディスクイメージを含むブート ISO。
- **汎用イメージ**: 特定ホストに関連付けられていないブート ISO。この場合、ISO はホストの MAC アドレスを **Satellite Server** に送信します。ここでは、ホストのエントリーに対してマッチングが行われます。このイメージは **Satellite Server** の `/bootdisk/disks/generic` URL から利用できます (例: `https://satellite.example.com/bootdisk/disks/generic`)。
- **サブネットイメージ**: **Capsule Server** 経由で通信する汎用ブート ISO。

CLI を使用する場合

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

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

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

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

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

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

--full オプションを使用して完全ホストイメージをダウンロードします。

■

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

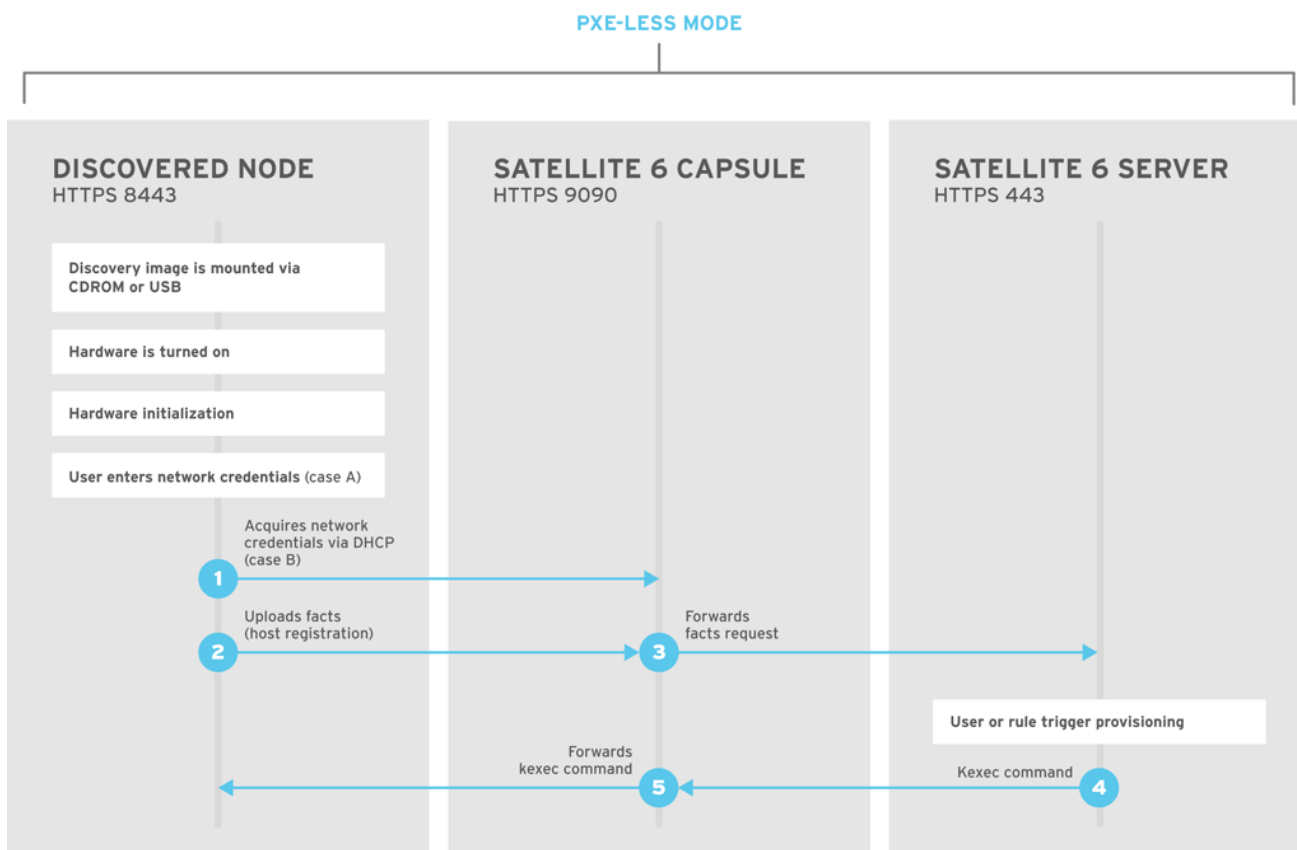
hammer bootdisk generic コマンドで汎用イメージをダウンロードします。

```
# hammer bootdisk generic
```

これによりホストで使用するブート ISO が作成されます。物理ホストの電源を ON にして ISO から起動する場合、ホストは **Satellite Server** に接続し、キックスタートツリーからの **Red Hat Enterprise Linux 7.2** のインストールを開始します。インストールが完了すると、ホストは **サンプル** アクティベーションキーを使用して **Satellite Server** に登録され、**Red Hat Satellite Tools** リポジトリから必要な設定および管理ツールがインストールされます。

6.7. PXE を使用しない検出の実装

Red Hat Satellite 6 は、PXE ベースのサービス (DHCP および TFTP) が必要な場合にも機能する PXE を使用しない検出サービスも提供します。これは、**Satellite Server** の検出イメージを使用して実行できます。



SATELLITE6_376339_1115

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

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

手動による使用

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

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

検出ブートメディアをベアメタルホストに挿入し、ホストを開始し、メディアから起動します。検出イメージには、**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、検出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ネットワークインターフェースが含まれます。この検出イメージは、**IPv4 アドレス**、**IPv4 ゲートウェイ** および **IPv4 DNS** サーバーなどのネットワークインターフェースの設定オプションも要求します。一例として、ACME は以下の詳細を使用する可能性があります。
 - **IPv4 Address: 192.168.140.20**
 - **IPv4 Gateway: 192.168.140.1**
 - **IPv4 DNS: 192.168.140.2 (Satellite Server の統合 Capsule)**
これらの詳細を入力した後に、**次へ** を選択します。
- **Discovery with DHCP** を選択する場合、検出イメージは **Satellite Server** に接続されるプライマリネットワークインターフェースのみを要求します。このサービスは、**Capsule Server** が提供するサーバーなどの DHCP サーバーを使用してネットワークインターフェースを自動的に設定しようとします。

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

https://satellite.example.com:9090

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

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

Satellite は **Satellite Server** の検出サービスとの正常な通信を報告します。ホスト > **検出されたホスト** に移動します。一覧には、新規に検出されたホストが含まれます。

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

手動による使用およびカスタマイズ

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

proxy.url

検出サービスを提供する **Capsule Server** の URL。

proxy.type

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

fdi.pxmac

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

fdi.pxip, fdi.pxgw, fdi.pxdns

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

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

カスタムファクト名を指定できます。

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

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

fdi.pxauto

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

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

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

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

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

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

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

検出されたホストをプロビジョニングするには、「[検出されたホストからの新規ホストの作成](#)」を参照してください。

最終メモ

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

- **kexec** テンプレート: **Discovery Red Hat kexec**

- **provision テンプレート: Satellite Kickstart Default**

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

6.8. 章の概要

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

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

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

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

7.1. KVM プロビジョニングの要件定義

KVM プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- KVM サーバーでネットワークを管理する Capsule Server。Capsule Server との競合を避けるためにその他の DHCP サービスがこのネットワーク上で実行されていないことを確認します。Capsule Server のネットワークサービス設定の詳細は、[4章 ネットワークの設定](#) を参照します。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。
- KVM 仮想化ツールを実行する Red Hat Enterprise Linux サーバー。詳細は、[Red Hat Enterprise Linux 7 仮想化スタートガイド](#) を参照してください。
- (イメージベースのプロビジョニングを使用する場合) 既存の仮想マシンイメージ。このイメージが KVM ホストのストレージプールにあることを確認します。デフォルトのストレージプールは通常 `/var/lib/libvirt/images` にあります。

7.2. SATELLITE SERVER での KVM 接続の設定

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

Satellite Server で `foreman` ユーザーに切り替えます。

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

キーペアを生成します。

```
$ ssh-keygen
```

公開キーを KVM サーバーにコピーします。以下は例になります。

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

以下のコマンドを使用して、KVM サーバーへの接続をテストします。

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

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

7.3. KVM 接続の SATELLITE SERVER への追加

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

Web UI を使用する場合

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

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

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

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

送信 をクリックして KVM 接続を保存します。

CLI を使用する場合

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

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

7.4. SATELLITE SERVER での KVM イメージの追加

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

Web UI を使用する場合

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

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

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

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

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

7.5. KVM の詳細のコンピュータプロファイルへの追加

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

Web UI を使用する場合

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

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

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

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

- **ネットワークタイプ**: ホストおよび NIC のネットワークのタイプを定義します。**物理 (ブリッジ)** または **仮想 (Nat)** のいずれかにすることができます。
- **ネットワーク**: ネットワークタイプの選択内容により、使用するホストの物理インターフェースか、または KVM サーバー上の仮想ネットワークのいずれかになります。
- **NIC タイプ**: **virtio** などの仮想 NIC タイプを定義します。
- **ストレージ**: ホストのボリュームを定義します。ホストに複数のボリュームを作成することができます。ストレージのオプションには以下が含まれます。
 - **ストレージプール**: ボリュームを含む KVM サーバーのプールを定義します。
 - **サイズ**: ボリュームのサイズ (GB) を定義します。
 - **割り当て**: ボリュームに事前に割り当てられた物理スペースのサイズを定義します。
 - **タイプ**: ボリュームのタイプを定義します。**RAW** または **QCOW2** のいずれかになります。

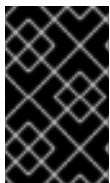
送信 をクリックしてコンピュータプロファイルを保存します。

CLI を使用する場合

コンピュータプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

7.6. KVM サーバーでのネットワークベースのホストの作成

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



重要

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

Web UI を使用する場合

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

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

のフィールドが自動的に設定されます。

- デプロイ先で、KVM 接続を選択します。この例では、**ACME's KVM Server** になります。仮想マシンの新規タブが表示されます。
- コンピュートプロファイルで、仮想マシンベースの設定を自動的にを行うために使用するプロファイルを選択します。例: **4-Example**
- インターフェース タブ:
 - ホストのインターフェースの **編集** をクリックします。
 - ほとんどのフィールドには、値が自動的に含まれるはずですが、とくに以下の点に注意してください。
 - ホストタブの **名前** は **DNS 名** になります。
 - **Satellite Server** は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** を空白にします。KVM サーバーは **MAC アドレス** をホストに割り当てません。
 - **Satellite Server** は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずですが、選択されていない場合は、それらを選択してください。
 - インターフェース画面には、コンピュートプロファイルの設定が入力された **KVM 固有** のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- オペレーティングシステムタブ:
 - すべてのフィールドには値が自動的に含まれるはずですが、オペレーティングシステムの各要素を確認してください。
 - **プロビジョニング方法** が **ネットワークベース** に設定されていることを確認します。
 - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- 仮想マシンタブ:
 - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずですが、必要に応じてこれらの設定を変更します。
- パラメータータブ:
 - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

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

```
# hammer host create --name "kvm-test1" --organization "ACME" --location
  "New York" --hostgroup "Base" --compute-resource "ACME's KVM Server" --
```

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



注記

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

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

7.7. KVM サーバーでのイメージベースのホストの作成

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

Web UI を使用する場合

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

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

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

送信 をクリックします。

CLI を使用する場合

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

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



注記

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

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

7.8. 章の概要

本章では、KVM サーバーを使用するよう Red Hat Satellite 6 を設定し、KVM サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。これには、ネットワークベースのホストおよびイメージベースのホストの両方が含まれました。

Red Hat Satellite 6 で設定する必要があるコンピュートリソースが他にない場合は、[13章 プロビジョニングの最終設定](#) のプロビジョニングについての最終メモを参照してください。

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

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

Red Hat Enterprise Virtualization は、Red Hat Enterprise Linux 上に構築するエンタープライズクラスのサーバーおよびデスクトップの仮想化プラットフォームです。Red Hat Satellite 6 は Red Hat Enterprise Virtualization の REST API で仮想化機能を管理できます。これには、新規仮想マシンの作成やそれらの電源状態の制御が含まれます。本章では、ACME の Red Hat Enterprise Virtualization 環境に接続を追加し、仮想マシンをプロビジョニングします。

8.1. RED HAT ENTERPRISE VIRTUALIZATION のプロビジョニングの要件定義

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

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- Red Hat Enterprise 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 DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、[4章 ネットワークの設定](#) を参照してください。
- イメージベースのプロビジョニングを使用する場合は、**空の** テンプレート以外の既存のテンプレート。仮想マシンのテンプレートを作成する方法についての詳細は、『[仮想マシン管理ガイド](#)』の「[テンプレート](#)」の章を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。

8.2. RED HAT ENTERPRISE VIRTUALIZATION ユーザーの作成

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

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

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

Red Hat Enterprise 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についての詳細は、『Red Hat Enterprise Virtualization 管理ガイド』の「[13.7.1. ユーザーの追加およびユーザーポータルパーミッションの割り当て](#)」を参照してください。

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

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

Web UI を使用する場合

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

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

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

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

送信 をクリックして接続を保存します。

CLI を使用する場合

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

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



注記

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

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

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

Web UI を使用する場合

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

- **名前:** イメージのテキスト形式の名前。例: **Test RHEV Image**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例: **RedHat 7.2**

- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例: `x86_64`
- **ユーザー名:** イメージへのアクセスのための SSH ユーザー名。通常、これは `root` ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ:** Red Hat Enterprise Virtualization 上のイメージの名前。一覧からイメージ名を選択します。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

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

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

8.5. RED HAT ENTERPRISE VIRTUALIZATION の詳細のコンピュートプロファイルへの追加

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

Web UI を使用する場合

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

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

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

- **名前:** ネットワークインターフェースの名前を定義します。
- **ネットワーク:** 使用する論理ネットワークを定義します。
- **ストレージ:** ホストのボリュームを定義します。ホストに複数のボリュームを作成することができます。ストレージのオプションには以下が含まれます。
 - **サイズ (GB):** ボリュームのサイズ (GB) を定義します。
 - **ストレージドメイン:** ボリュームのストレージドメインを定義します。
 - **ディスクの事前割り当て:** シンプロビジョニングまたはフルディスクの事前割り当てを使用するかどうかを定義します。
 - **ブート可能:** ブートボリュームを定義します。

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

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

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



重要

プロビジョニング目的で Red Hat Enterprise Virtualization サーバーで仮想ネットワークを使用している場合は、DHCP 割り当てを提供しないネットワークを選択します。そうしないと、新規ホストの起動時に Satellite Server との DHCP の競合が生じるためです。

Web UI を使用する場合

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

- **ホスト タブ:**
 - ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**rhev-test1** と入力します。
 - プロビジョニングコンテキスト (**組織** および **ロケーション**) は現在のコンテキストに自動的に設定されます。例: **ACME** および **New York**
 - **ホストグループ** フィールドから **Base** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。
 - **デプロイ先** で、Red Hat Enterprise Virtualization 接続を選択します。この例では、**ACME's RHEV** になります。仮想マシンの新規タブが表示されます。

- コンピュートプロファイルで、仮想マシンベースの設定を自動的に行うために使用するプロファイルを選択します。例: **4-Example**
- インターフェース タブ:
 - ホストのインターフェースの **編集** をクリックします。
 - ほとんどのフィールドには、値が自動的に含まれるはずですが、とくに以下の点に注意してください。
 - ホスト タブの **名前** は **DNS 名** になります。
 - **Satellite Server** は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** を空白にします。サーバーは **MAC アドレス** をホストに割り当てます。
 - **Satellite Server** は、ホストの最初のインターフェースで **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずですが、選択されていない場合は、それらを選択してください。
 - インターフェース画面には、コンピュートプロファイルの設定が入力された **Red Hat Enterprise Virtualization** 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- オペレーティングシステムタブ:
 - すべてのフィールドには値が自動的に含まれるはずですが、オペレーティングシステムの各要素を確認してください。
 - **プロビジョニング方法** が **ネットワークベース** に設定されていることを確認します。
 - **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- 仮想マシンタブ:
 - これらの設定には、選択されたホストグループおよびコンピュートプロファイルの詳細が入力されているはずですが、必要に応じてこれらの設定を変更します。
- パラメータータブ:
 - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

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

```
# hammer host create --name "rhev-test1" --organization "ACME" --location
  "New York" --hostgroup "Base" --compute-resource "ACME's RHEV" --
  provision-method build --build true --enabled true --managed true --
  interface
  "managed=true,primary=true,provision=true,compute_name=eth0,compute_networ
```

```
k=satnetwork" --compute-
attributes="cluster=Default,cores=1,memory=1073741824,start=true" --
volume="size_gb=20G,storage_domain=Data,bootable=true"
```



注記

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

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

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

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

Web UI を使用する場合

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

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

- Satellite Server は、ホストについて **Managed (管理)**、**Primary**、および **Provision** オプションを自動的に選択するはずですが、選択されていない場合は、それらを選択してください。
- インターフェース画面には、コンピュータプロファイルの設定が入力された **Red Hat Enterprise Virtualization** 固有のフィールドが表示されます。必要に応じてこれらの設定を変更します。
- **オペレーティングシステムタブ:**
 - すべてのフィールドには値が自動的に含まれるはずですが、オペレーティングシステムの各要素を確認してください。
 - **プロビジョニング方法がイメージベース** に設定されていることを確認します。新規のイメージフィールドが表示されます。このフィールドから、新規ホストの **root** ボリュームのベースとしてイメージを選択できます。これは、**ホスト** タブで選択されたコンピュータプロファイルから自動的に設定されます。
 - **プロビジョニングテンプレートで 解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
- **仮想マシンタブ:**
 - これらの設定には、選択されたホストグループおよびコンピュータプロファイルの詳細が入力されているはずですが、必要に応じてこれらの設定を変更します。
- **パラメータータブ:**
 - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

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

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



注記

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

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

8.8. 章の概要

本章では、Red Hat Enterprise Virtualization サーバーを使用するよう Red Hat Satellite 6 を設定し、Red Hat Enterprise Virtualization サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。これには、ネットワークベースのホストおよびイメージベースのホストの両方が含まれました。

Red Hat Satellite 6 で設定する必要があるコンピュートリソースが他にない場合は、[13章 プロビジョニングの最終設定](#)のプロビジョニングについての最終メモを参照してください。

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

第9章 VMWARE VSPHERE での仮想マシンのプロビジョニング

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

9.1. VMWARE VSPHERE プロビジョニングの要件定義

VMware vSphere プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- vSphere 環境でネットワークを管理する Capsule Server。Capsule Server との競合を避けるために DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、[4 章ネットワークの設定](#) を参照してください。
- 既存の VMware テンプレート (イメージベースのプロビジョニングを使用する場合)。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。

9.2. VMWARE VSPHERE ユーザーの作成

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

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

9.3. VMWARE VSPHERE 接続の SATELLITE SERVER への追加

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

Web UI を使用する場合

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

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

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

送信 をクリックして接続を保存します。

CLI を使用する場合

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

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

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

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

Web UI を使用する場合

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

リックすると、UIに VMware vSphere テンプレートの一連のフィールドが表示されます。

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

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

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

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

9.5. VMWARE VSPHERE の詳細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して実行できます。この例では、**4-Example** プロファイルにいくつかの基本的なハードウェア設定を組み込みます。

Web UI を使用する場合

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

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

- **CPU:** 新規ホストに割り当てる CPU の数。
- **1ソケットあたりのコア数:** 各 CPU に割り当てるコアの数。
- **メモリー:** 新規ホストに割り当てるメモリーの容量。
- **クラスター:** VMware 環境のターゲットホストクラスター。
- **リソースプール:** ホストの利用可能なリソース割り当てを含むリソースプール。

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

送信 をクリックしてコンピュータプロファイルを保存します。

CLI を使用する場合

コンピュータプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

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

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



重要

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

Web UI を使用する場合

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

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

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

送信 をクリックします。

CLI を使用する場合

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

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



注記

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

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

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

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

Web UI を使用する場合

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

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

ロファイルを選択します。例: **4-Example**

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

送信 をクリックします。

CLI を使用する場合

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

```
# hammer host create --name "vmware-test2" --organization "ACME" --
location "New York" --hostgroup "Base" --compute-resource "ACME's RHEV" --
provision-method image --image "Test RHEV Image" --enabled true --managed
true -interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,comput
```

```
e_network=mynetwork" --compute-  
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=  
MyVMs,start=true" --  
volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"
```



注記

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

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

9.8. 章の概要

本章では、VMware vSphere サーバーを使用するよう Red Hat Satellite 6 を設定し、VMware vSphere サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。これには、ネットワークベースのホストおよびイメージベースのホストの両方が含まれました。

Red Hat Satellite 6 で設定する必要があるコンピュートリソースが他にない場合は、[13章 プロビジョニングの最終設定](#)のプロビジョニングについての最終メモを参照してください。

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

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

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

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

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

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- OpenStack 環境でネットワークを管理する Capsule Server。詳細は、[4章 ネットワークの設定](#) を参照してください。
- イメージベースのプロビジョニング用に OpenStack Image Storage (glance) サービスに追加されたイメージ。詳細は、[Red Hat OpenStack Platform インスタンスおよびイメージガイド](#) を参照してください。
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。

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

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

Web UI を使用する場合

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

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

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

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

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

CLI を使用する場合

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

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

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

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

Web UI を使用する場合

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

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

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

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

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

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

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

Web UI を使用する場合

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

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

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

送信 をクリックしてコンピュータプロファイルを保存します。

CLI を使用する場合

コンピュータプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

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

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

Web UI を使用する場合

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

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

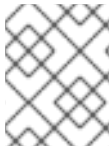
- パラメータータブ:
 - **kt_activation_keys** パラメーターが存在し、**サンプル**のアクティベーションキーを使用していることを確認します。

送信 をクリックします。

CLI を使用する場合

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

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



注記

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

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

10.6. 章の概要

本章では、**Red Hat OpenStack Platform** サーバーを使用するよう **Red Hat Satellite 6** を設定し、**Red Hat OpenStack Platform** サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。これには、ネットワークベースのホストおよびイメージベースのホストの両方が含まれました。

Red Hat Satellite 6 で設定する必要があるコンピュートリソースが他にない場合は、[13章 プロビジョニングの最終設定](#)のプロビジョニングについての最終メモを参照してください。

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

第11章 AMAZON EC2 でのクラウドインスタンスのプロビジョニング

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

11.1. AMAZON EC2 プロビジョニングの要件定義

Amazon EC2 プロビジョニングの要件には以下が含まれます。

- Red Hat Enterprise Linux 7 の同期済みのコンテンツリポジトリ。詳細は、[Red Hat Satellite 6 コンテンツ管理ガイド](#) を参照してください。
- EC2 環境でネットワークを管理する Capsule Server。ホストと Capsule Server 間のネットワークのセキュリティーを確保するために、Virtual Private Cloud (VPC) を使用するのが望ましいと言えます。
- イメージベースのプロビジョニングに選択された Amazon Machine Image (AMI)
- ホスト登録のためのアクティベーションキーのサンプル。詳細は、「[アクティベーションキーの作成](#)」を参照してください。

11.2. AMAZON EC2 接続の SATELLITE SERVER への追加

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



重要

Amazon Web Services は、認証プロセスの一部として時間設定を使用します。これは、Satellite Server の時間が正常に同期される必要があることを意味します。ntpd または chronyd などの NTP サービスが Satellite Server で適切に実行されていることを確認します。Amazon Web Services に正確な時間を指定できないと、認証が失敗する可能性があります。詳細は、『[Red Hat Satellite 6 インストールガイド](#)』の「[時間の同期](#)」を参照してください。

Web UI を使用する場合

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

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

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

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

送信 をクリックして Amazon EC2 接続を保存します。

CLI を使用する場合

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

```
# hammer compute-resource create --name "ACME's EC2" --provider "EC2" --
description "Amazon EC2 Public Cloud" --user "ABCDEFGHJIJ1234567" --
password "*****" --region "us-east-1" --locations "New York" --
organizations "ACME"
```

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

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

Web UI を使用する場合

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

- **名前:** イメージのテキスト形式の名前。例:**Test Amazon EC2 Image**
- **オペレーティングシステム:** イメージのベースオペレーティングシステムを選択するためのフィールド。例:**RedHat 7.2**
- **アーキテクチャー:** オペレーティングシステムのアーキテクチャーを選択するためのフィールド。例:**x86_64**
- **ユーザー名:** イメージへのアクセスのための SSH ユーザー名。通常、これは**root** ユーザーになります。
- **パスワード:** イメージにアクセスするための SSH パスワード。
- **イメージ ID:** イメージの Amazon Machine Image (AMI) ID。通常、この形式は **ami-xxxxxxx** になります。例:**ami-b32c14ad**
- **ユーザーデータ:** イメージが **cloud-init** データなどのユーザーデータ入力をサポートするかどうかを定義します。
- **IAM ロール:** イメージを作成するために使用される Amazon のセキュリティーロール。

送信 をクリックしてイメージの詳細を保存します。

CLI を使用する場合

hammer compute-resource image create コマンドでイメージを作成します。--**uuid** フィールド

ドを使用して Amazon EC2 サーバーのイメージの場所の完全パスを保存します。

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

11.4. AMAZON EC2 の詳細のコンピュートプロファイルへの追加

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

Web UI を使用する場合

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

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

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

送信 をクリックしてコンピュートプロファイルを保存します。

CLI を使用する場合

コンピュートプロファイルの CLI コマンドは Red Hat Satellite 6.2 ではまだ実装されていません。代替方法として、ホストの作成プロセスで同じ設定を直接組み込むことができます。

11.5. AMAZON EC2 でのイメージベースのホストの作成

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

Web UI を使用する場合

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

- **ホスト タブ:**

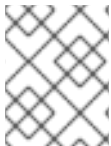
- ホストの **名前** を入力します。これはプロビジョニングされたシステムのホスト名になります。この例では、**ec2-test1** と入力します。
- プロビジョニングコンテキスト (**組織** および **ロケーション**) は現在のコンテキストに自動的に設定されます。例: **ACME** および **New York**
- **ホストグループ** を選択します。これにより、新規ホストのほとんどのフィールドが自動的に設定されます。例: **Base**
- **デプロイ先** で、**EC2 接続** を選択します。この例では、**ACME's EC2** になります。仮想マシ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

送信 をクリックします。

CLI を使用する場合

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

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



注記

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

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

11.6. 章の概要

本章では、**Amazon EC2** サーバーを使用するよう **Red Hat Satellite 6** を設定し、**Amazon EC2** サーバーを使用して新規ホストをプロビジョニングする方法を説明しました。これには、ネットワークベースのホストおよびイメージベースのホストの両方が含まれました。

Red Hat Satellite 6 で設定する必要があるコンピュートリソースが他にない場合は、[13章 プロビジョニングの最終設定](#)のプロビジョニングについての最終メモを参照してください。

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

第12章 コンテナのプロビジョニング

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

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

12.1. コンテナプロビジョニングの要件定義

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

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

12.2. RED HAT ENTERPRISE LINUX ATOMIC HOST の設定

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

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

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

OPTIONS パラメーターを見つけ、API を公開するように編集します。

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



重要

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

Satellite Server 証明書をインポートします。

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

docker サービスを再起動します。

```
$ systemctl restart docker
```

ポートが公開されていることを確認します。

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

12.3. ATOMIC HOST 接続の SATELLITE SERVER への追加

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

Web UI を使用する場合

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

- **名前:** リソースのテキスト形式の名前。例:**ACME's Atomic**
- **プロバイダー:** コンピュートリソースプロバイダーを選択するためのフィールド。**Docker** を選択すると、新規のフィールドのセットが表示されます。
- **説明:** リソースのテキスト形式の説明。例:**ACME's Atomic Host at atomic.example.com**
- **URL:** Atomic Host の Docker API をポイントする URL。例:
http://atomic.example.com:2375
- **ユーザー名、パスワード、Email (電子メール):** Docker ハブの認証の詳細。Satellite Server はこれらの詳細を使用して、Atomic Host に Docker ハブからイメージをダウンロードさせます。これらの詳細は、パブリックイメージまたは Satellite Server で管理されているイメージを使用している場合は不要です。

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

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

CLI を使用する場合

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

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

12.4. 外部レジストリーの SATELLITE SERVER への追加

[Red Hat Satellite 6 コンテンツ管理ガイド](#) では、Red Hat Satellite 6 でコンテンツビューを使って

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

Web UI を使用する場合

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

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

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

送信 をクリックして外部レジストリーを保存します。

CLI を使用する場合

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

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

12.5. SATELLITE SERVER でのコンテナの作成

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

Web UI を使用する場合

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

Preliminary (事前)

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

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

イメージ

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

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

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

設定

このセクションでは、コンテナの初期設定を提供します。

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

環境

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

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

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

CLI を使用する場合

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

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

次に Docker ハブからプロビジョニングを実行します。

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

最後に、外部レジストリーからプロビジョニングを実行します。

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

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

12.6. 章の概要

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

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

第13章 プロビジョニングの最終設定

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

13.1. シナリオ目標の完了

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

Red Hat Satellite 6 でのプロビジョニングの設定

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

ベアメタルホストのプロビジョニング

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

仮想マシンのプロビジョニング

本書では、KVM サーバー、Red Hat Enterprise Virtualization および VMware vSphere からプロビジョニングを実行するいくつかの例を説明しました。

クラウドインスタンスのプロビジョニング

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

コンテナのプロビジョニング

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

13.2. 他のアプリケーションの統合

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

Puppet

各 Capsule Server (統合された Capsule を含む) は Puppet マスターとして機能します。Satellite Server は Puppet エージェントをそれぞれの新規ホストにインストールします。これにより、ホストでリソースおよびサービスを自動的に設定する方法が提供されます。ホストのプロビジョニングプロセスでは、Puppet クラスを Puppet クラス タブで提供できます。詳細は、[Red Hat Satellite 6 Puppet Guide](#) を参照してください。

Red Hat CloudForms

Red Hat CloudForms は Red Hat Satellite 6 に接続して、特定のレベルのプロビジョニングの制御とホスト管理を行います。詳細は、[Red Hat CloudForms Integration with Red Hat Satellite 6 Guide](#) を参照してください。

13.3. 他のドキュメントの参照

以下のガイドでは、Red Hat Satellite 6 の関連情報と追加の例が提供されています。

Red Hat Satellite 6 ホスト設定ガイド

インフラストラクチャー管理やプロビジョニングなどの Red Hat Satellite 6 の主要機能についてのガイド

https://access.redhat.com/documentation/ja-jp/red_hat_satellite/6.2/html/host_configuration_guide/

Red Hat Satellite 6 サーバー管理ガイド

Red Hat Satellite 6 Server の管理についてのガイド

<https://access.redhat.com/documentation/ja-jp/red-hat-satellite/6.2/paged/server-administration-guide>

Red Hat Satellite 6 仮想インスタンスガイド

`virt-who` ツールを使用した仮想インスタンスの管理についてのガイド

<https://access.redhat.com/documentation/ja-jp/red-hat-satellite/6.2/paged/virtual-instances-guide>

Red Hat Satellite 6 Puppet ガイド

独自の Puppet モジュールを構築し、これを Red Hat Satellite 6 にインポートするためのガイド

<https://access.redhat.com/documentation/en/red-hat-satellite/6.2/paged/puppet-guide/>

付録A プロビジョニングのサンプル用の初期化スクリプト

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

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

```
#!/bin/bash

MANIFEST=$1

# Import the content from Red Hat CDN
hammer organization create --name "ACME" --label "ACME" --description "Our
example organization for managing content."
hammer subscription upload --file ~/${MANIFEST} --organization "ACME"
hammer repository-set enable --name "Red Hat Enterprise Linux 7 Server
(RPMs)" --releasever "7Server" --basearch "x86_64" --product "Red Hat
Enterprise Linux Server" --organization "ACME"
hammer repository-set enable --name "Red Hat Enterprise Linux 7 Server
(Kickstart)" --releasever "7Server" --basearch "x86_64" --product "Red Hat
Enterprise Linux Server" --organization "ACME"
hammer repository-set enable --name "Red Hat Satellite Tools 6.2 (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 6.2 for RHEL 7 Server RPMs x86_64" --
organization "ACME"
hammer content-view publish --name "Base" --description "Initial content
view for our operating system" --organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 --
to-lifecycle-environment "Development" --organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 --
to-lifecycle-environment "Testing" --organization "ACME"
hammer content-view version promote --content-view "Base" --version 1 --
to-lifecycle-environment "Production" --organization "ACME"
```

スクリプトに実行可能なパーミッションを設定します。

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


サブスクリプションmanifestのコピーを Red Hat カスタマーポータルからダウンロードし、manifestでスクリプトを実行します。

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

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

付録B HAMMER CLI の追加のホストパラメーター

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

B.1. 共通のインターフェースパラメーター

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

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

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

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

type が **bond** の場合に以下のパラメーターを使用します。

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

type が **bmc** の場合に以下のパラメーターを使用します。

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

B.2. EC2 パラメーター

--compute-attributes の利用可能なパラメーターです。

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

パラメーター	説明
availability_zone	EC2 プロバイダーのリージョンを含む利用可能なゾーン
security_group_ids	使用するセキュリティグループの ID
managed_ip	パブリックまたはプライベート IP を使用するかどうかを定義します。

B.3. LIBVIRT パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明
cpus	CPU 数
memory	メモリー容量 (バイト)
start	マシンを開始するかどうかを定義するブール値

--interface の利用可能なキーです。

パラメーター	説明
compute_type	bridge または network のいずれか
compute_network / compute_bridge	ネットワークまたは物理インターフェースの名前
compute_model	インターフェースモデル。 virtio 、 rtl8139 、 ne2k_pci 、 pcnet 、または e1000 のいずれかになります。

--volume の利用可能なキーです。

パラメーター	説明
pool_name	ボリュームを保管するストレージプール

パラメーター	説明
capacity	ボリュームの容量。例: 10G
format_type	ディスクタイプ。raw または qcow2 のいずれかになります。

B.4. RED HAT OPENSTACK PLATFORM パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明
flavor_ref	使用するフレーバー
image_ref	使用するイメージ
tenant_id	使用するテナント
security_groups	使用するセキュリティーグループの一覧
network	インスタンスを接続するためのネットワーク

B.5. RED HAT ENTERPRISE VIRTUALIZATION パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明
cluster	ホストを取得するためのクラスター ID
template	使用するハードウェアプロファイル
cores	使用する CPU コア数
memory	メモリー容量 (バイト)

--interface の利用可能なキーです。

パラメーター	説明
compute_name	インターフェース名。例: eth0
compute_network	使用するクラスター内のネットワーク

--volume の利用可能なキーです。

パラメーター	説明
size_gb	ボリュームサイズ (GB 単位)
storage_domain	使用するストレージドメインを定義します。
bootable	root ボリュームを定義するブール値。1つのボリュームのみを起動できます。

B.6. VMWARE INTERFACE パラメーター

--compute-attributes の利用可能なキーです。

パラメーター	説明
cpus	ホストの CPU 数
corespersocket	CPU ソケットあたりのコア数。ハードウェアの 10 未満のバージョンを使用するホストにのみ適用されます。
memory_mb	メモリーサイズ (MB)
cluster	ホストのクラスター ID
path	ホストを整理するためのフォルダーへのパス
guest_id	ゲスト OS ID
scsi_controller_type	VMware コントローラーの ID
hardware_version	VMware ハードウェアのバージョン ID

パラメーター	説明
start	マシンを開始するかどうかを定義するブール値

--interface の利用可能なキーです。

パラメーター	説明
compute_type	ネットワークアダプターのタイプ。VirtualVmxnet、VirtualVmxnet2、VirtualVmxnet3、VirtualE1000、VirtualE1000e、VirtualPCNet32 のいずれかになります。
compute_network	VMware ネットワーク ID

--volume の利用可能なキーです。

パラメーター	説明
datastore	データストア ID
name	ボリュームの名前
size_gb	サイズ (GB 単位)
thin	シンプロビジョニングを使用するかどうかを定義するブール値
eager_zero	Eager Zero シックプロビジョニングを使用するかどうかを定義するブール値

付録C FIPS 準拠ホストのプロビジョニング

Red Hat Satellite 6 は National Institute of Standards and Technology の [暗号モジュールのセキュリティ要件 \(Security Requirements for Cryptographic Modules\)](#) 標準 (参照番号 FIPS 140-2、FIPS と呼ばれる。) に準拠するプロビジョニングホストをサポートします。

Red Hat Satellite 6 は FIPS が有効なホストではサポートされません。

FIPS に準拠するホストのプロビジョニングを有効にするには、以下の変更を実行します。

- 関連するオペレーティングシステム、ロケーションおよび組織の特定
- FIPS プロビジョニングテンプレートの作成および有効化
- プロビジョニングのパスワードハッシュアルゴリズムの変更
- Puppet のメッセージダイジェストアルゴリズムの変更
- FIPS 有効化パラメーターの設定

これらの変更が完了したら、新規のプロビジョニングテンプレートが指定のオペレーティングシステム、ロケーションおよび組織に関連付けられます。ホストをそれらのオペレーティングシステム、ロケーション、および組織にプロビジョニングする際に、ホストには FIPS 準拠の設定が適用されます。これらの設定が正常に行われていることを確認するには、「[FIPS モードの有効化の確認](#)」のステップを実行します。

前提条件

- 『Hammer CLI ガイド』の「[認証](#)」セクションにある設定ステップを実行します。これにより、毎回 Satellite ユーザー名およびパスワードを指定せずに Hammer コマンドを実行できます。

C.1. 関連するオペレーティングシステム、ロケーションおよび組織の特定

FIPS 準拠のテンプレートを Satellite で作成する前に、FIPS 準拠のホストをデプロイするロケーション、組織およびオペレーティングシステムを特定する必要があります。たとえば、Red Hat Enterprise Linux 7 ホストを FIPS 準拠のホストとしてデプロイする場合、テンプレートは Red Hat Enterprise Linux 7 にのみ関連付けます。

1. すべてのロケーションを一覧表示します。

例

```
$ hammer location list
---|-----
ID | NAME
---|-----
2  | Default Location
---|-----
```

NAME 列にある、FIPS 準拠のホストをデプロイするロケーションの値をメモします。

2. すべての組織を一覧表示します。

例


```

---|-----|-----|-----
ID | NAME                | LABEL                | DESCRIPTION
---|-----|-----|-----
1  | Default Organization | Default_Organization |
2  | Sales                | Sales_Department     |
---|-----|-----|-----

```

NAME 列にある、FIPS 準拠のホストをデプロイする組織の値をメモします。

- すべてのオペレーティングシステムを一覧表示します。

例

```

$ hammer os list
---|-----|-----|-----
ID | TITLE                | RELEASE NAME | FAMILY
---|-----|-----|-----
2  | RedHat 6.6           |               | Redhat
3  | RedHat 7.1           |               | Redhat
1  | RedHat 7.2           |               | Redhat
4  | RedHat 6.7           |               | Redhat
---|-----|-----|-----

```

TITLE 列にある、FIPS 準拠のホストをデプロイするオペレーティングシステムの値をメモします。

C.2. FIPS プロビジョニングテンプレートの作成および有効化

FIPS プロビジョニングテンプレートは **git** リポジトリで提供されます。この手順では、それらを **Satellite** 環境にインポートした後に、必要なオペレーティングシステム、ロケーションおよび組織に関連付けます。

- Satellite Server** では、FIPS が有効なテンプレートを含む **git** リポジトリのクローンを作成してから、リポジトリのディレクトリーに切り替えます。

```

$ git clone https://github.com/RedHatSatellite/satellite6-fips-
client
$ cd satellite6-fips-client

```

このリポジトリには、以下の **Embedded RuBy (ERB)** テンプレートが含まれます。それらはテキスト形式のため、テンプレートに含まれる設定内容の詳細を表示し、確認することができます。

- **Kickstart_Default_PXELinux_FIPS.erb**
 - 更新済みの PXELinux テンプレート
- **fips_packages.erb**
 - FIPS モードに必要なパッケージ (例: **dracut-fips**)
- **Satellite_Kickstart_Default_FIPS.erb**
 - **fips_packages** スニペットを呼び出すための修正を含むキックスタートテンプレート

- `puppet.conf.erb`

- 更新された (SHA256) メッセージダイジェストアルゴリズムを含む更新済みの `puppet.conf` 設定ファイル

2. PXELinux FIPS テンプレートを追加します。

```
$ hammer template create --name "Kickstart Default PXELinux FIPS" \
  --file Kickstart_Default_PXELinux_FIPS.erb \
  --locations LOCATIONS \
  --organizations ORGANIZATION \
  --operatingsystems OS \
  --type PXELinux
```

プレースホルダーの値 **LOCATIONS**、**ORGANIZATION**、および **OS** を「[関連するオペレーティングシステム、ロケーションおよび組織の特定](#)」でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create --name "Kickstart Default PXELinux FIPS" \
  --file Kickstart_Default_PXELinux_FIPS.erb \
  --locations "Default Location" \
  --organizations "Default Organization","Sales" \
  --operatingsystems "RedHat 6.6","RedHat 7.1","RedHat 7.2","RedHat 6.7" \
  --type PXELinux
```

3. Satellite Kickstart Default FIPS テンプレートを追加します。

```
$ hammer template create --name "Satellite Kickstart Default FIPS" \
  --file Satellite_Kickstart_Default_FIPS.erb \
  --locations LOCATIONS \
  --organizations ORGANIZATION \
  --operatingsystems OS \
  --type provision
```

プレースホルダーの値 **LOCATIONS**、**ORGANIZATION**、および **OS** を「[関連するオペレーティングシステム、ロケーションおよび組織の特定](#)」でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create --name "Satellite Kickstart Default FIPS" \
  --file Satellite_Kickstart_Default_FIPS.erb \
  --locations "Default Location" \
  --organizations "Default Organization","Sales" \
```

```
--operatingsystems "RedHat 6.6","RedHat 7.1","RedHat 7.2","RedHat
6.7" \
--type provision
```

4. FIPS Packages スニペットを追加します。

```
$ hammer template create --name "fips_packages" \
--file fips_packages.erb \
--locations LOCATIONS \
--organizations ORGANIZATION \
--type snippet
```

プレースホルダーの値 **LOCATIONS** および **ORGANIZATION** を「[関連するオペレーティングシステム、ロケーションおよび組織の特定](#)」でメモした値に置き換えます。値にアルファベット以外の文字が含まれる場合は、値を引用符 (") で囲みます。

Config template created というメッセージは成功したことを示します。

例

```
$ hammer template create --name "fips_packages" \
--file fips_packages.erb \
--locations "Default Location" \
--organizations "Default Organization","Sales" \
--type snippet
```

5. デフォルトの Puppet 設定スニペットを更新します。

```
$ hammer template update --name puppet.conf \
--file puppet.conf.erb \
--type snippet
```

Config template created というメッセージは成功したことを示します。

6. 新規テンプレートを使用するようにオペレーティングシステムオブジェクトを更新します。新規の FIPS テンプレートが **Satellite** に追加され、必要なオペレーティングシステムのデフォルト テンプレートとして設定されます。

- a. **Satellite Kickstart Default FIPS** および **Kickstart Default PXELinux FIPS** テンプレートの ID を特定します。

例

```
$ hammer template list
---|-----|-----
ID | NAME                                | TYPE
---|-----|-----
41 | redhat_register                     | snippet
42 | saltstack_minion                    | snippet
53 | Kickstart Default PXELinux FIPS     | PXELinux
46 | Satellite Kickstart Default         | provision
48 | Satellite Kickstart Default Finish  | finish
54 | Satellite Kickstart Default FIPS    | provision
47 | Satellite Kickstart Default User Data | user_data
50 | subscription_manager_registration   | snippet
```

```

29 | UserData default | user_data
30 | WAIK default PXELinux | PXELinux
---|-----|-----

```

この例では、ID はそれぞれ **54** および **53** になります。これらの ID はインストールに固有の ID です。

- b. FIPS テンプレートをデフォルトとして指定します。

```

$ hammer os set-default-template --config-template-id TEMPLATE --id OS

```

プレースホルダー **TEMPLATE** および **OS** を、FIPS テンプレートと先にメモしておいたオペレーティングシステムの ID に置き換えます。FIPS テンプレートとオペレーティングシステムのすべての組み合わせに対してこのコマンドを繰り返し実行します。コンマ区切りの値の一覧は使用できません。

この例では、FIPS テンプレートが前の例で ID 1 と特定された Red Hat Enterprise Linux 7.2 のデフォルトとして設定されています。

例

```

$ hammer os set-default-template --config-template-id 54 --id 1
$ hammer os set-default-template --config-template-id 53 --id 1

```

C.3. プロビジョニングのパスワードハッシュアルゴリズムの変更

プロビジョニングで使用されるパスワードのハッシュアルゴリズムを **SHA256** に設定します。この設定は、FIPS 準拠としてデプロイする各オペレーティングシステムに適用される必要があります。



注記

これは Red Hat Satellite 6 が Satellite 6.1 からアップグレードされた場合のみ必要になります。Satellite 6.2 はデフォルトで **SHA256** を使用します。

1. オペレーティングシステム ID を特定します。

例

```

$ hammer os list
---|-----|-----|-----
ID | TITLE          | RELEASE NAME | FAMILY
---|-----|-----|-----
2  | RedHat 6.6     |              | Redhat
3  | RedHat 7.1     |              | Redhat
1  | RedHat 7.2     |              | Redhat
4  | RedHat 6.7     |              | Redhat
---|-----|-----|-----

```

2. 各オペレーティングシステムのパスワードハッシュ値を更新します。

```

$ hammer os update --title OS \
  --password-hash SHA256

```

TITLE 列の一致する値を使用して、必要なオペレーティングシステムのそれぞれについてこのコマンドを繰り返し実行します。コマンド区切りの値の一覧は使用できません。

例

```
$ hammer os update --title "RedHat 7.2" \  
--password-hash SHA256
```

C.4. PUPPET の FIPS 準拠メッセージアルゴリズムへの切り替え

Satellite Server やすべての外部 Capsule Server およびすべての既存ホストで SHA256 メッセージダイジェストアルゴリズムを使用するように Puppet を設定します。

`digest_algorithm = sha256` の行を `[main]` スタンザに追加して、`/etc/puppet/puppet.conf` ファイルを編集します。



注記

この変更は Satellite がアップグレードされるたびに上書きされるため、再度適用する必要があります。

Puppet メッセージダイジェストアルゴリズムは Satellite Server およびすべての Capsule Server で変更されるため、FIPS 準拠でないものも含め、すべてのホストで変更する必要があります。

メッセージダイジェストアルゴリズムに不一致がある場合、クライアントはファクトを再度ダウンロードします。これにより、Satellite Server または外部の Capsule Server での負荷が大幅に増大します。

C.5. FIPS 有効化パラメーターの設定

FIPS に準拠したホストをプロビジョニングするには、FIPS テンプレートで `fips_enabled` という名前のパラメーターを `true` に設定する必要があります。これが `true` に設定されていないか、またこれがない場合は、FIPS 固有の変更は適用されません。このパラメーターは、個別のホストまたはホストグループセットのプロビジョニング時に指定できます。遡及的にホストで FIPS 準拠を有効にすることについては、本書の対象外であり、かつこれを実行すると問題が生じる可能性があります。

ホストのプロビジョニング時にこのパラメーターを設定するには、`--parameters fips_enabled=true` を Hammer コマンドに追加します。

既存のホストグループでこのパラメーターを設定するには、Hammer サブコマンドの `set-parameter` を使用します。詳細は、コマンド `hammer hostgroup set-parameter --help` の出力を参照してください。このホストグループにプロビジョニングされるすべてのホストはホストグループから `fips_enabled` パラメーターを継承します。

例

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

C.6. FIPS モードの有効化の確認

これらの FIPS 準拠に関する変更が正常に行われたことを確認するには、ホストのプロビジョニングを実行し、その設定を確認する必要があります。

1. FIPS テンプレートを使用してホストをデプロイし、**fips_enabled** という名前のパラメーターが **true** に設定されていることを確認します。
2. **root** と同等のアカウントで新規ホストにログインします。
3. コマンド **cat /proc/sys/crypto/fips_enabled** を実行します。値 **1** は FIPS モードが有効にされていることを示します。