



Red Hat Satellite 6.14

ホストのプロビジョニング

プロビジョニングリソースとネットワークの設定、物理マシンのプロビジョニング、クラウドプロバイダーまたは仮想化インフラストラクチャーへの仮想マシンのプロビジョニング、Discovery サービスまたは手動によるホストの作成

Red Hat Satellite 6.14 ホストのプロビジョニング

プロビジョニングリソースとネットワークの設定、物理マシンのプロビジョニング、クラウドプロバイダーまたは仮想化インフラストラクチャーへの仮想マシンのプロビジョニング、Discovery サービスまたは手動によるホストの作成

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

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

Linux[®] 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 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解説します。本書には、必要なネットワークポロジのセットアップや、必要なサービスの設定、ネットワーク上でのホストのプロビジョニングに必要な他の設定に関する情報すべてが記載されています。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 プロビジョニングについて	6
1.1. プロビジョニングの概要	6
1.2. サポートされているクラウドプロバイダー	6
1.3. サポートされている仮想化インフラストラクチャー	6
1.4. ネットワークブートのプロビジョニングワークフロー	7
第2章 プロビジョニングリソースの設定	10
2.1. プロビジョニングコンテキスト	10
2.2. プロビジョニングコンテキストの設定	10
2.3. オペレーティングシステムの作成	10
2.4. 複数のオペレーティングシステムの詳細更新	12
2.5. アーキテクチャーの作成	12
2.6. ハードウェアモデルの作成	13
2.7. ホストのオペレーティングシステム用の同期済みキックスタートリポジトリの使用	13
2.8. SATELLITE へのインストールメディアの追加	14
2.9. パーティションテーブルの作成	15
2.10. 動的パーティションの例	17
2.11. プロビジョニングテンプレート	17
2.12. プロビジョニングテンプレートのタイプ	18
2.13. プロビジョニングテンプレートの作成	20
2.14. プロビジョニングテンプレートのクローン作成	20
2.15. カスタムプロビジョニングスニペットの作成	21
2.16. RED HAT ENTERPRISE LINUX のカスタムプロビジョニングスニペットの例	22
2.17. テンプレートとオペレーティングシステムの関連付け	22
2.18. コンピュートプロファイルの作成	23
2.19. ホストへのデフォルト暗号化 ROOT パスワードの設定	24
2.20. NOVNC を使用した仮想マシンへのアクセス	25
第3章 ネットワークの設定	26
3.1. ファクトと NIC フィルタリング	26
3.2. データベースから NIC を削除することによるパフォーマンスの最適化	26
3.3. ネットワークリソース	27
3.4. SATELLITE および DHCP オプション	28
3.5. SATELLITE で DHCP の問題をトラブルシューティングする手順	29
3.6. イメージベースのプロビジョニングの要件	31
3.7. ネットワークサービスの設定	32
3.8. SATELLITE SERVER へのドメインの追加	35
3.9. SATELLITE SERVER へのサブネットの追加	36
第4章 INFOBLOX を DHCP および DNS プロバイダーとして使用する	39
4.1. INFOBLOX の制限	39
4.2. INFOBLOX の前提条件	39
4.3. CAPSULE SERVER への INFOBLOX の CA 証明書のインストール	39
4.4. DHCP INFOBLOX モジュールのインストール	40
4.5. DNS INFOBLOX モジュールのインストール	40
第5章 PXE を使用したホストのプロビジョニング	42
5.1. ベアメタルプロビジョニングの前提条件	42
5.2. セキュリティートークンの有効期間の設定	43
5.3. 無人プロビジョニングによるホストの作成	43

5.4. PXE を使用しないプロビジョニングによるホストの作成	45
5.5. UEFI HTTP ブートプロビジョニングを使用したホストの作成	47
5.6. プロビジョニング中の SSH キーのデプロイ	50
第6章 IPXE を使用したプロビジョニング時間の短縮	52
6.1. IPXE を使用するための前提条件	52
6.2. IPXE 環境の設定	52
6.3. 仮想マシンの起動	53
6.4. PXELINUX からの IPXE のチェーンブート	55
第7章 DISCOVERY サービスの設定	57
7.1. DISCOVERY サービスのインストール	58
7.2. テンプレート PXELINUX DISCOVERY スニペットのプロビジョニング	58
7.3. 検出されたホストの自動コンテキスト	59
7.4. DISCOVERY テンプレートとスニペットの設定	60
7.5. 検出されたホストからのホストの作成	61
7.6. DISCOVERY ルールの作成	62
7.7. PXE を使用しない DISCOVERY の実装	64
7.8. 無人での使用、カスタマイズ、イメージのリマスター	66
7.9. DISCOVERY イメージの拡張	68
7.10. DISCOVERY のトラブルシューティング	69
第8章 RED HAT IMAGE BUILDER イメージを使用したプロビジョニング	71
第9章 KVM (LIBVIRT) での仮想マシンのプロビジョニング	72
9.1. SATELLITE SERVER の KVM 接続設定	72
9.2. SATELLITE SERVER への KVM 接続の追加	73
9.3. SATELLITE SERVER での KVM イメージの追加	74
9.4. KVM の詳細のコンピュートプロファイルへの追加	75
9.5. KVM でのホストの作成	76
第10章 RED HAT VIRTUALIZATION での仮想マシンのプロビジョニング	79
10.1. SATELLITE SERVER への RED HAT VIRTUALIZATION 接続の追加	80
10.2. RED HAT VIRTUALIZATION での CLOUD-INIT イメージの準備	81
10.3. SATELLITE SERVER への RED HAT VIRTUALIZATION イメージの追加	81
10.4. CLOUD-INIT テンプレートの準備	82
10.5. RED HAT VIRTUALIZATION の詳細のコンピュートプロファイルへの追加	84
10.6. RED HAT VIRTUALIZATION でのホストの作成	85
第11章 VMWARE VSPHERE での仮想マシンのプロビジョニング	88
11.1. VMWARE プロビジョニングの前提条件	88
11.2. VMWARE ユーザーの作成	88
11.3. SATELLITE SERVER への VMWARE 接続の追加	89
11.4. VMWARE イメージを SATELLITE SERVER に追加	91
11.5. コンピューティングプロファイルへの VMWARE の詳細の追加	91
11.6. VMWARE でのホストの作成	93
11.7. プロビジョニングでの VMWARE CLOUD-INIT と USERDATA テンプレートの使用	95
11.8. VMWARE 上の VM の削除	100
11.9. VMWARE から SATELLITE への仮想マシンのインポート	100
11.10. コンピュートリソースのキャッシング	100
第12章 OPENSIFT VIRTUALIZATION での仮想マシンのプロビジョニング	102
12.1. SATELLITE SERVER への OPENSIFT VIRTUALIZATION 接続の追加	102
第13章 RED HAT OPENSTACK PLATFORM でのクラウドインスタンスのプロビジョニング	104

13.1. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM 接続の追加	104
13.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM イメージの追加	105
13.3. RED HAT OPENSTACK PLATFORM の詳細のコンピュートプロファイルへの追加	106
13.4. RED HAT OPENSTACK PLATFORM でのイメージベースのホストの作成	107
第14章 AMAZON EC2 でのクラウドインスタンスのプロビジョニング	110
14.1. AMAZON EC2 プロビジョニングの要件	110
14.2. AMAZON EC2 プラグインのインストール	110
14.3. SATELLITE SERVER への AMAZON EC2 接続の追加	110
14.4. コンピュートリソースと HTTP プロキシとの連携	112
14.5. AMAZON EC2 のイメージの作成	112
14.6. SATELLITE SERVER への AMAZON EC2 イメージの追加	113
14.7. AMAZON EC2 の詳細のコンピュートプロファイルへの追加	114
14.8. AMAZON EC2 でのイメージベースのホストの作成	114
14.9. SSH での AMAZON EC2 インスタンスへの接続	116
14.10. AMAZON WEB SERVICE EC2 環境向け FINISH テンプレートの設定	117
14.11. AMAZON EC2 での仮想マシンの削除	118
14.12. AMAZON WEB SERVICES および SATELLITE に関する詳細情報	118
第15章 GOOGLE COMPUTE ENGINE でのクラウドインスタンスのプロビジョニング	119
15.1. SATELLITE SERVER への GOOGLE GCE 接続の追加	119
15.2. SATELLITE SERVER への GOOGLE COMPUTE ENGINE イメージの追加	120
15.3. コンピュティングプロファイルへの GOOGLE GCE の詳細の追加	121
15.4. GOOGLE COMPUTE ENGINE でのイメージベースのホストの作成	122
15.5. GOOGLE GCE での VM の削除	124
第16章 MICROSOFT AZURE RESOURCE MANAGER でのクラウドインスタンスのプロビジョニング	125
16.1. MICROSOFT AZURE RESOURCE MANAGER 接続の SATELLITE SERVER への追加	125
16.2. MICROSOFT AZURE RESOURCE MANAGER イメージの SATELLITE SERVER への追加	127
16.3. MICROSOFT AZURE RESOURCE MANAGER の詳細のコンピュティングプロファイルへの追加	128
16.4. MICROSOFT AZURE RESOURCE MANAGER でのイメージベースのホスト作成	130
16.5. MICROSOFT AZURE 上の VM の削除	132
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	133
付録B FIPS 準拠ホストのプロビジョニング	135
B.1. プロビジョニングパスワードハッシュアルゴリズムの変更	135
B.2. FIPS 有効化パラメーターの設定	135
B.3. FIPS モードの有効化の確認	136
付録C RED HAT SATELLITE 向けクラウドイメージの構築	137
C.1. RED HAT ENTERPRISE LINUX のカスタムイメージの作成	137
C.2. 登録時にサポートされるクライアント	138
C.3. ホスト登録の設定	138
C.4. ホストの登録	138
C.5. KATELLO エージェントのインストール	141
C.6. PUPPET AGENT の手動でのインストールと設定	142
C.7. RED HAT ENTERPRISE LINUX 7 イメージの完了	143
C.8. RED HAT ENTERPRISE LINUX 6 イメージの完了	144
C.9. 次のステップ	146
付録D ホストパラメーター階層	147
付録E ホストをプロビジョニングするために必要なパーミッション	148

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。また、改善点があればお知らせください。

フィードバックを提供するには、Red Hat Jira の **Create Issue** フォームを使用します。Jira の問題は Red Hat Satellite Jira プロジェクトに作成され、その進捗状況を追跡できます。

前提条件

- [Red Hat アカウント](#) が登録されている。

手順

1. **Create Issue** にアクセスします。Jira でログインエラーが表示された場合は、フォームにリダイレクトされた後、ログインして続行します。
2. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
3. **Create** をクリックします。

第1章 プロビジョニングについて

1.1. プロビジョニングの概要

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

プロビジョニングには多数の方法があります。たとえば、Satellite Server の統合 Capsule や外部 Capsule Server は、PXE ベースと PXE を使用しない方法の両方を使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、ホストの作成、管理、更新など Red Hat Satellite アプリケーションライフサイクルに含まれます。

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

ベアメタルプロビジョニング

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

Cloud Providers

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

仮想化インフラストラクチャー

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

1.2. サポートされているクラウドプロバイダー

次のクラウドプロバイダーをコンピューティングリソースとして Satellite に接続できます。

- Red Hat OpenStack Platform
- Amazon EC2
- Google Compute Engine
- Microsoft Azure

1.3. サポートされている仮想化インフラストラクチャー

次の仮想化インフラストラクチャーをコンピューティングリソースとして Satellite に接続できます。

- KVM (libvirt)
- Red Hat Virtualization (非推奨)
- VMware
- OpenShift Virtualization

1.4. ネットワークブートのプロビジョニングワークフロー

物理または仮想 BIOS ホストの場合:

1. 最初の起動デバイスをネットワークを使用した起動設定として設定します。
2. 2番目の起動デバイスをハードドライブからの起動として設定します。Satellite は TFTP ブート設定ファイルを管理するため、再起動するだけでホストを簡単にプロビジョニングできます。

物理または仮想 EFI ホストの場合:

1. 最初の起動デバイスをネットワークを使用した起動設定として設定します。
2. EFI ファームウェアのタイプおよび設定に応じて、OS インストーラーは通常、OS ブートローダーを最初のエントリーとして設定します。
3. インストーラーを再起動するには、**efibootmgr** ユーティリティを使用して、ネットワークからの起動に切り替えます。

プロビジョニングプロセスは、基本的な PXE のワークフローに従います。

1. ホストを作成して、ドメインとサブネットを選択します。Satellite は、サブネットに関連付けられている DHCP Capsule Server、または Satellite の PostgreSQL データベースから、利用可能な IP アドレスを要求します。次に、Satellite は、この IP アドレスを Create Host ウィンドウの **IP address** フィールドに投入します。新しいホストのオプションをすべて入力したら、新しいホストリクエストを送信します。
2. ホストとそのドメインおよびサブネットの設定仕様にに応じて、Satellite は次の設定を作成します。
 - サブネットに関連付けられた Capsule Server の DHCP レコード
 - ドメインに関連付けられた Capsule Server の正引き DNS レコード
 - サブネットに関連付けられた DNS Capsule Server の逆引き DNS レコード
 - サブネットが関連付けられた TFTP Capsule Server にあるホストの PXELinux、Grub、Grub2、iPXE 設定ファイル
 - 関連する Puppet サーバーの Puppet 証明書
 - 関連付けられた ID サーバーのレルム
3. ホストは、ネットワークから最初のデバイスとして起動し、HDD を 2 番目のデバイスとして起動するように設定されています。
4. 新規ホストが DHCP サーバーから DHCP 予約を要求します。
5. DHCP サーバーは予約要求に応答し、TFTP の **next-server**、**filename** オプションを返します。
6. ホストは、PXELoader の設定をもとに、TFTP サーバーからブートローダーやメニューを要求します。
7. ブートローダーは TFTP 経由で返されます。

8. ブートローダーはプロビジョニングインターフェイスの MAC アドレスを介してホストの設定を取得します。
9. ブートローダーはオペレーティングシステムのインストーラーカーネル、初期 RAM ディスク、およびブートパラメーターを取得します。
10. インストーラーは Satellite からプロビジョニングテンプレートを要求します。
11. Satellite はプロビジョニングテンプレートをレンダリングし、結果をホストに返します。
12. インストーラーはオペレーティングシステムをインストールします。
 - インストーラーは、Subscription Manager を使用してホストを Satellite に登録します。
 - インストーラーは **katello-agent** や **puppet** などの管理ツールをインストールします。
 - インストーラーは Satellite に対し、**postinstall** スクリプトで正常なビルドについて通知します。
13. PXE 設定ファイルはローカルブートテンプレートに戻ります。
14. ホストは再起動します。
15. 新規ホストが DHCP サーバーから DHCP 予約を要求します。
16. DHCP サーバーは予約要求に応答し、TFTP の **next-server**、**filename** オプションを返します。
17. ホストは、PXELoader の設定をもとに、TFTP サーバーからブートローダーやメニューを要求します。
18. ブートローダーは TFTP 経由で返されます。
19. ブートローダーはプロビジョニングインターフェイスの MAC アドレスを介してホストの設定をフェッチします。
20. ブートローダーはローカルドライブから起動を開始します。
21. ホストで Puppet クラスを使用するように設定している場合は、ホストはモジュールを使用して設定を行います。

完全にプロビジョニングされたホストは、次のワークフローを実行します。

1. ホストは、ネットワークから最初のデバイスとして起動し、HDD を 2 番目のデバイスとして起動するように設定されています。
2. 新規ホストが DHCP サーバーから DHCP 予約を要求します。
3. DHCP サーバーは予約要求に応答し、TFTP の **next-server**、**filename** オプションを返します。
4. ホストは、PXELoader の設定をもとに、TFTP サーバーからブートローダーやメニューを要求します。
5. ブートローダーは TFTP 経由で返されます。
6. ブートローダーは、プロビジョニングインターフェイスの MAC アドレスを介してホストの設定を取得します。

7. BIOS ホストの場合:

- ブートローダーは起動不可能なデバイスを返すため、BIOS は次のデバイスにスキップしません (HDD からの起動)。

8. EFI ホストの場合:

- ブートローダーは ESP パーティションで Grub2 を見つけ、それをチェーンブートします。

9. ホストが Satellite に認識されていない場合は、デフォルトのブートローダー設定が提供されません。Discovery サービスが有効になっている場合は、Discovery で起動します。それ以外の場合は、HDD から起動します。

ワークフローはカスタムのオプションにより異なります。以下に例を示します。

検出

検出サービスを使用している場合は、Satellite は新規ホストの MAC アドレスを自動的に検出し、要求の送信後にホストを再起動します。Satellite でホストを再起動できるように、ホストのタッチ先の Capsule から TCP ポート 8443 にアクセスできなければならないことに注意してください。

PXE なしのプロビジョニング

新規ホストの要求の送信後に、Satellite Server からダウンロードして、ホストの USB ポートを使用して転送したブートディスクで特定のホストを起動する必要があります。

コンピュータリソース

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



注記

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

第2章 プロビジョニングリソースの設定

2.1. プロビジョニングコンテキスト

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

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

ロケーションは組織と同様に機能します。相違点は、ロケーションが物理的な設定または地理的な設定をもとにしている点です。ユーザーは階層でロケーションをネスト化できます。ロケーションの詳細は、**Red Hat Satellite の管理**の [ロケーションの管理](#) を参照してください。

2.2. プロビジョニングコンテキストの設定

プロビジョニングコンテキストの設定時に、プロビジョニングホストに使用する組織およびロケーションを定義します。

組織およびロケーションのメニューは、Satellite Web UI の左上のメニューバーにあります。使用する組織またはロケーションを選択していない場合には、メニューには **任意の組織** および **任意のロケーション** と表示されます。

手順

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

各ユーザーはアカウント設定でデフォルトのプロビジョニングコンテキストを設定できます。Satellite Web UI の右上のユーザー名をクリックし、**マイアカウント** を選択してユーザーアカウントの設定を編集します。

CLI 手順

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

```
# hammer host list --organization "My_Organization" --location "My_Location"
```

このコマンドは、**My_Organization** および **My_Location** に割り当てられたホストを出力します。

2.3. オペレーティングシステムの作成

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

Red Hat の CDN からオペレーティングシステムをインポートすると、**ホスト > オペレーティングシステム** ページで新規エントリーが作成されます。Red Hat の CDN からオペレーティングシステムをインポートするには、オペレーティングシステムの Red Hat リポジトリを有効にし、リポジトリを Satellite に同期します。詳細は、[コンテンツの管理](#) の Red Hat リポジトリの有効化 および リポジトリの同期 を参照してください。

以下の手順を使用して、カスタムのオペレーティングシステムも追加できます。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

手順

1. Satellite Web UI で **ホスト > オペレーティングシステム** に移動して、**新規オペレーティングシステム** をクリックします。
2. **名前** フィールドには、オペレーティングシステムエントリーの名前を入力します。
3. **メジャー** フィールドには、オペレーティングシステムのメジャーバージョンに対応する数字を入力します。
4. **マイナー** フィールドには、オペレーティングシステムのマイナーバージョンに対応する数字を入力します。
5. **説明** フィールドには、オペレーティングシステムの説明を加えます。
6. **ファミリー** リストから、オペレーティングシステムのファミリーを選択します。
7. **root パスワードのハッシュ** リストから、Root パスワードのエンコード方法を選択します。
8. **アーキテクチャー** リストから、オペレーティングシステムがシステムが使用するアーキテクチャーを選択します。
9. **Partition table** タブをクリックして、対象のオペレーティングシステムに適用するパーティションテーブルの候補を選択します。
10. オプション: Red Hat 以外のコンテンツを使用する場合には、**Installation Media** タブをクリックして、対象のオペレーティングシステムに適用するインストールメディアを選択します。詳細は、[Satellite へのインストールメディアの追加](#) を参照してください。
11. **Templates** タブをクリックして、オペレーティングシステムで使用する **PXELinux template**、**Provisioning template** および **Finish template** を選択します。プロビジョニングに iPXE を使用予定の場合には、**iPXE テンプレート** など、他のテンプレートを選択してください。
12. **Submit** をクリックしてプロビジョニングテンプレートを保存します。

CLI 手順

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

```
# hammer os create \  
--architectures "x86_64" \  
--description "My_Custom_Operating_System" \  
--family "Redhat" \  
--major 7 \  
--media "Red_Hat" \  
--minor 3 \  

```

```
--name "My_Operating_System" \  
--partition-tables "My_Partition" \  
--provisioning-templates "My_Provisioning_Template"
```

2.4. 複数のオペレーティングシステムの詳細更新

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

手順

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

```
PARTID=$(hammer --csv partition-table list | grep "Kickstart default," | cut -d, -f1)  
PXEID=$(hammer --csv template list --per-page=1000 | grep "Kickstart default PXELinux" |  
cut -d, -f1)  
SATID=$(hammer --csv template list --per-page=1000 | grep "provision" | grep ",Kickstart  
default" | cut -d, -f1)  
  
for i in $(hammer --no-headers --csv os list | awk -F, {'print $1'})  
do  
  hammer partition-table add-operatingsystem --id="{PARTID}" --operatingsystem-id="{i}"  
  hammer template add-operatingsystem --id="{PXEID}" --operatingsystem-id="{i}"  
  hammer os set-default-template --id="{i}" --config-template-id={PXEID}  
  hammer os add-config-template --id="{i}" --config-template-id={SATID}  
  hammer os set-default-template --id="{i}" --config-template-id={SATID}  
done
```

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

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

2.5. アーキテクチャーの作成

Satellite 内のアーキテクチャーはホストおよびオペレーティングシステムの論理グループを表します。アーキテクチャーは、ホストが Puppet に接続する際に Satellite によって自動的に作成されます。Satellite には、基本的な i386 と x86_64 のアーキテクチャーが事前設定されています。

以下の手順を使用して Satellite のアーキテクチャーを作成します。

サポートされるアーキテクチャー

PXE、Discovery およびブートディスクを使用したプロビジョニングをサポートするのは Intel x86_64 アーキテクチャーのみです。詳細は、Red Hat ナレッジベースソリューション [Supported architectures and provisioning scenarios in Satellite 6](#) を参照してください。

手順

1. Satellite Web UI で **ホスト > アーキテクチャー** に移動して、**アーキテクチャーの作成** をクリックします。

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

CLI手順

- **hammer architecture create** コマンドを入力して、アーキテクチャーを作成します。このアーキテクチャーに含める名前とオペレーティングシステムを指定します。

```
# hammer architecture create \  
--name "My_Architecture" \  
--operatingsystems "My_Operating_System"
```

2.6. ハードウェアモデルの作成

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

手順

1. Satellite Web UI で **ホスト > ハードウェアモデル** に移動して、**モデルの作成** をクリックします。
2. **名前** フィールドには、ハードウェアモデルの名前を入力します。
3. 必要に応じて、**Hardware Model** フィールドと **Vendor Class** フィールドに、該当するシステムの情報を入力できます。
4. **Info** フィールドに、ハードウェアモデルの説明を入力します。
5. **Submit** をクリックしてハードウェアモデルを保存します。

CLI手順

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

```
# hammer model create \  
--hardware-model "My_Hardware_Model" \  
--info "My_Description" \  
--name "My_Hardware_Model_Name" \  
--vendor-class "My_Vendor_Class"
```

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

Satellite には、同期済みのキックスタートリポジトリのセットが含まれており、これは、プロビジョニングしたホストのオペレーティングシステムをインストールする時に使用します。リポジトリの追加の詳細は、[コンテンツの管理](#) の [リポジトリの同期](#) を参照してください。

この手順を使用して、キックスタートリポジトリを設定します。

前提条件

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

手順

1. 既存のコンテンツビューに、使用を希望する同期済みのキックスタートリポジトリを追加するか、コンテンツビューを新たに作成してキックスタートリポジトリを追加します。
Red Hat Enterprise Linux 8 の場合は、**Red Hat Enterprise Linux 8 for x86_64 - AppStream Kickstart x86_64 8** と **Red Hat Enterprise Linux 8 for x86_64 - BaseOS Kickstart x86_64 8** の両方のリポジトリが追加されていることを確認します。

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

2. キックスタートリポジトリを追加した新しいバージョンのコンテンツビューを公開し、必要なライフサイクル環境にプロモートします。詳細は、[コンテンツの管理](#) の [コンテンツビューの管理](#) を参照してください。
3. ホストの作成時に、**オペレーティングシステム** タブの **メディアの選択** で、**同期済みのコンテンツ** のチェックボックスを選択します。

キックスタートツリーを表示するには、以下のコマンドを入力します。

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

2.8. SATELLITE へのインストールメディアの追加

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

インストールメディアを表示するには、**Hosts > Provisioning Setup > Installation Media** に移動します。

インストールメディアは、オペレーティングシステムのインストールツリーの形式で提供され、インストーラーをホストするマシンから HTTP URL 経由でアクセスできる必要があります。

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

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

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

手順

1. Satellite Web UI で **ホスト > インストールメディア** に移動して、**メディアの作成** をクリックします。
2. **Name** フィールドには、インストールメディアエントリーの名前を入力します。
3. **Path** には、インストールツリーを含む URL を入力します。複数の異なるシステムアーキテクチャーおよびバージョンを表すために以下の変数をパスで使用できます。
 - **\$arch**:-: システムアーキテクチャー
 - **\$version**:-: オペレーティングシステムのバージョン
 - **\$major**:-: オペレーティングシステムのメジャーバージョン
 - **\$minor**:-: オペレーティングシステムのマイナーバージョン
 HTTP パスの例:

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

CLI 手順

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

```
# hammer medium create \
--locations "My_Location" \
--name "My_OS" \
--organizations "My_Organization" \
--os-family "Redhat" \
--path "http://download.example.com/centos/$version/Server/$arch/os/"
```

2.9. パーティションテーブルの作成

パーティションテーブルは、テンプレート的一种で、Satellite Server が新規ホストで利用可能なディスクを設定する方法を定義します。パーティションテーブルは、プロビジョニングテンプレートと同じ ERB 構文を使用します。Red Hat Satellite には、**Kickstart default** などの、デフォルトのパーティションテーブルのセットが含まれます。また、パーティションテーブルのエントリーを編集して、任意のパーティションスキームの設定やパーティションテーブルのエントリー作成を行い、そのエントリーをオペレーティングシステムのエントリーに追加することができます。

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

手順

1. Satellite Web UI で **ホスト > パーティションテーブル** に移動して、**パーティションテーブルの作成** をクリックします。
2. **Name** フィールドにパーティションテーブルの名前を入力します。
3. テンプレートを新しい組織またはロケーションに自動的に関連付けるように設定する場合は、**Default** チェックボックスをオンにします。
4. テンプレートを他のパーティションテーブルで再利用可能なスニペットに指定する場合は、**Snippet** チェックボックスをオンにします。
5. **Operating System Family** リストから、パーティションレイアウトのディストリビューションまたはファミリーを選択します。たとえば、Red Hat Enterprise Linux、CentOS、および Fedora は、Red Hat ファミリーに属します。
6. **Template editor** フィールドに、ディスクパーティションのレイアウトを入力します。レイアウトのフォーマットは、オペレーティングシステムのフォーマットと一致する必要があります。たとえば、Red Hat Enterprise Linux には、以下のような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。

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

詳細は、「[動的パーティションの例](#)」を参照してください。

テンプレートエディターでファイルブラウザーを使用して、ファイルからレイアウトをインポートすることもできます。

7. **Audit Comment** フィールドに、パーティションレイアウトの変更の概要を追加します。
8. **Organizations** タブと **Locations** タブをクリックして、パーティションテーブルに関連付ける他のプロビジョニングコンテキストを追加します。Satellite により、現在のプロビジョニングコンテキストに、そのパーティションテーブルが追加されます。
9. **Submit** をクリックしてパーティションテーブルを保存します。

CLI 手順

1. パーティションレイアウトを含む `~/My_Partition_Table` などのプレーンテキストファイルを作成します。レイアウトのフォーマットは、オペレーティングシステムのフォーマットと一致する必要があります。たとえば、Red Hat Enterprise Linux には、以下のような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。

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

詳細は、「[動的パーティションの例](#)」を参照してください。

2. **hammer partition-table create** コマンドを使用してインストールメディアを作成します。

```
# hammer partition-table create \
--file "~/My_Partition_Table" \
--locations "My_Location" \
```

```
--name "My Partition Table" \  
--organizations "My_Organization" \  
--os-family "My_OS_Family" \  
--snippet false
```

2.10. 動的パーティションの例

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

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

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

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

```
#Dynamic (do not remove this line)  
  
MEMORY=$((`grep MemTotal: /proc/meminfo | sed 's/^MemTotal: */'|sed 's/ .*' / 1024))  
if [ "$MEMORY" -lt 2048 ]; then  
    SWAP_MEMORY=$((MEMORY * 2))  
elif [ "$MEMORY" -lt 8192 ]; then  
    SWAP_MEMORY=MEMORY  
elif [ "$MEMORY" -lt 65536 ]; then  
    SWAP_MEMORY=$((MEMORY / 2))  
else  
    SWAP_MEMORY=32768  
fi  
  
cat <<EOF > /tmp/diskpart.cfg  
zerombr yes  
clearpart --all --initlabel  
part /boot --fstype ext4 --size 200 --asprimary  
part swap --size "$SWAP_MEMORY"  
part / --fstype ext4 --size 1024 --grow  
EOF
```

2.11. プロビジョニングテンプレート

プロビジョニングテンプレートは、Satellite Server がホストにオペレーティングシステムをインストールする方法を定義します。

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

Red Hat でサポートされているテンプレートは、Red Hat アイコンで示されます。

サポートされていないテンプレートを非表示にするには、Satellite Web UI で **Administer > Settings** に移動します。 **Provisioning** タブで、 **Show unsupported provisioning templates** の値を **false** に設定し、 **Submit** をクリックします。クエリー `supported = true` を作成して、サポートされているテンプレートを除外することもできます。

サポートされているテンプレートを複製すると、複製されたテンプレートはサポートされなくなります。

Embedded Ruby (ERB) 構文を受け入れるテンプレート。詳細は、 **ホストの管理** の [テンプレート作成リファレンス](#) を参照してください。

プロビジョニングテンプレートはダウンロードが可能です。ただし、テンプレートのダウンロード前にデバッグ証明書を作成する必要があります。詳細は、 **Red Hat Satellite の管理** の [組織のデバッグ証明書の作成](#) を参照してください。

Satellite Server と Git リポジトリまたはローカルディレクトリの間でテンプレートを同期できます。詳細は、 **ホストの管理** の [テンプレートリポジトリの同期](#) を参照してください。

テンプレートの変更履歴を表示するには、 **ホスト > プロビジョニングテンプレート** に移動してテンプレートを選択し、 **履歴** をクリックします。 **戻す** をクリックすると、以前のバージョンでコンテンツを上書きできます。さらに前の変更に戻すこともできます。特定の変更に関する情報を表示するには、 **Show Diff** をクリックします。

- **Template Diff** タブには、プロビジョニングテンプレートの本文の変更が表示されます。
- **Details** タブには、テンプレートの説明の変更が表示されます。
- **History** タブには、テンプレートに変更を加えたユーザーと変更日が表示されます。

2.12. プロビジョニングテンプレートのタイプ

プロビジョニングテンプレートにはさまざまな種類があります。

プロビジョニング

プロビジョニングプロセスのテンプレート（例：キックスタートテンプレート）。キックスタートの構文およびコマンドの詳細は、以下の資料を参照してください。

- [高度な RHEL 9 インストールの実行のキックスタートの参照](#)
- [高度な RHEL 8 インストールの実行のキックスタートの参照](#)
- [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](#)

PXELinux、PXEGrub、PXEGrub2

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

終了

主なプロビジョニングプロセスの完了時に、SSH 接続を使用して実行するプロビジョニング後の設定スクリプト。 **finish** テンプレートは、 `user_data` をサポートしない仮想またはクラウド環境でのイメージベースのプロビジョニングにのみ使用できます。Foreman Discovery イメージと呼ばれる場

合もあるので、Foreman Discovery ISO のイメージと混同しないようにしてください。このコンテキストのイメージは、デプロイメントの簡素化を目指した、仮想化環境でのインストールイメージを指します。

Red Hat Satellite は、終了スクリプトがリターンコード **0** で正常に終了すると、このコードが成功したとして処理し、ホストはビルドモードを終了します。

終了スクリプトには、**call back** HTTP 呼び出しを使用するビルドモードを含むものが複数ある点に注意してください。上記のスクリプトは、イメージベースのプロビジョニングには使用されず、Debian、Ubuntu、BSD などのオペレーティングシステムをインストールした後の設定に使用されます。Red Hat は、Red Hat Enterprise Linux 以外のオペレーティングシステムのプロビジョニングをサポートしていません。

user_data

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

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

cloud_init

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

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

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

Bootdisk

PXE 以外の起動方法に使用するテンプレート

カーネル実行 (kexec)

PXE 以外の起動方法に使用するカーネル実行テンプレート



注記

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

スクリプト

デフォルトで使用されないが、カスタムタスクに役立つ任意のスクリプト。

ZTP

Zero Touch Provisioning テンプレート。

POAP

PowerOn Auto Provisioning テンプレート。

iPXE

PXELinux の代わりに **iPXE** または **gPXE** 環境で使用するテンプレート。

2.13. プロビジョニングテンプレートの作成

プロビジョニングテンプレートは、Satellite Server がホストにオペレーティングシステムをインストールする方法を定義します。この手順を使用して、新規のプロビジョニングテンプレートを作成します。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動し、**Create Template** をクリックします。
2. **名前** フィールドには、プロビジョニングテンプレートの名前を入力します。
3. 必要に応じて残りのフィールドに入力します。**Help** タブに、テンプレートの構文に関する情報と、テンプレート内のさまざまな種類のオブジェクトで呼び出すことができる使用可能な関数、変数、メソッドの詳細が表示されます。

CLI 手順

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

```
# hammer template create \  
--file ~/my-template \  
--locations "My_Location" \  
--name "My_Provisioning_Template" \  
--organizations "My_Organization" \  
--type provision
```

2.14. プロビジョニングテンプレートのクローン作成

プロビジョニングテンプレートは、Satellite Server がホストにオペレーティングシステムをインストールする方法を定義します。この手順を使用して、テンプレートのクローンを作成して、更新をこのクローンに追加します。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動し、使用するテンプレートを検索します。
2. **クローン** をクリックして、テンプレートを複製します。

3. **Name** フィールドに、プロビジョニングテンプレートの名前を入力します。
4. テンプレートを新規の組織またはロケーションに自動的に関連付けられるように設定するには、**デフォルト** のチェックボックスを選択します。
5. **テンプレートエディター** フィールドには、プロビジョニングテンプレートのボディを入力します。**Template** ファイルブラウザを使用してテンプレートファイルをアップロードすることもできます。
6. **監査コメント** には、監査を目的とするプロビジョニングテンプレートへの変更についての概要を入力します。
7. **タイプ** タブをクリックして、テンプレートがスニペットの場合には、**スニペット** のチェックボックスを選択します。スニペットは、スタンドアロンのプロビジョニングテンプレートではありませんが、他のプロビジョニングテンプレートに挿入可能なプロビジョニングテンプレートに含まれています。
8. **タイプ** のリストから、テンプレートのタイプを選択します。(例: **プロビジョニングテンプレート**)
9. **関連付け** タブをクリックして、**適用可能なオペレーティングシステム** リストから、このプロビジョニングテンプレートに関連付けるオペレーティングシステム名を選択します。
10. またオプションとして、**組み合わせの追加** をクリックして **ホストグループ** の一覧からホストグループを1つ選択するか、**環境** の一覧から環境を1つ選択すると、指定したホストグループと環境と、プロビジョニングテンプレートに関連付けることができます。
11. **組織** および **ロケーション** タブをクリックして、テンプレートに別のコンテキストを追加します。
12. **Submit** をクリックしてプロビジョニングテンプレートを保存します。

2.15. カスタムプロビジョニングスニペットの作成

カスタムプロビジョニングスニペットを使用すると、ホストのプロビジョニング中にカスタムコードを実行できます。プロビジョニングプロセスの前後にコードを実行できます。

前提条件

- プロビジョニングテンプレートによっては、カスタムプロビジョニングスニペットを含めるために使用できる複数のカスタムスニペットフックが存在します。最初にプロビジョニングテンプレートをチェックして、使用できるカスタムスニペットを確認してください。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動し、**Create Template** をクリックします。
2. **Name** フィールドに、カスタムプロビジョニングスニペットの名前を入力します。名前は、カスタムプロビジョニングスニペットを含めることをサポートするプロビジョニングテンプレートの名前で始まる必要があります。
 - ホストをプロビジョニングする前にコードを実行するには、プロビジョニングテンプレートの名前に「custom pre」を追加します。

- ホストのプロビジョニング後にコードを実行するには、プロビジョニングテンプレートの名前に「custom post」を追加します。
3. **Type** タブで、**Snippet** を選択します。
 4. **Submit** をクリックして、カスタムプロビジョニングスニペットを作成します。

CLI手順

1. CLIでテンプレートを作成する前に、カスタムスニペットを含むプレーンテキストファイルを作成します。
2. **hammer** を使用してテンプレートを作成します。

```
# hammer template create \
--file "/path/to/My_Snippet" \
--locations "My_Location" \
--name "My_Template_Name_custom_pre" --organizations "_My_Organization" \
--type snippet
```

2.16. RED HAT ENTERPRISE LINUX のカスタムプロビジョニングスニペットの例

Custom Post スニペットを使用して、ホストをプロビジョニングした直後に、プロビジョニングテンプレート内から外部 API を呼び出すことができます。

Kickstart default finish custom post Red Hat Enterprise Linux の例

```
echo "Calling API to report successful host deployment"

yum install -y curl ca-certificates

curl -X POST \
-H "Content-Type: application/json" \
-d '{"name": "<%= @host.name %>", "operating_system": "<%= @host.operatingsystem.name %>",
"status": "provisioned",}' \
"https://api.example.com/"
```

2.17. テンプレートとオペレーティングシステムの関連付け

テンプレートを Satellite のオペレーティングシステムに関連付けることができます。次の例では、プロビジョニングテンプレートをオペレーティングシステムエントリーに追加します。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
2. プロビジョニングテンプレートを選択します。
3. **Association** タブで、該当するすべてのオペレーティングシステムを選択します。
4. **Submit** をクリックして変更を保存します。

CLI手順

1. オプション: すべてのテンプレートを表示します。

```
# hammer template list
```

2. オプション: すべてのオペレーティングシステムを表示します。

```
# hammer os list
```

3. テンプレートをオペレーティングシステムに関連付けます。

```
# hammer template add-operatingsystem \  
--id My_Template_ID \  
--operatingsystem-id My_Operating_System_ID
```

2.18. コンピュートプロファイルの作成

コンピュートプロファイルを使用して、CPU、メモリー、ストレージなどの仮想マシンハードウェアの詳細を事前定義できます。

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

Red Hat Satellite のデフォルトのインストールには、3つの定義済みプロファイルが含まれています。

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

コンピュートプロファイルは、サポートされているすべてのコンピュートリソースに適用できます。

- 「サポートされているクラウドプロバイダー」
- 「サポートされている仮想化インフラストラクチャー」

手順

1. Satellite Web UI で **インフラストラクチャー > コンピュートプロファイル** に移動して、**コンピュートプロファイルの作成** をクリックします。
2. **Name** フィールドには、プロファイル名を入力します。
3. **Submit** をクリックします。新しいウィンドウが開き、コンピューティングプロファイルの名前が表示されます。
4. 新しいウィンドウで、各コンピュートリソースの名前をクリックし、このコンピューティングプロファイルに設定する属性を編集します。

CLI手順

1. 新しいコンピュートプロファイルを作成します。

```
# hammer compute-profile create --name "My_Compute_Profile"
```

- 2. コンピュートプロファイルの属性を設定します。

```
# hammer compute-profile values create \
--compute-attributes "flavor=m1.small,cpus=2,memory=4GB,cpu_mode=default" \
--compute-resource "My_Compute_Resource" \
--compute-profile "My_Compute_Profile" \
--volume size=40GB
```

- 3. オプション: コンピュートプロファイルの属性を更新するには、変更する属性を指定します。たとえば、CPU の数とメモリーサイズを変更するには、次のようにします。

```
# hammer compute-profile values update \
--compute-resource "My_Compute_Resource" \
--compute-profile "My_Compute_Profile" \
--attributes "cpus=2,memory=4GB" \
--interface "type=network,bridge=br1,index=1" \
--volume "size=40GB"
```

- 4. オプション: コンピュートプロファイルの名前を変更するには、**--new-name** 属性を使用します。

```
# hammer compute-profile update \
--name "My_Compute_Profile" \
--new-name "My_New_Compute_Profile"
```

関連情報

- Hammer を使用してコンピューティングプロファイルを作成する方法の詳細は、**hammer compute-profile --help** と入力してください。

2.19. ホストへのデフォルト暗号化 ROOT パスワードの設定

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

デフォルトの root パスワードは、ホストグループによって継承され、結果的にそのグループ内のホストによって継承される可能性があります。

パスワードを変更し、パスワードを継承するグループ内のホストを再プロビジョニングすると、ホストでパスワードが上書きされます。

手順

1. 暗号化されたパスワードを生成します。

```
$ python3 -c 'import crypt;pw=getpass.getpass(); print(crypt.crypt(pw)) if (pw==getpass.getpass("Confirm: ")) else exit()'
```

2. 後で使用するために、パスワードをコピーしておきます。
3. Satellite Web UI で、**Administer > Settings** に移動します。
4. **設定** ページで、**プロビジョニング** タブを選択します。

5. **Name** コラムで **Root** パスワードを探して、**クリックして編集** をクリックします。
6. 暗号化パスワードを貼り付け、**保存** をクリックします。

2.20. NOVNC を使用した仮想マシンへのアクセス

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

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

- VMware
- Libvirt
- Red Hat Virtualization

前提条件

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

手順

1. 管理対象のホストで、ポート 5900 から 5930 で VNC サービスを許可するようにファイアウォールを設定します。

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

2. Satellite Web UI で、**Infrastructure > Compute Resources** に移動し、コンピュートリソースの名前を選択します。
3. **Virtual Machines** タブで、仮想マシンの名前を選択します。マシンの電源がオンになっていることを確認してから、**Console** を選択します。

第3章 ネットワークの設定

それぞれのプロビジョニングタイプにはネットワーク設定が必要です。本章を参照して、Satellite Server 上に統合された Capsule でネットワークサービスを設定します。

新規ホストは、Capsule Server にアクセスできる必要があります。Capsule Server は、Satellite Server 上に統合された Capsule、または外部 Capsule Server のいずれかになります。ホストが分離したネットワーク上にあり、Satellite Server に直接接続できない場合、またはコンテンツと Capsule Server を同期している場合には、外部 Capsule Server からのホストのプロビジョニングを推奨します。また、外部 Capsule Server を使用したプロビジョニングは、ネットワークの帯域幅を節約できます。

Capsule Server の設定には、基本的な要件が 2 つあります。

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

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

3.1. ファクトと NIC フィルタリング

ファクトは、ホストによって報告された合計メモリー、オペレーティングシステムのバージョン、またはアーキテクチャーなどに関する情報を説明します。[監視 > ファクト](#) で、ファクトを検索し、ファクトを使用してホストを検索したり、テンプレートでファクトを使用したりできます。

Satellite は複数のソースからファクトを収集します:

- **subscription manager**
- **ansible**
- **puppet**

Satellite は、ホストとネットワークインターフェイスのインベントリシステムです。ハイパーバイザーまたはコンテナホストの場合には、ホストごとに数千のインターフェイスを追加したり、数分ごとにインベントリを更新したりするのは不十分です。報告された個々の NIC ごとに、Satellite は NIC エントリを作成し、それらのエントリはデータベースから削除されません。すべてのファクトを解析し、データベース内のすべてのレコードを比較すると、Satellite は非常に遅くなり、使用できなくなります。さまざまなアクションのパフォーマンスを最適化する場合に、最も重要なのはファクトインポートです。[管理 > 設定](#) の [ファクト](#) タブのオプションを使用できます。

3.2. データベースから NIC を削除することによるパフォーマンスの最適化

Satellite に保存されたファクトのパターンを除外すると、識別子が一致するインターフェイスを無視するオプションを使用して、接続をフィルタリングして除外します。デフォルトでは、これらのオプションは最も一般的なハイパーバイザーに設定されています。仮想インターフェイスに別の名前を付け

る場合は、このフィルターを更新し、要件に合わせて使用できます。

手順

1. Satellite Web UI で、**Administer** > **Settings** に移動し、**Facts** タブを選択します。
2. **blu** などの特定の名前で始まるすべてのインターフェイスを除外するには、**blu*** を **識別子が一致するインターフェイスを無視する** オプションに追加します。
3. **blu** などの特定の名前で始まるインターフェイスに関連するファクトをデータベースが保存されないようにするには、**blu*** を **Satellite** に保存されたファクトのパターンを除外する オプションに追加します。
デフォルトでは、**識別子が一致するインターフェイスを無視する** オプションと同じリストが含まれています。要件に基づいて上書きできます。これにより、ファクトは保存せずに完全に除外されます。
4. データベースからファクトを削除するには、次のコマンドを入力します。

```
# foreman-rake facts:clean
```

このコマンドは、**管理** > **設定** > **ファクト** > **Satellite** に保存されたファクトのパターンを除外する オプションに追加されたフィルターに一致するすべてのファクトを削除します。

5. データベースからインターフェイスを削除するには、次のコマンドを入力します。

```
# foreman-rake interfaces:clean
```

このコマンドは、**管理** > **設定** > **ファクト** > **識別子が一致するインターフェイスを無視する** オプションで追加されたフィルターと一致するすべてのインターフェイスを削除します。

3.3. ネットワークリソース

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

ドメイン

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

サブネット

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

- **DHCP**: DHCP Capsule は、範囲の最初のアドレスから開始し、予約されているアドレスをすべて飛ばして、次に利用可能な IP アドレスを検索することで、IP アドレスの割り当てを管理します。Capsule は IP アドレスを割り当てる前に、ICMP と TCP に Ping を送信して、IP アドレスが使用中かどうかを確認します。ホストの電源がオフの場合または、ファイアウォールの設定で接続が無効な場合には、Satellite は IP アドレスが利用可能であると誤っ

て判断します。このチェックは、電源がオフになっているホストでは機能しないので、DHCP オプションは Satellite が制御し、外部で作成されたホストが含まれていないサブネットでのみ利用できます。

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

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

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

DHCP の範囲

Satellite Server では、検出されたシステムおよびプロビジョニングシステムの両方に同じ DHCP 範囲を定義することはできませんが、各サービスに同じサブネット内の別の範囲を使用することを推奨します。

3.4. SATELLITE および DHCP オプション

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

next-server オプション

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

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

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

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

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

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

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

filename オプション

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

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

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

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

同期していない DHCP レコード

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

手順

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

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

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

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

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

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

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

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

PXE ローダーオプションの変更

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

DHCP ファイルでの不正なパーミッション

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

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

DHCP Capsule エントリーの変更

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

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

dhcpd.leases ファイルで削除済みのホストエントリー

DHCP リースに対する変更は、**dhcpd.leases** ファイルの最後に追加されます。エントリーがこのファイルに追加されるので、**dhcpd.leases** ファイルに、同時に同じリースのエントリーが2つ以上存在する可能性があります。同じリースのエントリーが2つ以上ある場合には、ファイルの最後のエントリーが優先されます。リースファイルのグループ、サブグループ、ホストの宣言も、同じ方法で処理されます。リースが削除されると、`{ deleted; }` が宣言に追加されます。

3.6. イメージベースのプロビジョニングの要件

ブート後の設定方法

finish ブート後設定スクリプトを使用するイメージは、Satellite の統合 Capsule または外部 Capsule など、管理された DHCP サーバーが必要です。ホストは DHCP Capsule と関連付けられたサブネットで作成する必要があり、ホストの IP アドレスは、DHCP 範囲の有効な IP アドレスでなければなりません。

外部の DHCP サービスを使用することは可能ですが、IP アドレスは手動で入力する必要があります。イメージの設定に対応する SSH 認証情報は、ブート後の設定を実行できるように Satellite に設定しなければなりません。

設定後スクリプトに依存するイメージからブートした仮想マシンをトラブルシューティングする場合には、以下の項目を確認してください。

- ホストには、Satellite Server に割り当てられたサブネットがあること。
- サブネットには、Satellite Server に割り当てられた DHCP Capsule があること。
- ホストには、Satellite Server に割り当てられた有効な IP アドレスがあること。
- DHCP を使用した仮想マシンが取得した IP アドレスは、Satellite Server に設定されたアドレスと一致すること。
- イメージから作成された仮想マシンは、SSH リクエストに応答すること。
- イメージから作成された仮想マシンは、SSH を介して、デプロイされたイメージと関連付けられている、ユーザーとパスワードを承認すること。
- Satellite Server で SSH キーを使用して仮想マシンにアクセスできること。これは、仮想マシンが Satellite Server から設定後のスクリプトを受信するために必要です。

ブート前の初期化の設定方法

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

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

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

仮想マシンイメージの **finish** および **cloud-init** スクリプトに対するさまざまなサポートレベルに関する詳細は、Red Hat カスタマーポータル [の Red Hat ナレッジベースソリューション **What are the supported compute resources for the finish and cloud-init scripts**](#) を参照してください。

3.7. ネットワークサービスの設定

一部のプロビジョニング方法では Capsule Server サービスを使用します。たとえば、ネットワークで Capsule Server を DHCP サーバーとして機能させる必要がある場合があります。また、ネットワークで PXE ブートサービスを使用して、新規ホストにオペレーティングシステムをインストールすることも可能です。この場合には、主な PXE ブートサービスである DHCP、DNS および TFTP を使用できるように Capsule Server を設定する必要があります。

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

外部の Capsule Server にこれらのサービスを設定するには、**satellite-installer --scenario capsule** を実行します。

手順

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

```
# satellite-installer --foreman-proxy-dhcp true \
--foreman-proxy-dhcp-gateway "192.168.140.1" \
--foreman-proxy-dhcp-managed true \
--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; 8.8.4.4" \
--foreman-proxy-dns-managed true \
--foreman-proxy-dns-reverse "140.168.192.in-addr.arpa" \
--foreman-proxy-dns-server "127.0.0.1" \
--foreman-proxy-dns-zone "example.com" \
--foreman-proxy-tftp true \
--foreman-proxy-tftp-managed true
```

2. 設定する Capsule Server を検索します。

```
# hammer capsule list
```

3. Capsule Server の機能を更新して変更を表示します。

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

4. Capsule Server に設定されたサービスを確認します。

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

3.7.1. インストーラーを使用した複数のサブネットまたはドメイン

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

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

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

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

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

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

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

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

3.7.2. ネットワーク設定の DHCP オプション

`--foreman-proxy-dhcp`

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

`--foreman-proxy-dhcp-managed`

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

`--foreman-proxy-dhcp-gateway`

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

`--foreman-proxy-dhcp-interface`

要求をリッスンするために DHCP サービスのインターフェイスを設定します。これは、**eth1** に設定します。

`--foreman-proxy-dhcp-nameservers`

DHCP でクライアントに提供されたネームサーバーのアドレスを設定します。これは、**eth1** の Satellite Server のアドレスに設定します。

`--foreman-proxy-dhcp-range`

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

--foreman-proxy-dhcp-server

管理する DHCP サーバーのアドレスを設定します。

satellite-installer --help を実行して、DHCP およびその他の Capsule サービスに関連するその他のオプションを表示します。

3.7.3. ネットワーク設定の DNS オプション

--foreman-proxy-dns

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

--foreman-proxy-dns-provider

使用するプロバイダーを選択します。

--foreman-proxy-dns-managed

インストーラーに ISC BIND を管理させます。これは、**nsupdate** および **nsupdate_gss** プロバイダーを使用する場合にのみ関連します。このオプションは、**true** または **false** に設定します。

--foreman-proxy-dns-forwarders

DNS フォワーダーを設定します。ISC BIND がインストーラーによって管理されている場合にのみ使用されます。これを DNS リカーサーに設定します。

--foreman-proxy-dns-interface

DNS 要求をリッスンするためのインターフェイスを設定します。ISC BIND がインストーラーによって管理されている場合にのみ使用されます。これは、**eth1** に設定します。

--foreman-proxy-dns-reverse

DNS 逆引きゾーン名です。ISC BIND がインストーラーによって管理されている場合にのみ使用されます。

--foreman-proxy-dns-server

DNS サーバーのアドレスを設定します。**nsupdate**、**nsupdate_gss**、および **infoblox** プロバイダーでのみ使用されます。

--foreman-proxy-dns-zone

DNS ゾーン名に設定します。ISC BIND がインストーラーによって管理されている場合にのみ使用されます。

satellite-installer --help を実行して、DNS およびその他の Capsule サービスに関連するその他のオプションを表示します。

3.7.4. ネットワーク設定の TFTP オプション

--foreman-proxy-tftp

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

--foreman-proxy-tftp-managed

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

--foreman-proxy-tftp-servername

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

satellite-installer --help を実行して、TFTP およびその他の Capsule サービスに関連するその他のオプションを表示します。

3.7.5. NAT 経由での TFTP サービスの使用

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

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

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

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

Red Hat Enterprise Linux 6 で NAT を使用する場合:

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

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

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

```
IPTABLES_MODULES="ip_conntrack_tftp"
```

3.8. SATELLITE SERVER へのドメインの追加

Satellite Server はネットワーク上の各ホストのドメイン名を定義します。Satellite Server には、ドメイン名を割り当てる Capsule Server とドメインに関する情報が必要です。

既存のドメインの確認

Satellite Server には、Satellite Server のインストールの一環として関連するドメインがすでに作成されている可能性があります。コンテキストを **Any Organization** および **Any Location** に切り替えてから、ドメインの一覧でこれが存在するかどうかを確認します。

DNS サーバー設定の考慮事項

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

- システム全体のリゾルバー (**Administer** > **Settings** > **Query local nameservers** が **true** に設定されている場合)
- ホストに関連付けられたサブネットで定義されているネームサーバー
- ホストに関連付けられたドメイン名から SOA に照会される信頼できる NS レコード

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

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

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

手順

1. Satellite Web UI で、**Infrastructure** > **Domains** に移動して、**Create Domain** をクリックします。
2. **DNS Domain** フィールドに、完全な DNS ドメイン名を入力します。
3. **Fullname** フィールドで、プレーンテキストのドメイン名を入力します。
4. **Parameters** タブでは、任意のドメインレベルのパラメーターを設定し、このドメインに割り当てられたホストに適用します。たとえば、テンプレートで使用するユーザー定義ブール値またはストリングパラメーターなどです。
5. **Add Parameter** をクリックし、**Name** および **Value** フィールドに入力します。
6. **Locations** タブをクリックして、ドメインがある場所を追加します。
7. **Organizations** タブをクリックして、ドメインが属する組織を追加します。
8. **Submit** をクリックして変更を保存します。

CLI 手順

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

```
# hammer domain create \  
--description "My_Domain" \  
--dns-id My_DNS_ID \  
--locations "My_Location" \  
--name "my-domain.tld" \  
--organizations "My_Organization"
```

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

3.9. SATELLITE SERVER へのサブネットの追加

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

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

手順

1. Satellite Web UI で、**Infrastructure > Subnets** に移動して、サブネットウィンドウで **Create Subnet** をクリックします。
2. **Name** フィールドに、サブネットの名前を入力します。
3. **Description** フィールドには、サブネットの説明を入力します。
4. **Network address** フィールドには、サブネットのネットワークアドレスを入力します。
5. **Network prefix** フィールドには、サブネットのネットワーク接頭辞を入力します。
6. **Network mask** フィールドには、サブネットのネットワークマスクを入力します。
7. **Gateway address** フィールドには、サブネットの外部ゲートウェイを入力します。
8. **Primary DNS server** フィールドには、サブネットのプライマリー DNS を入力します。
9. **Secondary DNS server** には、サブネットのセカンダリー DNS を入力します。
10. **IPAM** には、IP アドレス管理 (IPAM) に使用するメソッドを選択します。IPAM の詳細は、[3 章 ネットワークの設定](#) を参照してください。
11. 選択した IPAM メソッドの情報を入力します。**Remote Execution** タブをクリックして、リモート実行を制御する Capsule を選択します。
12. **Domains** タブをクリックして、このサブネットに適用するドメインを選択します。
13. **Capsule** タブをクリックして、DHCP、TFTP、および逆引き DNS サービスなど、サブネットの各サービスに適用する Capsule を選択します。
14. **Parameters** タブをクリックして、任意のサブネットレベルのパラメーターを設定し、このサブネットに割り当てられたホストに適用します。たとえば、テンプレートで使用するユーザー定義ブール値またはストリングパラメーターなどです。
15. **Locations** タブをクリックして、この Capsule を使用するロケーションを選択します。
16. **Organizations** タブをクリックして、この Capsule を使用する組織を選択します。
17. **Submit** をクリックしてサブネットの情報を保存します。

CLI 手順

- 以下のコマンドでサブネットを作成します。

```
# hammer subnet create \  
--boot-mode "DHCP" \  
--description "My_Description" \  
--dhcp-id My_DHCP_ID \  
--dns-id My_DNS_ID \  
--dns-primary "192.168.140.2" \  
--dns-secondary "8.8.8.8" \  
--domains "my-domain.tld" --from "192.168.140.111" --gateway "192.168.140.1" --  
ipam "DHCP" --locations "_My_Location" \  
--mask "255.255.255.0" \  
--name "My_Network" \  
--network "192.168.140.0" \  

```

```
--organizations "My_Organization"  
--tftp-id My_TFTP_ID \  
--to "192.168.140.250" \  

```



注記

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

第4章 INFOBLOX を DHCP および DNS プロバイダーとして使用する

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

Infoblox バージョンは、NIOS 8.0 以降がサポートされています。

4.1. INFOBLOX の制限

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

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

Infoblox の TFTP 機能を使用した PXE 関連のファイルのホストはサポートされません。PXE プロビジョニングには、TFTP サービスとして Capsule を使用する必要があります。詳細は、[3章 ネットワークの設定](#)を参照してください。

Satellite IPAM 機能は Infoblox と統合できません。

4.2. INFOBLOX の前提条件

- DHCP と DNS エントリを Satellite で管理するには、Infoblox アカウントの認証情報が必要です。
- **DHCP Admin** および **DNS Admin** という名前で、Infoblox の管理ロールを設定されていることを確認してください。
- 管理ロールは、パーミッションを割り当てるか、対象アカウントによる Infoblox API でタスク実行を許可する管理グループに所属している必要があります。

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

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

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

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

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

CA 証明書をテストするには、CURL クエリーを使用します。

```
# curl -u admin:password https://infoblox.example.com/wapi/v2.0/network
```

証明書に問題がない場合の応答例:

```
[
  {
    "_ref":
    "network/ZG5zLm5ldHdvcmskMTkyLjE2OC4yMDluMC8yNC8w:infoblox.example.com/24/default",
    "network": "192.168.202.0/24",
    "network_view": "default"
  }
]
```

4.4. DHCP INFOBLOX モジュールのインストール

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

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

DHCP Infoblox のレコードタイプに関する考慮事項

DHCP モジュールと DNS Infoblox モジュールと一緒に使用する場合は、**ixedaddress** レコードタイプのみを使用して DHCP Infoblox モジュールを設定します。**host** レコードタイプは DNS 競合を引き起こすため、サポートされていません。

DHCP Infoblox モジュールを **host** レコードタイプで設定する場合は、Infoblox が独自に DNS 管理を行うため、Infoblox が管理するサブネット上で DNS Capsule と Reverse DNS Capsule の両方のオプション設定を解除する必要があります。**host** のレコードタイプを使用すると、競合が発生し、Satellite でホストの名前変更ができなくなります。

手順

1. Capsule で、以下のコマンドを実行します。

```
# satellite-installer --enable-foreman-proxy-plugin-dhcp-infoblox \
--foreman-proxy-dhcp true \
--foreman-proxy-dhcp-provider infoblox \
--foreman-proxy-dhcp-server infoblox.example.com \
--foreman-proxy-plugin-dhcp-infoblox-username admin \
--foreman-proxy-plugin-dhcp-infoblox-password infoblox \
--foreman-proxy-plugin-dhcp-infoblox-record-type fixedaddress \
--foreman-proxy-plugin-dhcp-infoblox-dns-view default \
--foreman-proxy-plugin-dhcp-infoblox-network-view default
```

2. オプション: Satellite Web UI で、**Infrastructure > Capsules** に移動し、DHCP Infoblox モジュールを含む Capsule を選択し、**dhcp** 機能がリストされていることを確認します。
3. Satellite Web UI で、**Infrastructure > Subnets** に移動します。
4. 全サブネットを Infoblox で管理する場合は、サブネットの IP アドレス管理 (IPAM) 方式が **DHCP** に設定されていることを確認してください。

4.5. DNS INFOBLOX モジュールのインストール

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

手順

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

```
# satellite-installer --enable-foreman-proxy-plugin-dns-infoblox \  
--foreman-proxy-dns true \  
--foreman-proxy-dns-provider infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-server infoblox.example.com \  
--foreman-proxy-plugin-dns-infoblox-username admin \  
--foreman-proxy-plugin-dns-infoblox-password infoblox \  
--foreman-proxy-plugin-dns-infoblox-dns-view default
```

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

2. オプション: Satellite Web UI で、**Infrastructure > Capsules** に移動し、Infoblox DNS モジュールを含む Capsule を選択し、**DNS 機能がリストされていることを確認**します。
3. Satellite Web UI で、**Infrastructure > Domains** に移動します。
4. 全ドメインを Infoblox で管理する場合は、それらのドメインに対して **DNS Proxy** が設定されていることを確認してください。
5. Satellite Web UI で、**Infrastructure > Subnets** に移動します。
6. 全サブネットを Infoblox で管理する場合には、**DNS Capsule** および **Reverse DNS Capsule** がそのサブネットに設定されていることを確認します。

第5章 PXE を使用したホストのプロビジョニング

次のいずれかの方法を使用して、Satellite でベアメタルインスタンスをプロビジョニングできます。

無人プロビジョニング

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

Discovery を使用した無人プロビジョニング

新規ホストは PXE ブートを使用して Satellite Discovery サービスをロードします。このサービスはホストのハードウェア情報を特定し、ホストをプロビジョニング可能なホストとしてリスト表示します。詳細は、[7章 Discovery サービスの設定](#) を参照してください。

PXE なしのプロビジョニング

新しいホストには、Satellite Server が生成するブートディスクイメージがプロビジョニングされません。

BIOS および UEFI サポート

Red Hat Satellite では、BIOS および UEFI ベースの PXE プロビジョニングの両方を実行できます。BIOS および UEFI インターフェイスはいずれも、コンピューターのオペレーティングシステムとファームウェアの間のインタープリターとして機能し、ブート時にハードウェアコンポーネントを初期化して、オペレーティングシステムを起動します。

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

Satellite のプロビジョニングでは、PXE ローダーオプションが、プロビジョニング時に使用する DHCP の **filename** オプションを定義します。BIOS システムの場合は、**PXELinux BIOS** オプションを使用してプロビジョニングノードを有効化し、TFTP 経由で **pxelinux.0** ファイルをダウンロードします。UEFI システムの場合は、**PXEGrub2 UEFI** オプションを使用して TFTP クライアントが **grub2/grubx64.efi** ファイルをダウンロードできるようにするか、**PXEGrub2 UEFI HTTP** オプションを使用して UEFI HTTP クライアントが HTTP ブート機能を備え Capsule から **grubx64.efi** をダウンロードできるようにします。

BIOS プロビジョニングでは、PXELinux テンプレートとオペレーティングシステムを関連付ける必要があります。UEFI プロビジョニングでは、PXEGrub2 テンプレートとオペレーティングシステムを関連付ける必要があります。PXELinux と PXEGrub2 のテンプレートの両方を関連付ける場合は、PXE ローダー間で簡単に切り替えができるように、Satellite で、両テンプレートの設定ファイルを TFTP にデプロイできます。

5.1. ベアメタルプロビジョニングの前提条件

ベアメタルプロビジョニングの要件には以下が含まれます。

- ベアメタルホストのネットワークを管理する Capsule Server。無人プロビジョニングおよび Discovery ベースのプロビジョニングの場合に、Satellite Server は PXE サーバーの設定が必要です。ネットワーク要件の詳細は、[3章 ネットワークの設定](#) を参照してください。

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

- ベアメタルホストまたは空の仮想マシン
- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。

- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理](#)の [アクティベーションキーの作成](#) を参照してください。

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

5.2. セキュリティートークンの有効期間の設定

あらゆる種類のプロビジョニングを行う場合には、セキュリティー措置として、Satellite は自動的に一意のトークンを生成し、このトークンを PXE 設定ファイル (PXELinux、Grub2) のキックスタート URL に追加します。デフォルトでは、トークンの有効期限は 360 分です。ホストのプロビジョニング時に、この時間内にホストを再起動するようにしてください。トークンの有効期限が切れると、トークンは無効になり、404 エラーが送出され、オペレーティングシステムのインストーラーのダウンロードが失敗します。

手順

1. Satellite Web UI で、**Administer > Settings** に移動して、**Provisioning** タブをクリックします。
2. **トークンの期間** オプションを検索して、編集アイコンをクリックし、期間を変更するか、**0** と入力してトークンの生成を無効にします。トークンの生成が無効になっている場合には、攻撃者はクライアントの IP アドレスを偽装して、Satellite Server から、暗号化された root パスワードなど、キックスタートをダウンロードできます。

5.3. 無人プロビジョニングによるホストの作成

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

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

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
7. フィールドに値が投入されていることを確認します。特に以下に注意してください。

- Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC address** フィールドには、ホストのプロビジョニングインターフェイスの MAC アドレスを入力します。これにより、PXE ブートプロセス中のホストが識別されます。
 - **ホスト タブの名前は DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
8. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
 9. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。オペレーティングシステムの各要素を確認してください。
 10. オプション: **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
プロビジョニングテンプレートの関連付けの詳細は、「[プロビジョニングテンプレート](#)」を参照してください。
 11. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
 12. **Submit** をクリックしてホストの詳細を保存します。
ネットワークインターフェイスの詳細は、**ホストの管理** の [ネットワークインターフェイスの追加](#) を参照してください。

これで、ホストのエントリーおよび関連するプロビジョニングの設定が作成されます。これには、PXE を使用したベアメタルホストの起動に必要なディレクトリーおよびファイルを作成することも含まれます。物理ホストを起動して、ブートモードを PXE に設定すると、ホストは Satellite Server の統合 Capsule の DHCP サービスを検出し、キックスタートツリーの HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

インストールが完了したら、ホストはアクティベーションキーを使用して Satellite Server にも登録し、必要な設定と管理ツールを Satellite Client 6 のリポジトリーからインストールします。

CLI 手順

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

```
# hammer host create \  
--build true \  
--enabled true \  
--hostgroup "My_Host_Group" \  
--location "My_Location" \  
--mac "My_MAC_Address" \  
--managed true \  
--name "My_Host_Name" \  
--organization "My_Organization"
```

2. **hammer host interface update** コマンドを使用し、ネットワークインターフェイスのオプションが設定されていることを確認します。

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

5.4. PXE を使用しないプロビジョニングによるホストの作成

一部のハードウェアには PXE ブートインターフェイスがありません。Satellite では、PXE ブートなしでホストをプロビジョニングできます。これは、PXE を使用しないプロビジョニングとしても知られ、ホストが使用できるブート ISO が生成されます。この ISO を使用して、ホストは Satellite Server に接続してインストールメディアを起動し、オペレーティングシステムをインストールできます。

Satellite には PXE を使用しない検出サービスがあり、DHCP や TFTP など PXE ベースのサービスなしで操作できます。詳細は、「[PXE を使用しない Discovery の実装](#)」を参照してください。

ブート ISO のタイプ

ブート ISO には次の種類があります。

完全ホストイメージ

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

サブネットイメージ

特定ホストに関連付けられていないブート ISO。ISO はホストの MAC アドレスを Capsule Server に送信し、ホストのエントリーと照合します。イメージには IP アドレスの詳細は保存されず、ブートストラップするにはネットワーク上の DHCP サーバーにアクセスする必要があります。このイメージは、同じサブネットのプロビジョニングした NIC を伴うすべてのホストに対して汎用性があります。イメージは iPXE ブートファームウェアに基づいており、限られた数のネットワークカードのみがサポートされます。



注記

フルホストイメージは SYSLINUX と Grub に基づいており、ほとんどのネットワークカードで動作します。サブネットイメージを使用する場合、iPXE ベースのブートディスクで動作すると予想されるネットワークカードドライバのリストについては、ipxe.org でサポートされているハードウェアを参照してください。

完全なホストイメージにはプロビジョニングトークンが含まれているため、生成されたイメージの有効期間は限られています。セキュリティトークン設定の詳細は、「[セキュリティトークンの有効期間の設定](#)」を参照してください。

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。

4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
7. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC address** フィールドには、ホストのプロビジョニングインターフェイスの MAC アドレスを入力します。これにより、PXE ブートプロセス中のホストが識別されます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
8. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
9. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。オペレーティングシステムの各要素を確認してください。
10. **Provisioning Templates** で **Resolve** をクリックして、新しい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識別できることを確認します。
プロビジョニングテンプレートの関連付けの詳細は、「[プロビジョニングテンプレート](#)」を参照してください。
11. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
12. **Submit** をクリックしてホストの詳細を保存します。これで、ホストのエントリーが作成され、ホストの詳細ページが表示されます。
13. Satellite Server からブートディスクをダウンロードします。
 - ホストの詳細ページの **Full host image** の場合は、縦リーダー記号をクリックして **Full host 'My_Host_Name' イメージ** を選択します。
 - **サブネットイメージ** の場合は、**Infrastructure > Subnets** に移動し、必要なサブネットの **Actions** 列のドロップダウンメニューをクリックして、**Subnet generic image** を選択します。
14. 必要に応じて、**dd** ユーティリティーまたは **livecd-tools** を使用して USB ストレージデバイスに ISO を書き込みます。
15. ホストを起動して、ISO または USB ストレージデバイスからブートする場合には、ホストは Satellite Server に接続し、キックスタートツリーからオペレーティングシステムのインストールを開始します。
インストールが完了したら、ホストはアクティベーションキーを使用して Satellite Server にも登録し、必要な設定と管理ツールを **Satellite Client 6** のリポジトリからインストールします。

CLI 手順

1. **hammer host create** コマンドを使用してホストを作成します。

```
# hammer host create \  
--build true \  
--enabled true \  
--hostgroup "My_Host_Group" \  
--location "My_Location" \  
--mac "My_MAC_Address" \  
--managed true \  
--name "My_Host_Name" \  
--organization "My_Organization"
```

2. **hammer host interface update** コマンドを使用し、お使いのネットワークインターフェイスのオプションが設定されていることを確認します。

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

3. **Hammer bootdisk** コマンドを使用して、Satellite Server からブートディスクをダウンロードします。

- 完全ホストイメージ 向け。

```
# hammer bootdisk host \  
--full true \  
--host My_Host_Name
```

- サブネットイメージ 向け。

```
# hammer bootdisk subnet --subnet My_Subnet_Name
```

これにより、使用するホストのブート ISO が作成されます。

4. 必要に応じて、**dd** ユーティリティーまたは **livecd-tools** を使用して USB ストレージデバイスに ISO を書き込みます。
5. 物理ホストを起動して、ISO または USB ストレージデバイスからブートする場合には、ホストは Satellite Server に接続し、キックスタートツリーからオペレーティングシステムのインストールを開始します。
インストールが完了したら、ホストはアクティベーションキーを使用して Satellite Server にも登録し、必要な設定と管理ツールを **Satellite Client 6** のリポジトリからインストールします。

5.5. UEFI HTTP ブートプロビジョニングを使用したホストの作成

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

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

前提条件

- HTTP 起動要件を満たすようにしてください。詳細は、[Satellite の概要、概念、およびデプロイメントの考慮事項](#) の [HTTP ブートの要件](#) を参照してください。

手順

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

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

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

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

3. Capsule で TFTP および HTTPBoot 機能が認識されていることを確認します。Satellite Web UI で、**Infrastructure > Capsules** に移動し、Capsule をクリックして、認識されている機能のリストを表示します。不足している機能がある場合は、**Refresh Features** をクリックします。
4. Capsule がプロビジョニングサブネットに関連付けられていることを確認します。Satellite Web UI で、**Infrastructure > Subnets > Edit Subnet > Capsules** に移動し、**TFTP** および **HTTPBoot** オプションの両方で Capsule を選択します。
5. **OK** をクリックして保存します。
6. Satellite Web UI で、**Hosts > Create Host** に移動します。
7. **Name** フィールドには、ホストの名前を入力します。
8. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
9. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
10. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
11. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
12. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC address** フィールドには、ホストのプロビジョニングインターフェイスの MAC アドレスを入力します。これにより、PXE ブートプロセス中のホストが識別されます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。

13. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
14. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。オペレーティングシステムの各要素を確認してください。
15. **PXE Loader** のリストから **Grub2 UEFI HTTP** を選択します。
16. オプション: **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
プロビジョニングテンプレートの関連付けの詳細は、「[プロビジョニングテンプレートの作成](#)」を参照してください。
17. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
18. **Submit** をクリックしてホストの詳細を保存します。
ネットワークインターフェイスの詳細は、**ホストの管理** の [ネットワークインターフェイスの追加](#) を参照してください。
19. ネットワークから UEFI モードで起動するようにホストを設定します。
20. ホストを起動します。
21. 起動メニューから、**Kickstart default PXEGrub2** を選択します。

これで、ホストのエントリーおよび関連するプロビジョニングの設定が作成されます。これには、UEFI を使用したベアメタルホストの起動に必要なディレクトリーおよびファイルを作成することも含まれます。物理ホストを起動して、ブートモードを UEFI HTTP に設定すると、ホストは定義済みの DHCP サービスを検出し、キックスターツリーで Capsule の HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

インストールが完了したら、ホストはアクティベーションキーを使用して Satellite Server にも登録し、必要な設定と管理ツールを Satellite Client 6 のリポジトリーからインストールします。

CLI 手順

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

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

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

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

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

```
# hammer host create \
--build true \
```

```
--enabled true \
--hostgroup "My_Host_Group" \
--location "My_Location" \
--mac "My_MAC_Address" \
--managed true \
--name "My_Host_Name" \
--organization "My_Organization" \
--pxe-loader "Grub2 UEFI HTTP"
```

4. **hammer host interface update** コマンドを使用し、ネットワークインターフェイスのオプションが設定されていることを確認します。

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

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

これで、ホストのエントリおよび関連するプロビジョニングの設定が作成されます。これには、UEFI を使用したベアメタルホストの起動に必要なディレクトリーおよびファイルを作成することも含まれます。物理ホストを起動して、ブートモードを UEFI HTTP に設定すると、ホストは定義済みの DHCP サービスを検出し、キックスタートツリーで Capsule の HTTP エンドポイントを受信して、オペレーティングシステムをインストールします。

インストールが完了したら、ホストはアクティベーションキーを使用して Satellite Server にも登録し、必要な設定と管理ツールを Satellite Client 6 のリポジトリーからインストールします。

5.6. プロビジョニング中の SSH キーのデプロイ

この手順を使用して、ユーザーに追加した SSH キーは、プロビジョニング中にデプロイします。ユーザーに SSH キーを追加する方法については、[Red Hat Satellite の管理の ユーザーの SSH キー管理](#) を参照してください。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
2. プロビジョニングテンプレートを作成するか、クローンを作成して既存のテンプレートを編集します。詳細は、「[プロビジョニングテンプレートの作成](#)」を参照してください。
3. テンプレートで **テンプレート** タブをクリックします。
4. **テンプレートエディター** のフィールドで、**create_users** スニペットを **%post** セクションに追加します。

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

5. **デフォルト** チェックボックスを選択します。

6. **Association** タブをクリックします。
7. **適用可能なオペレーティングシステム** リストから適切なオペレーティングシステムを選択します。
8. **Submit** をクリックしてプロビジョニングテンプレートを保存します。
9. ホストをプロビジョニングテンプレートに関連付けて作成するか、修正したテンプレートが関連付けられた OS を使用してホストを再ビルドします。詳細は、[ホストの管理](#) の [ホストの作成](#) を参照してください。

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

第6章 IPXE を使用したプロビジョニング時間の短縮

iPXE は、オープンソースのネットワークブートファームウェアです。HTTP サーバーからのブートなどの追加機能で強化された完全な PXE 実装を提供します。iPXE の詳細は、iPXE の [Web サイト](#) を参照してください。

以下の制限により PXE を使用できない場合は、iPXE を使用できます。

- マネージド外の DHCP サーバーのあるネットワーク。
- ファイアウォールの制限などにより到達できない PXE サービス。
- 低帯域幅ネットワークなどが原因で、信頼性の低い TFTP UDP ベースのプロトコル。

6.1. iPXE を使用するための前提条件

以下の場合には、iPXE を使用して仮想マシンを起動できます。

- 仮想マシンが iPXE をプライマリーファームウェアとして使用するハイパーバイザー上で動作する。
- 仮想マシンが BIOS モードにある。この場合、iPXE をチェーンブートし、HTTP プロトコルを使用してブートするように PXELinux を設定できます。

HTTP を使用して UEFI モードで仮想マシンを起動する場合は、代わりに「[UEFI HTTP ブートプロビジョニングを使用したホストの作成](#)」の説明に従います。

サポート性

Red Hat は、Red Hat Satellite での iPXE を正式にサポートしていません。詳細は、[Red Hat ナレッジベースの Supported architectures and kickstart scenarios in Satellite 6](#) を参照してください。

ホスト要件

- プロビジョニングインターフェイスの MAC アドレスが、ホスト設定と一致する。
- ホストのプロビジョニングインターフェイスには、有効な DHCP 予約がある。
- NIC は PXE ブートが可能である。詳細は、[supported hardware on ipxe.org](#) のページで、iPXE ベースのブートディスクと機能することが想定されているハードウェアドライブのリストを参照してください。
- NIC は iPXE と互換性がある。

6.2. iPXE 環境の設定

iPXE プロビジョニングに使用するすべての Capsule で iPXE 環境を設定します。



重要

Red Hat Enterprise Linux では、iPXE のセキュリティー関連機能はサポートされておらず、iPXE バイナリーはセキュリティー機能なしで構築されています。このため、HTTPS ではなく、HTTP のみが使用できます。詳細情報は、[Red Hat Enterprise Linux HTTPS support in iPXE](#) を参照してください。

前提条件

- Satellite Server の代わりに Capsule Server を使用する場合は、Capsule Server が適切に設定されていることを確認してください。詳細は、[Capsule Server のインストールのホストの登録とプロビジョニングのための Capsule の設定](#) を参照してください。

手順

1. Capsule で TFTP サービスおよび HTTPboot サービスを有効にします。

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

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

```
# dnf install ipxe-bootimgs
```

3. iPXE ファームウェアを TFTP ディレクトリーにコピーします。

- Linux カーネルヘッダーで iPXE ファームウェアをコピーします。

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

- UNDI iPXE ファームウェアをコピーします。

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

4. SELinux ファイルコンテキストを修正します。

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

5. HTTP URL を設定します。

- Satellite Server を起動に使用する場合は、Satellite Server で次のコマンドを実行します。

```
# satellite-installer \  
--foreman-proxy-dhcp-ipxefilename "http://satellite.example.com/unattended/iPXE?  
bootstrap=1"
```

- Capsule Server を起動に使用する場合は、Capsule Server で以下のコマンドを実行します。

```
# satellite-installer --foreman-proxy-dhcp-ipxe-bootstrap true
```

6.3. 仮想マシンの起動

仮想化ハイパーバイザーの一部は、PXE ブートのプライマリーファームウェアとして iPXE を使用します。このようなハイパーバイザーを使用する場合は、TFTP および PXELinux なしで仮想マシンを起動できます。

仮想マシンの起動には、次のワークフローがあります。

1. 仮想マシンが起動します。
2. iPXE が DHCP を使用して、HTTP URL を含むネットワーク認証情報を取得します。
3. iPXE は、Capsule から iPXE ブートストラップテンプレートをロードします
4. iPXE は、Foreman から URL パラメーターとして MAC を使用して iPXE テンプレートをロードします
5. iPXE が、インストーラーのカーネルおよび初期 RAM ディスクをロードします。

前提条件

- お使いのハイパーバイザーが iPXE をサポートしている。以下の仮想化ハイパーバイザーは、iPXE をサポートします。
 - libvirt
 - Red Hat Virtualization (非推奨)
- iPXE 環境を設定している。詳細は、「[iPXE 環境の設定](#)」を参照してください。



注記

以下で説明するように、Satellite に同梱されている元のテンプレートを使用できます。元のテンプレートを変更する必要がある場合は、テンプレートのクローンを作成し、クローンを編集して、元のテンプレートの代わりにクローンを関連付けます。詳細は、「[プロビジョニングテンプレートのクローン作成](#)」を参照してください。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
2. **Kickstart default iPXE** テンプレートを検索します。
3. テンプレートの名前をクリックします。
4. **Association** タブをクリックし、ホストが使用するオペレーティングシステムを選択します。
5. **Locations** タブをクリックして、ホストがある場所を追加します。
6. **Organizations** タブをクリックして、ホストが属する組織を追加します。
7. **Submit** をクリックして変更を保存します。
8. Satellite Web UI で、**Hosts > Operating systems** に移動し、ホストのオペレーティングシステムを選択します。
9. **Templates** タブをクリックします。
10. **iPXE template** リストから、**Kickstart default iPXE** テンプレートを選択します。
11. **Submit** をクリックして変更を保存します。
12. Satellite Web UI で、**Hosts > All Hosts** に移動します。
13. **Hosts** のページで、使用するホストを選択します。

14. **Operating System** タブを選択します。
15. **PXE Loader** を **iPXE Embedded** に設定します。
16. **Templates** タブを選択します。
17. **Provisioning Templates** で **Resolve** をクリックし、**iPXE template** が必要なテンプレートに解決されることを確認します。
18. **Submit** をクリックしてホスト設定を保存します。

6.4. PXELINUX からの iPXE のチェーンブート

ネットワーク通信の組み込みドライバー (**ipxe.lkrn**) またはユニバーサルネットワークデバイスインターフェイス (UNDI) (**undionly-ipxe.0**) を使用するように iPXE をセットアップできます。ネットワークハードウェアの機能と iPXE ドライバーの可用性に応じて、どちらのファイルをロードするかを選択できます。

UNDI は、TFTP クライアントを実装する最小限の UDP/IP スタックです。ただし、UNDI は HTTP などの他のプロトコルをサポートできません。iPXE で HTTP を使用するには、組み込みのドライバー (**ipxe.lkrn**) を備えた iPXE ビルドを使用します。

iPXE のチェーンブートには次のワークフローがあります。

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

前提条件

- iPXE 環境を設定している。詳細は、[「iPXE 環境の設定」](#) を参照してください。



注記

以下で説明するように、Satellite に同梱されている元のテンプレートを使用できます。元のテンプレートを変更する必要がある場合は、テンプレートのクローンを作成し、クローンを編集して、元のテンプレートの代わりにクローンを関連付けます。詳細は、[「プロビジョニングテンプレートのクローン作成」](#) を参照してください。

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
2. 必要な PXELinux テンプレートを検索します。

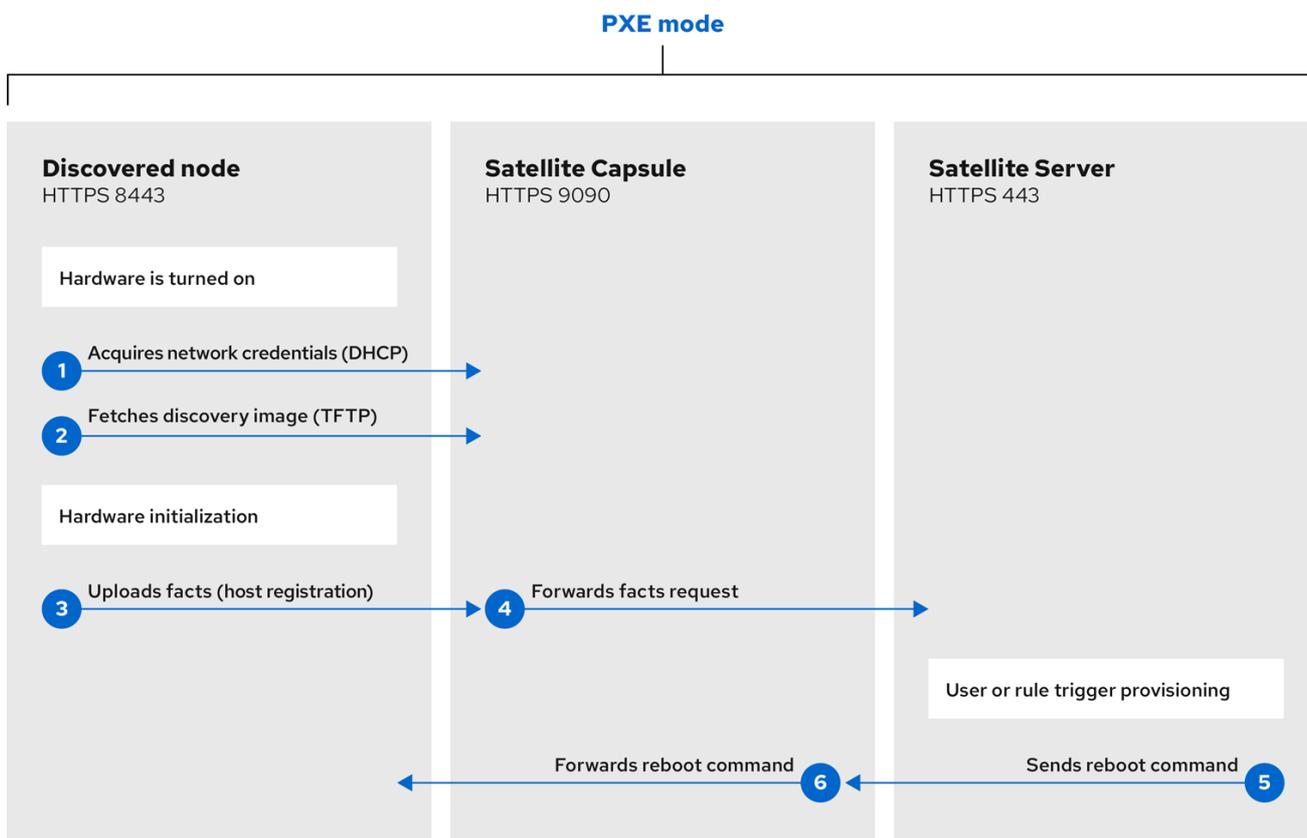
- **ipxe.lkrn** を使用するための **PXELinux chain iPXE**
 - **undionly-ipxe.0** を使用するための **PXELinux chain iPXE UNDI**
3. 使用するテンプレートの名前をクリックします。
 4. **Association** タブをクリックし、ホストが使用するオペレーティングシステムを選択します。
 5. **Locations** タブをクリックして、ホストがある場所を追加します。
 6. **Organizations** タブをクリックして、ホストが属する組織を追加します。
 7. **Submit** をクリックして変更を保存します。
 8. **Provisioning Templates** ページで、**Kickstart default iPXE** テンプレートを検索します。
 9. テンプレートの名前をクリックします。
 10. **Association** タブをクリックし、ホストが使用するオペレーティングシステムにテンプレートを関連付けます。
 11. **Locations** タブをクリックして、ホストがある場所を追加します。
 12. **Organizations** タブをクリックして、ホストが属する組織を追加します。
 13. **Submit** をクリックして変更を保存します。
 14. Satellite Web UI で、**Hosts > Operating systems** に移動し、ホストのオペレーティングシステムを選択します。
 15. **Templates** タブをクリックします。
 16. **PXELinux template** リストから、使用するテンプレートを選択します。
 17. **iPXE template** リストから、**Kickstart default iPXE** テンプレートを選択します。
 18. **Submit** をクリックして変更を保存します。
 19. Satellite Web UI で、**Configure > Host Groups** に移動し、設定するホストグループを選択します。
 20. **Operating System** タブを選択します。
 21. **アーキテクチャー** と **オペレーティングシステム** を選択します。
 22. **PXE Loader** を設定します。
 - **PXELinux BIOS** を選択して、PXELinux から iPXE (**ipxe.lkrn**) をチェーンブートします。
 - **iPXE Chain BIOS** を選択して、**undionly-ipxe.0** を直接ロードします。

第7章 DISCOVERY サービスの設定

Red Hat Satellite は、Satellite インベントリに含まれていないネットワーク上に存在するホストを検出できます。これらのホストは、ハードウェアの検出を実行し、この情報を Satellite Server に送り返す検出イメージを起動します。これにより、各ホストの MAC アドレスを入力せずに Satellite Server でプロビジョニング可能なホストのリストが作成されます。

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

Satellite Server では、Discovery サービスはデフォルトで有効になっていますが、グローバルテンプレートは、デフォルトで、ローカルハードドライブから起動するように設定されています。デフォルトの設定を変更するには、Satellite Web UI で、**Administer > Settings** に移動して、**Provisioning** タブをクリックします。デフォルトの PXE グローバルテンプレートエントリーの列を特定し **値** の行に **discovery** と入力します。次に、PXE のデフォルト設定を構築します。**Hosts > Templates > Provisioning Templates** に移動し、**Build PXE Default** をクリックします。Satellite は、PXE 設定を任意の TFTP Capsule に配布します。



278_Satellite_0922

検出に使用する予定のすべてのサブネットの DHCP 範囲が、マネージド DHCP サービス用に設定された DHCP リースプールと交差しないことを確認してください。DHCP 範囲は Satellite Web UI で設定され、リースプール範囲は **satellite-installer** コマンドを使用して設定されます。たとえば、10.1.0.0/16 のネットワーク範囲では、10.1.0.0 から 10.1.127.255 をリースに割り当て、10.1.128.0 から 10.1.255.254 を予約に割り当てることができます。

ネットワークインターフェイス名は、Red Hat Enterprise Linux の **メジャーバージョン間** で常に同じであるとは限らないため、ホストまたは仮想マシンが誤ったネットワーク設定で作成される可能性があります。Dell のサーバーでは、新しい命名スキームは **biosdevname=1** カーネルコマンドラインオプション

ンを介して無効にできます。他のハードウェアまたは仮想マシンの場合、新しい命名スキームは **net.ifnames=1** を介して完全にオフにできます。



注記

Satellite で提供される Foreman Discovery イメージは、Red Hat Enterprise Linux 8 をベースにしています。

手順

1. Discovery を Satellite Server にインストールします。

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

2. **foreman-discovery-image** をインストールします。

```
# satellite-maintain packages install foreman-discovery-image
```

foreman-discovery-image パッケージは、Discovery ISO を **/usr/share/foreman-discovery-image/** ディレクトリーにインストールします。

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

7.1. DISCOVERY サービスのインストール

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

手順

1. Capsule Server で以下のコマンドを入力します。

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

2. **foreman-discovery-image** をインストールします。

```
# satellite-maintain packages install foreman-discovery-image
```

サブネット

検出可能なホストを含むすべてのサブネットには、Discovery サービスを提供するために適切な Capsule Server が選択されている必要があります。

Satellite Web UI で、**Infrastructure > Subnets** に移動してサブネットを選択します。次に、Capsule タブに移動し、使用する **Discovery Proxy** を選択します。使用するサブネットごとにこれを実行します。

7.2. テンプレート PXELINUX DISCOVERY スニペットのプロビジョニング

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

```
LABEL discovery
MENU LABEL Foreman Discovery Image
KERNEL boot/fdi-image/vmlinuz0
APPEND initrd=boot/fdi-image/initrd0.img rootflags=loop root=live:/fdi.iso rootfstype=auto ro
rd.live.image acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset
proxy.url=<%= foreman_server_url %> proxy.type=foreman
IPAPPEND 2
```

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

UEFI プロビジョニングの場合は、**ホスト > テンプレートのプロビジョニング** ウィンドウの **PXEgrub2 global default** テンプレートに、スニペット **pxegrub2_discovery** が含まれます。

```
menuentry 'Foreman Discovery Image' --id discovery {
  linuxefi boot/fdi-image/vmlinuz0 rootflags=loop root=live:/fdi.iso rootfstype=auto ro rd.live.image
  acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset proxy.url=<%=
  foreman_server_url %> proxy.type=foreman BOOTIF=01-$mac
  initrdefi boot/fdi-image/initrd0.img
}
```

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

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

7.3. 検出されたホストの自動コンテキスト

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

1. 検出されたホストが Satellite で定義されたサブネットを使用する場合には、このホストは、サブネットに関連付けられた最初の組織およびロケーションを使用します。
2. デフォルトの場所と組織がグローバル設定で指定されている場合に、検出されたホストはこの組織と場所に配置されます。これらの設定を設定するには、**Administer > Settings > Discovery** に移動し、**Discovery organization** と **Discovery location** の設定の値を選択します。検出ホストのサブネットも、選択した組織と場所に属していることを確認してください。そうでない場合、Satellite はセキュリティー上の理由から設定を拒否します。
3. 上記の条件がいずれも満たされない場合、Satellite は名前順に最初の組織と場所を割り当てます。

Discovered Hosts ページの一括アクションを使用して、組織または場所を手動で変更できます。変更する検出されたホストを選択し、**Select Action** メニューから **Assign Organization** または **Assign Location** を選択します。

7.4. DISCOVERY テンプレートとスニペットの設定

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

Discovery サービスのデフォルト設定

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

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

テンプレートを使用するには、Satellite Web UI で、**Administer > Settings** に移動して、**Provisioning** タブをクリックし、使用するテンプレートを設定します。

テンプレートとスニペットのカスタマイズ

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

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

Satellite Web UI で、**Hosts > Provisioning Templates** に移動し、**Build PXE Default** をクリックします。

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

追加の設定

proxy.url の引数

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

proxy.type の引数

proxy.url の引数に Capsule Server の FQDN を使用する場合は、**proxy.type** 引数を **proxy** に設定することを確認します。Satellite の FQDN を使用する場合は、**proxy.type** の引数を **foreman** に更新してください。

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

Capsule のホスト名のレンダリング

Satellite は、全 TFTP Capsules に同じテンプレートをデプロイし、Capsule のホスト名をレンダリングする変数やマクロがありません。ハードコードされた **proxy.url** は、複数の TFTP Capsule を連携でき

ません。回避策として、**PXE デフォルトの構築** をクリックするたびに、SSH を使用して TFTP ディレクトリーの設定ファイルを編集するか、適切なサブネットの共通 DNS エイリアスを使用してください。

タグ付けされた VLAN プロビジョニング

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

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

7.5. 検出されたホストからのホストの作成

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

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

前提条件

- Satellite でドメインとサブネットを設定する。ネットワーク要件の詳細は、[3章 ネットワークの設定](#) を参照してください。
- Satellite で Discovery サービスを設定する。詳細は、「[Discovery サービスのインストール](#)」を参照してください。
- ベアメタルホストまたは空の仮想マシン
- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。

セキュリティトークンの詳細は、「[セキュリティトークンの有効期間の設定](#)」を参照してください。

手順

1. Satellite Web UI で **ホスト > 検出されたホスト** に移動します。使用するホストを選択して、リストの右側にある **プロビジョニング** をクリックします。
2. 以下の2つのオプションから1つ選択します。
 - ホストグループからホストをプロビジョニングするには、ホストグループ、組織、場所を選択してから、**ホストの作成** をクリックします。
 - さらにカスタマイズしてホストをプロビジョニングするには、**ホストのカスタマイズ** をクリックして、新規ホストに指定する追加情報を入力します。
3. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - Satellite Server は新規ホストの IP アドレスを自動的に割り当てます。

- Satellite Server は Discovery の結果をもとに MAC アドレスを自動的に設定します。
4. Satellite Server が、ホストの最初のインターフェイスに **Managed (管理)**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
 5. **オペレーティングシステム** タブをクリックして、すべてのフィールドに値が含まれていることを確認します。オペレーティングシステムの各要素を確認してください。
 6. **プロビジョニングテンプレートの 解決** をクリックし、新規ホストが使用する適切なプロビジョニングテンプレートを特定できることを確認します。ホストは、以下のプロビジョニングテンプレートを解決する必要があります。
 - **kexec テンプレート: Discovery Red Hat kexec**
 - **provision テンプレート: Satellite Kickstart Default**
プロビジョニングテンプレートの関連付けの詳細は、[「プロビジョニングテンプレート」](#)を参照してください。
 7. **Submit** をクリックしてホストの詳細を保存します。

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

CLI 手順

1. プロビジョニングに使用する検出されたホストを特定します。

```
# hammer discovery list
```

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

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

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

7.6. DISCOVERY ルールの作成

検出されたホストのプロビジョニングプロセスの自動化方法として、Satellite は Discovery ルールを作成する機能を提供します。これらのルールは、検出されたホストが、割り当てられたホストグループをベースに自らを自動的にプロビジョニングする方法を定義します。たとえば、CPU 数の多いホストを

ハイパーバイザーとして自動的にプロビジョニングすることができます。同様に、ハードディスクが大容量のホストは、ストレージサーバーとしてプロビジョニングすることもできます。

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

NIC の考慮事項

現在、自動プロビジョニングでは NIC の設定はできません。すべてのシステムは、Discovery 時に検出された NIC 設定でプロビジョニングされていますが、スクリプトを使用するか、後で設定管理を設定して **kickstart** スクリプトレットで NIC を設定できます。

手順

1. Satellite Web UI で、**Configure > Discovery rules** に移動し、**Create Rule** を選択します。
2. **名前** フィールドには、ルールの名前を入力します。
3. **検索** には、ホストをプロビジョニングするかどうかを決定するためのルールを入力します。このフィールドには、入力する値についての推奨案が提供され、複数のルールに演算子を使用できます。例: **cpu_count > 8**
4. **ホストグループ** リストから、このホストのテンプレートとして使用するホストグループを選択します。
5. **ホスト名** フィールドには、複数ホストのホスト名を決定するためのパターンを入力します。これはプロビジョニングテンプレートと同じ ERB 構文を使用します。ホスト名は、ホスト固有の値に **@host** 属性を、乱数に **rand** マクロを、または値を増やすために **sequence_hostgroup_param_next** マクロを使用できます。テンプレートのプロビジョニングに関する詳細は、「[プロビジョニングテンプレート](#)」および API ドキュメントを参照してください。
 - **myhost-<%= sequence_hostgroup_param_next("EL7/MyHostgroup", 10, "discovery_host") %>**
 - **myhost-<%= rand(99999) %>**
 - **abc-<%= @host.facts['bios_vendor'] %>-<%= rand(99999) %>**
 - **xyz-<%= @host.hostgroup.name %>**
 - **srv-<%= @host.discovery_rule.name %>**
 - **server-<%= @host.ip.gsub(':', '-') + '-' + @host.hostgroup.subnet.name %>**
ホスト名のパターンの作成時には、作成するホスト名が一意の名前であることを確認してください。ホスト名は数字で始めることができず、アンダースコアやドットを含めることができません。適切な方法として、**facter** で提供される固有の情報 (MAC アドレス、BIOS、またはシリアル ID など) を使用することができます。
6. **ホストの制限** フィールドには、ルールを使用してプロビジョニングできるホストの最大数を入力します。無制限に設定するには **0** を使用します。
7. **優先度** フィールドには、ルール間の優先度を設定する数値を入力します。値が低いルールほど優先度が高くなります。
8. **有効化** リストから、ルールを有効化するかどうかを選択します。

9. ルールに異なるプロビジョニングコンテキストを設定するには、**組織** および **ロケーション** タブをクリックして、使用するコンテキストを選択します。
10. **Submit** をクリックしてルールを保存します。
11. Satellite Web UI で、**Hosts > Discovered Host** に移動して、以下の 2 つのオプションから 1 つを選択します。
 - 右側の **検出されたホスト** リストから、**自動プロビジョニング** を選択して、単一のホストを自動的にプロビジョニングします。
 - ウィンドウの右上の **すべてを自動プロビジョニング** をクリックして、全ホストを自動的にプロビジョニングします。

CLI 手順

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

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

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

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

7.7. PXE を使用しない DISCOVERY の実装

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

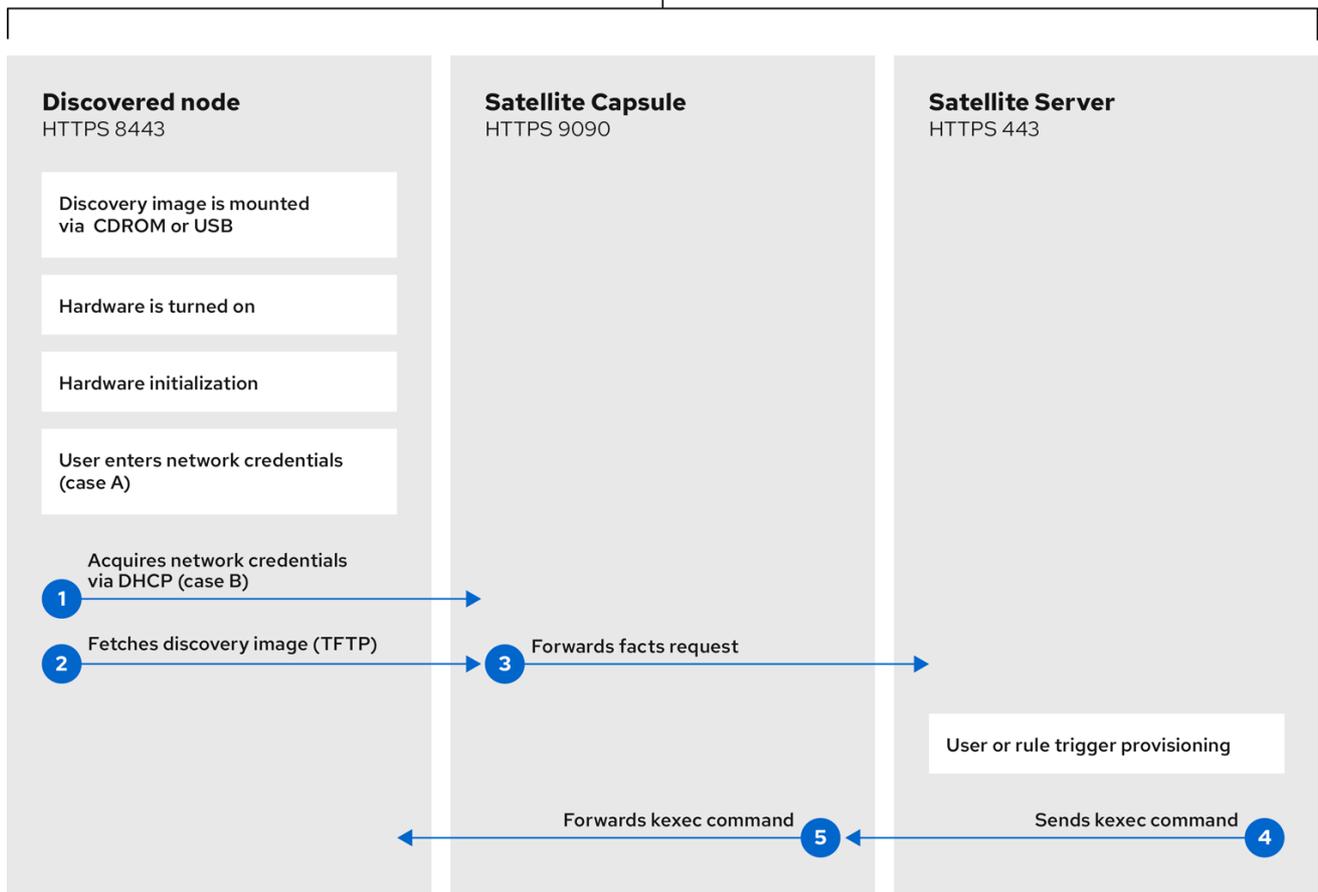
重要

Discovery **kexec** は、テクノロジープレビュー機能のみとなっています。テクノロジープレビュー機能は、Red Hat 製品サポートのサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。これらの機能により、近日発表予定の製品機能をリリースに先駆けてご提供でき、お客様は開発プロセス時に機能をテストして、フィードバックをお寄せいただくことができます。Red Hat テクノロジープレビュー機能のサポート範囲に関する情報は、<https://access.redhat.com/ja/support/offerings/techpreview> を参照してください。

既知の問題

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

PXE-less mode



278_Satellite_0922

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

手動による使用

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

```
# dd bs=4M \
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.4.4-5.iso \
of=/dev/sdb
```

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

Manual network setup を選択する場合には、Discovery イメージはネットワークオプションのセットを要求します。これには、Satellite Server に接続されるプライマリネットワークインターフェイスが含まれます。この Discovery イメージは、**IPv4 アドレス**、**IPv4 ゲートウェイ** および **IPv4 DNS サーバー** などのネットワークインターフェイスの設定オプションも要求します。

3. これらの詳細を入力した後に、**次へ** を選択します。
4. **Discovery with DHCP** を選択する場合には、Discovery イメージは Satellite Server に接続されるプライマリネットワークインターフェイスのみを要求します。このサービスは、Capsule

Server が提供するサーバーなどの DHCP サーバーを使用してネットワークインターフェイスを自動的に設定しようとします。

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

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

6. **Connection type** を **Proxy** に設定し、**次へ** を選択します。
7. オプション: Discovery イメージは、Facter ツールが Satellite Server に送り返す **Custom facts (カスタムファクト)** を入力するための一連のフィールドも提供します。これらは **名前-値** の形式で入力します。必要なカスタムファクトを指定し、**確認** を選択して続きます。

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

検出されたホストのプロビジョニングに関する詳細情報は、[「検出されたホストからのホストの作成」](#) を参照してください。

7.8. 無人での使用、カスタマイズ、イメージのリマスター

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

fdi.cachefacts

キャッシュせずにアップロードされたファクトの数。デフォルトでは、Satellite はアップロードされたファクトをキャッシュしません。

fdi.countdown

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

fdi.dhcp_timeout

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

fdi.dns_nameserver

DNS SRV レコードに使用するネームサーバー。

fdi.dns_ndots

DNS SRV レコードに使用する **ndots** オプション。

fdi.dns_search

DNS SRV レコードに使用する検索ドメイン。

fdi.initnet

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

fdi.ipv4.method

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

fdi.ipv6.method

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

fdi.ipwait

HTTP プロキシの SSL 証明書の開始で IP が使用可能になるまで待機する期間 (秒単位)。デフォルトでは、Satellite は 120 秒待機します。

fdi.nmwait

NetworkManager の **nmcli -wait** オプション。デフォルトでは、**nmcli** は 120 秒待機します。

fdi.proxy_cert_days

自己署名 HTTPS 証明書が有効な日数。デフォルトでは、証明書は 999 日間有効です。

fdi.pxauto

自動化モードまたは準自動化モードを設定します。0 に設定した場合には、イメージは準自動化モードを使用します。このモードでは、一連のダイアログオプションで選択内容を確認できます。1 に設定した場合、イメージは自動化モードを使用し、確認なしに次に進みます。

fdi.pxfactname1、fdi.pxfactname2 ... fdi.pxfactnameN

カスタムファクト名を指定できます。

fdi.pxfactvalue1、fdi.pxfactvalue2 ... fdi.pxfactvalueN

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

fdi.pxip、fdi.pxgw、fdi.pxdns

プライマリーネットワークインターフェイスの IP アドレス (**fdi.pxip**)、ゲートウェイ (**fdi.pxgw**)、および DNS (**fdi.pxdns**) を手動で設定します。これらのパラメーターを省略する場合、イメージは DHCP を使用してネットワークインターフェイスを設定します。コンマ区切りリストに複数の DNS エントリーを追加できます。[1] たとえば、**fdi.pxdns=192.168.1.1,192.168.200.1** などです。

fdi.pxmac

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

fdi.rootpw

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

fdi.ssh

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

fdi.uploadsleep

factor 実行間の期間 (秒単位)。デフォルトでは、factor は 30 秒ごとに実行されます。

fdi.vlan.primary

プライマリーインターフェイスに設定する VLAN タグ付け ID。

fdi.zips

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

fdi.zipserver

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

proxy.type

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

proxy.url

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

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

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

```
# discovery-remaster ~/iso/foreman-discovery-image-3.4.4-5.iso \  
"fdi.pxip=192.168.140.20/24 fdi.pxgw=192.168.140.1 \  
fdi.pxdns=192.168.140.2 proxy.url=https://satellite.example.com:9090 \  
proxy.type=proxy fdi.pxfactname1=customhostname fdi.pxfactvalue1=myhost \  
fdi.pxmac=52:54:00:be:8e:8c fdi.pxauto=1"
```

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

```
# dd bs=4M \  
if=/usr/share/foreman-discovery-image/foreman-discovery-image-3.4.4-5.iso \  
of=/dev/sdb
```

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

検出されたホストのプロビジョニングに関する詳細情報は、「[検出されたホストからのホストの作成](#)」を参照してください。

7.9. DISCOVERY イメージの拡張

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

手順

- 最初に、以下のディレクトリー構造を作成します。

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

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

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

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

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

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

既存の環境変数 (**PATH**、**LD_LIBRARY_PATH**、**RUBYLIB** および **FACTERLIB**) に、新規ディレクティブとオプションを追加できます。スクリプトで明示的にパスを指定する場合には、**.zip** ファイルのコメントが、イメージの **/opt/extension** ディレクトリーにデプロイメントされます。

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

7.10. DISCOVERY のトラブルシューティング

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

- Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
 1. **Build PXE Default** をクリックして、デフォルトの PXELinux テンプレートを再デプロイします。
- TFTP Capsule Server で **pxelinux.cfg/default** 設定ファイルを確認します。
- ホスト、Capsule Server、および Satellite Server 間で適切なネットワーク接続があることを確認します。
- 使用している PXELinux テンプレートに含まれている PXE 検出スニペットを確認します。スニペットの名前は **pxelinux_discovery**、**pxegrub_discovery**、または **pxegrub2_discovery** です。PXE 検出スニペットの **proxy.url** オプションと **proxy.type** オプションを検証してください。

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

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

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

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

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

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

[1] NetworkManager は、リスト区切り文字として ; を想定していますが、現在は , も想定しています。詳細は、**man nm-settings-keyfile** および [Shell-like scripting in GRUB](#) を参照してください。

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

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



注記

TAR イメージをビルドするためのブループリントには、常にカーネルパッケージが含まれている必要があります。

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

前提条件

- Red Hat Image Builder を使用して作成された既存の TAR イメージ。

手順

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

ベアメタルプロビジョニングや、コンピュートリソースを使用したプロビジョニングに、このイメージを使用できます。ベアメタルプロビジョニングの詳細は、[5章 PXE を使用したホストのプロビジョニング](#) を参照してください。異なるコンピュートリソースを使用したプロビジョニングの詳細は、使用するコンピュートリソースに関する章を参照してください。

第9章 KVM (LIBVIRT) での仮想マシンのプロビジョニング

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

管理できるのは、Satellite を介して作成された仮想マシンのみです。ディレクトリーストレージプールタイプ以外の仮想マシンはサポートされていません。

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

前提条件

- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。
- KVM サーバーでネットワークを管理する Capsule Server。Capsule Server との競合を避けるために、他の DHCP サービスがこのネットワークで実行されていないことを確認します。Capsule Server のネットワークサービス設定の詳細は、[ホストのプロビジョニングのネットワークの設定](#) 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 (libvirt daemon)。詳細は、[Red Hat Enterprise Linux 8 仮想化の設定と管理](#) を参照してください。
- 既存の仮想マシンイメージ (イメージベースのプロビジョニングを使用する場合)。このイメージが KVM ホストのストレージプールにあることを確認します。[デフォルトの](#) ストレージプールは通常 `/var/lib/libvirt/images` にあります。Satellite 経由で管理できるのは、ディレクトリーのプールストレージタイプのみです。
- オプション: この手順の例では、KVM の root ユーザーを使用します。KVM サーバーで root 以外のユーザーを使用する場合には、KVM サーバーの **libvirt** グループにユーザーを追加します。

```
# usermod -a -G libvirt non_root_user
```

関連情報

- 管理者以外のユーザーがホストをプロビジョニングするために必要なパーミッションのリストは、[付録E ホストをプロビジョニングするために必要なパーミッション](#) を参照してください。

9.1. SATELLITE SERVER の KVM 接続設定

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

手順

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

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

2. キーペアを生成します。

```
$ ssh-keygen
```

3. 公開キーを KVM サーバーにコピーします。

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

4. **foreman** ユーザーのバッシュシェルを終了します。

```
$ exit
```

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

```
# satellite-maintain packages install libvirt-client
```

6. 以下のコマンドを使用して、KVM サーバーへの接続をテストします。

```
# su foreman -s /bin/bash -c 'virsh -c qemu+ssh://root@kvm.example.com/system list'
```

9.2. SATELLITE SERVER への KVM 接続の追加

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

手順

1. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、**Create Compute Resource** をクリックします。
2. **Name** フィールドに、新しいコンピュートリソースの名前を入力します。
3. **Provider** リストから、**Libvirt** を選択します。
4. **Description** フィールドに、コンピュートリソースの説明を入力します。
5. **URL** フィールドには、KVM サーバーへの接続 URL を入力します。以下に例を示します。

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

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

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

- パスワードは、virt-managerなどで、仮想マシンのコンソールを開くたびに無作為に生成されます。
8. **テスト接続** をクリックして Satellite Server が KVM サーバーに問題なく接続できることを確認します。
 9. **ロケーション** および **組織** タブは現在のコンテキストに自動的に設定されていることを確認します。他のコンテキストをこれらのタブに追加します。
 10. **Submit** をクリックして KVM 接続を保存します。

CLI手順

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

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

9.3. SATELLITE SERVER での KVM イメージの追加

イメージベースのプロビジョニングを使用してホストを作成するには、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動し、KVM 接続の名前をクリックします。
2. **イメージの作成** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。
4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。
7. **パスワード** フィールドには、イメージにアクセスするための SSH パスワードを入力します。
8. **イメージパス** フィールドには、KVM サーバーのイメージを参照する完全パスを入力します。以下に例を示します。

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

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

CLI手順

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

```
# hammer compute-resource image create \
--name "KVM Image" \
--compute-resource "My_KVM_Server" \
--operatingsystem "RedHat version" \
--architecture "x86_64" \
--username root \
--user-data false \
--uuid "/var/lib/libvirt/images/KVMimage.qcow2" \
```

9.4. KVM の詳細のコンピュートプロファイルへの追加

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. KVM コンピュートリソースの名前をクリックします。
4. **CPU** フィールドには、新規ホストに割り当てる CPU の数を入力します。
5. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を入力します。
6. **イメージ** リストから、イメージベースのプロビジョニングを実行する場合には、使用するイメージを選択します。
7. **ネットワークインターフェイス** リストから、ホストのネットワークインターフェイスのネットワークパラメーターを選択します。ネットワークインターフェイスは、複数作成することができますが、少なくとも1つのインターフェイスが Capsule で管理されるネットワークを参照している必要があります。
8. **ストレージ** エリアには、ホストのストレージパラメーターを入力します。ホストのボリュームは複数作成できます。
9. **Submit** をクリックしてコンピュートプロファイルの設定を保存します。

CLI 手順

1. 以下のコマンドを実行してコンピュータプロファイルを作成します。

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

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

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

9.5. KVM でのホストの作成

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

- ネットワーク接続経由でホストを作成する場合には、ホストが PXE プロビジョニングサービスにアクセスできるように、新規ホストは KVM 仮想マシン上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスする必要があります。この新しいホストエントリーにより、KVM サーバーが仮想マシンを作成して起動するようにトリガーされます。仮想マシンが仮想ネットワークで定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、選択したオペレーティングシステムのインストールを開始します。
- 既存のイメージでホストを作成する場合は、新規ホストのエントリーは、KVM サーバーが新規ボリュームのベースとして既存のイメージを使用し、仮想マシンを作成するようトリガーします。

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

DHCP の競合

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、KVM 接続を選択します。

7. **コンピュータプロファイル** リストから、仮想マシンの設定を自動的に投入するために使用するプロファイルを選択します。KVM 固有のフィールドにコンピュータプロファイルの設定が入力されていることを確認します。必要に応じてこれらの設定を変更します。
8. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
9. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - **Satellite** は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。KVM は、プロビジョニング中にホストに MAC アドレスを割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - **Satellite** が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
10. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
11. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
12. **Provisioning Method** のページで、使用するホストを選択します。
 - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
 - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
13. **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。
14. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュータプロファイルからの情報が入力されていることを確認します。必要に応じてこれらの設定を変更します。
15. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
16. **Submit** をクリックしてホストエントリーを保存します。

CLI手順

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

```
# hammer host create \
--build true \
--compute-attributes="cpus=1,memory=1073741824" \
--compute-resource "My_KVM_Server" \
--enabled true \
--hostgroup "My_Host_Group" \
--interface
```

```
"managed=true,primary=true,provision=true,compute_type=network,compute_network=exam  
plenetwork" \  
--location "My_Location" \  
--managed true \  
--name "My_Host_Name" \  
--organization "My_Organization" \  
--provision-method "build" \  
--root-password "My_Password" \  
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```

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

```
# hammer host create \  
--compute-attributes="cpus=1,memory=1073741824" \  
--compute-resource "My_KVM_Server" \  
--enabled true \  
--hostgroup "My_Host_Group" \  
--image "My_KVM_Image" \  
--interface  
"managed=true,primary=true,provision=true,compute_type=network,compute_network=exampl  
enetwork" \  
--location "My_Location" \  
--managed true \  
--name "My_Host_Name" \  
--organization "My_Organization" \  
--provision-method "image" \  
--volume="pool_name=default,capacity=20G,format_type=qcow2"
```

このコンピュートリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

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

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

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

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

前提条件

- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。
- Red Hat Virtualization 環境で、論理ネットワークを管理する Capsule Server。Capsule Server との競合を避けるために、他の DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、[ホストのプロビジョニングのネットワークの設定](#) を参照してください。
- イメージベースのプロビジョニングを使用する場合の、**blank** テンプレート以外の既存のテンプレート。仮想マシンのテンプレートを作成する方法についての詳細は、[Virtual Machine Management Guideの Templates](#) を参照してください。
- Satellite Server との通信に使用する Red Hat Virtualization で管理者に相当するユーザー。この通信には、**admin@internal** ユーザーを使用しないでください。代わりに、以下のパーミッションで新しい Red Hat Virtualization ユーザーを作成します。
 - システム > システムの設定 > ログインパーミッション
 - ネットワーク > vNIC プロファイルの設定 > 作成
 - ネットワーク > vNIC プロファイルの設定 > プロパティの編集
 - ネットワーク > vNIC プロファイル > 削除
 - ネットワーク > vNIC プロファイル > vNIC プロファイルの仮想マシンへの割り当て
 - ネットワーク > vNIC プロファイル > vNIC プロファイルのテンプレートへの割り当て
 - テンプレート > 操作のプロビジョニング > インポート/エクスポート
 - VM > 操作のプロビジョニング > 作成
 - VM > 操作のプロビジョニング > 削除
 - VM > 操作のプロビジョニング > インポート/エクスポート

- VM > 操作のプロビジョニング > ストレージの編集
- ディスク > 操作のプロビジョニング > 作成
- ディスク > ディスクプロファイル > ディスクプロファイルのタッチ
Red Hat Virtualization で新規ユーザーを作成し、パーミッションを追加する方法の詳細は、[Red Hat Virtualization Administration Guide](#)の [Administering User Tasks From the Administration Portal](#) を参照してください。

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

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

手順

1. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、**Create Compute Resource** をクリックします。
2. **Name** フィールドに、新しいコンピュートリソースの名前を入力します。
3. **プロバイダー** のリストから **RHV** を選択します。
4. **Description** フィールドに、コンピュートリソースの説明を入力します。
5. **URL** フィールドに、Red Hat Virtualization Manager の API の接続 URL を **https://rhv.example.com/ovirt-engine/api/v4** の形式で入力します。
6. **ユーザー** フィールドには、ユーザー名と、Red Hat Virtualization Manager のリソースにアクセスするためのパーミッションを入力します。
7. **パスワード** フィールドで、ユーザーのパスワードを入力します。
8. **データセンターのロード** をクリックして、**データセンター** リストに Red Hat Virtualization 環境のデータセンターを入力します。
9. **データセンター** リストからデータセンターを選択します。
10. **クォータ ID** から Satellite Server で利用可能なリソースを制限するためにクォータを選択します。
11. **X509 証明機関** フィールドには、SSL/TLS アクセスの証明機関を入力します。または、フィールドを空白にすると、サーバーによる最初の API 要求時に、自己署名証明書が生成されます。
12. **ロケーション** タブをクリックして、使用するロケーションを選択します。
13. **組織** タブをクリックして、使用する組織を選択します。
14. **Submit** をクリックしてコンピュートリソースを保存します。

CLI 手順

- **--provider** で **Ovirt** を、**--datacenter** で使用するデータセンター名を指定して、**hammer compute-resource create** コマンドを入力します。

```
# hammer compute-resource create \
```

```
--name "My_RHV" --provider "Ovirt" \
--description "RHV server at rhv.example.com" \
--url "https://rhv.example.com/ovirt-engine/api/v4" \
--user "Satellite_User" --password "My_Password" \
--locations "New York" --organizations "My_Organization" \
--datacenter "My_Datacenter"
```

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

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

手順

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

```
# dnf install cloud-init
```

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

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

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

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

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

イメージベースのプロビジョニングを使用してホストを作成するには、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動し、Red Hat Virtualization 接続の名前をクリックします。
2. **イメージの作成** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。
4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。通常、これは **root** ユーザーになります。

7. パスワード フィールドには、イメージにアクセスするための SSH パスワードを入力します。
8. イメージ リストの Red Hat Virtualization コンピュートリソースからイメージを選択します。
9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
10. **Submit** をクリックしてイメージの詳細を保存します。

CLI 手順

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

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

10.4. CLOUD-INIT テンプレートの準備

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動し、**Create Template** をクリックします。
2. **名前** フィールドには、テンプレートの名前を入力します。
3. **Editor** フィールドで、以下のテンプレートの情報を入力します。

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

<%# Allow user to specify additional SSH key as host parameter -%>
<% if @host.params['sshkey'].present? ||
@host.params['remote_execution_ssh_keys'].present? -%>
ssh_authorized_keys:
<% if @host.params['sshkey'].present? -%>
- <%= @host.params['sshkey'] %>
<% end -%>
<% if @host.params['remote_execution_ssh_keys'].present? -%>
<% @host.params['remote_execution_ssh_keys'].each do |key| -%>
- <%= key %>
<% end -%>
<% end -%>
runcmd:
- |
```

```

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

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

```

4. **タイプ** タブをクリックして、**タイプ** リストから**ユーザーデータのテンプレート** を選択します。
5. **関連付け** タブをクリックして、**適用可能なオペレーティングシステム** リストから、このテンプレートに関連付ける**オペレーティングシステム**を選択します。
6. **ロケーション** タブをクリックして、**ロケーション** リストから、テンプレートに関連付ける**ロケーション**を選択します。
7. **組織** タブをクリックして、**組織** リストから、テンプレートに関連付ける**組織**を選択します。
8. **Submit** をクリックします。
9. Satellite Web UI で、**Hosts > Operating Systems** に移動し、テンプレートに関連付ける**オペレーティングシステム**を選択します。

10. **Templates** タブをクリックし、**User data template** リストから新しいテンプレートの名前を選択します。
11. **Submit** をクリックします。

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

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Red Hat Virtualization コンピュートリソースの名前をクリックします。
4. **クラスター** リストから Red Hat Virtualization 環境のターゲットホストクラスターを選択します。
5. **テンプレート** リストから **コア数** および **メモリー** 設定に使用する RHV テンプレートを選択します。
6. **コア数** フィールドには、新規ホストに割り当てる CPU コア数を入力します。
7. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を入力します。
8. **イメージ** リストから、イメージベースのプロビジョニングに使用するイメージを選択します。
9. **ネットワークインターフェイス** エリアには、ホストのネットワークインターフェイスのネットワークパラメーターを入力します。ネットワークインターフェイスは、複数作成することができますが、少なくとも1つのインターフェイスが Capsule で管理されるネットワークを参照している必要があります。ネットワークインターフェイスごとに以下の情報を入力します。
 - a. **名前** フィールドに、ネットワークインターフェイスの名前を入力します。
 - b. **ネットワーク** リストから、使用する論理ネットワークを選択します。
10. **ストレージ** エリアには、ホストのストレージパラメーターを入力します。ホストのボリュームは複数作成できます。ボリュームごとに以下の情報を入力します。
 - a. **Size (GB)** には、新しいボリュームのサイズを GB 単位で入力します。
 - b. **ストレージドメイン** リストからボリュームのストレージドメインを選択します。
 - c. **ディスクの事前割り当て** から、シンプロビジョニングまたはフルディスクの事前割り当てのいずれかを選択します。
 - d. **Bootable** リストから、ブータブルか、ブータブル以外のボリュームのいずれかを選択します。

11. **Submit** をクリックしてコンピュータプロファイルを保存します。

CLI 手順

1. 以下のコマンドを実行してコンピュータプロファイルを作成します。

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

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

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

10.6. RED HAT VIRTUALIZATION でのホストの作成

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

- ネットワーク接続経由でホストを作成する場合には、ホストが PXE プロビジョニングサービスにアクセスできるように、新規ホストは Red Hat Virtualization 仮想ネットワーク上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスする必要があります。この新しいホストエントリーにより、Red Hat Virtualization サーバーがトリガーされ、仮想マシンを作成して起動します。仮想マシンが仮想ネットワークで定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、選択したオペレーティングシステムのインストールを開始します。
- 既存のイメージでホストを作成する場合は、新規ホストのエントリーは、Red Hat Virtualization サーバーが新規ボリュームのベースとして既存のイメージを使用し、仮想マシンを作成するようトリガーします。

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

DHCP の競合

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。

5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、Red Hat Virtualization 接続を選択します。
7. **コンピュートプロファイル** リストから、仮想マシンの設定を自動的に投入するために使用するプロファイルを選択します。Red Hat Virtualization 固有のフィールドにコンピュートプロファイルの設定が入力されていることを確認します。必要に応じてこれらの設定を変更します。
8. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
9. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。Red Hat Virtualization は、プロビジョニング中にホストに MAC アドレスを割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
10. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
11. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
12. **Provisioning Method** のページで、使用するホストを選択します。
 - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
 - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
13. **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。
14. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュートプロファイルからの情報が入力されていることを確認します。必要に応じてこれらの設定を変更します。
15. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
16. **Submit** をクリックしてホストエントリーを保存します。

CLI 手順

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

```
# hammer host create \  
--build true \  
|
```

```

--compute-attributes="cluster=Default,cores=1,memory=1073741824,start=true" \
--compute-resource "MyRHV_" \
--enabled true \
--hostgroup "My_Host_Group" \
--interface
"managed=true,primary=true,provision=true,compute_name=eth0,compute_network=satnetwork" \
--location "My_Location" \
--managed true \
--name "My_Host_Name" \
--organization "My_Organization" \
--provision-method build \
--volume="size_gb=20G,storage_domain=Data,bootable=true"

```

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

```

# hammer host create \
--compute-attributes="cluster=Default,cores=1,memory=1073741824,start=true" \
--compute-resource "MyRHV_" \
--enabled true \
--hostgroup "My_Host_Group" \
--image "MyRHV_Image_" \
--interface
"managed=true,primary=true,provision=true,compute_name=eth0,compute_network=satnetwork" \
--location "My_Location" \
--managed true \
--name "My_Host_Name" \
--organization "My_Organization" \
--provision-method "image" \
--volume="size_gb=20G,storage_domain=Data,bootable=true"

```

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

第11章 VMWARE VSPHERE での仮想マシンのプロビジョニング

VMware vSphere は VMware のエンタープライズクラスの仮想化プラットフォームです。Red Hat Satellite は、新規仮想マシンの作成やそれらの電源状態の制御を含む、vSphere プラットフォームとの対話を実行することができます。

11.1. VMWARE プロビジョニングの前提条件

VMware vSphere プロビジョニングの要件には以下が含まれます。

- サポートされているバージョンの VMware vCenter Server。次のバージョンは、Satellite で完全にテストされています。
 - vCenter Server 7.0
 - vCenter Server 6.7 (EOL)
 - vCenter Server 6.5 (EOL)
- vSphere 環境でネットワークを管理する Capsule Server。Capsule Server との競合を避けるために、他の DHCP サービスがこのネットワークで実行されていないことを確認します。詳細は、[3章 ネットワークの設定](#) を参照してください。
- 既存の VMware テンプレート (イメージベースのプロビジョニングを使用する場合)。
- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。

11.2. VMWARE ユーザーの作成

VMware vSphere サーバーには、Satellite Server の通信用に管理者に相当するユーザーが必要になります。セキュリティ上の理由により、この通信に **administrator** ユーザーを使用しないでください。その代わりに、以下のパーミッションを持つユーザーを作成してください。

VMware vCenter Server バージョン 6.7 の場合は、以下のパーミッションを設定します。

- All Privileges (すべての権限) → Datastore (データストア) → Allocate Space (領域の割り当て)、Browse datastore (データストアの参照)、Update Virtual Machine files (仮想マシンファイルの更新)、Low level file operations (ローレベルのファイル操作)
- All Privileges (すべての権限) → Network (ネットワーク) → Assign Network (ネットワークの割り当て)
- All Privileges (すべての権限) → Resource (リソース) → Assign virtual machine to resource pool (仮想マシンのリソースプールへの割り当て)
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Change Config (All) (設定の変更 (すべて))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Interaction (All) (対話 (すべて))
- All Privileges (すべての権限) → Virtual Machine (仮想マシン) → Edit Inventory (インベントリ編集)

- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Provisioning (All) (プロビジョニング (すべて))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Guest Operations (All) (ゲストオペレーション (すべて))

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

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

- All Privileges (すべての権限) → Datastore (データストア) → Allocate Space (領域の割り当て)、Browse datastore (データストアの参照)、Update Virtual Machine files (仮想マシンファイルの更新)、Low level file operations (ローレベルのファイル操作)
- All Privileges (すべての権限)→ Network (ネットワーク) → Assign Network (ネットワークの割り当て)
- All Privileges (すべての権限) → Resource (リソース) → Assign virtual machine to resource pool (仮想マシンのリソースプールへの割り当て)
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン) → Configuration (All) (設定 (すべて))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Interaction (All) (対話 (すべて))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン) → Inventory (All) (インベントリ (All))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Provisioning (All) (プロビジョニング (すべて))
- All Privileges (すべての権限)→ Virtual Machine (仮想マシン)→ Guest Operations (All) (ゲストオペレーション (すべて))

11.3. SATELLITE SERVER への VMWARE 接続の追加

この手順を使用して、Satellite Server のコンピュータリソースに VMware vSphere 接続を追加します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

前提条件

- ホストおよびネットワークベースのファイアウォールが、TCP ポート 443 での Satellite Server から vCenter への通信を許可するように設定されていることを確認します。
- Satellite Server と vCenter が互いのホスト名を解決できることを確認します。

手順

1. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、コンピュータリソースのウィンドウで **Create Compute Resource** をクリックします。
2. **Name** フィールドに、リソースの名前を入力します。
3. **Provider** の一覧から **VMware** を選択します。
4. **Description** フィールドには、リソースの説明を入力します。

5. **VCenter/Server** フィールドには、vCenter サーバーの IP アドレスまたはホスト名を入力します。
6. **User** フィールドには、ユーザー名と、vCenter のリソースにアクセスするためのパーミッションを入力します。
7. **Password** フィールドで、ユーザーのパスワードを入力します。
8. **データセンターのロード** をクリックして、VMware vSphere 環境からデータセンターリストを生成します。
9. **Datacenter** リストから、このリストから管理する特定のデータセンターを選択します。
10. **Fingerprint** フィールドは、データセンターからのフィンガープリントが投入されていることを確認します。
11. **Display Type** リストから、**VNC** や **VMRC** などのコンソールタイプを選択します。VNC コンソールは VMware ESXi 6.5 以降ではサポートされません。
12. オプション: **VNC Console Passwords** フィールドでは、**Set a randomly generated password on the display connection** チェックボックスを選択して、無作為に作成されたパスワードで、新規ホストへのアクセスのセキュリティを確保します。**libvirt** ホストからゲスト仮想マシンコンソールにアクセスするための VNC コンソールのパスワードは、以下のコマンドの出力から取得できます。

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

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

13. **Enable Caching** リストから、コンピューティングリソースのキャッシュを有効にするかどうかを選択できます。詳細は、[「コンピュートリソースのキャッシング」](#) を参照してください。
14. **Locations** および **Organizations** タブをクリックして、値が現在のコンテキストに自動的に設定されていることを確認します。また別のコンテキストを追加することもできます。
15. **Submit** をクリックして接続を保存します。

CLI 手順

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

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

11.4. VMWARE イメージを SATELLITE SERVER に追加

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。

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. VMware コンピュートリソースを選択します。
3. **イメージの作成** をクリックします。
4. **名前** フィールドに、イメージの名前を入力します。
5. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
6. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
7. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。デフォルトでは、これは **root** に設定されます。
8. イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**User data** チェックボックスをクリックします。
9. オプション: **Password** フィールドに、イメージにアクセスするための SSH パスワードを入力します。
10. **Image** リストから、VMware のイメージを選択します。
11. **Submit** をクリックしてイメージの詳細を保存します。

CLI 手順

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

```
# hammer compute-resource image create \  
--architecture "My_Architecture" \  
--compute-resource "My_VMware" \  
--name "My_Image" \  
--operatingsystem "My_Operating_System" \  
--username root \  
--uuid "My_UUID"
```

11.5. コンピューティングプロファイルへの VMWARE の詳細の追加

VMware vSphere の仮想マシンの特定のハードウェア設定を事前に定義することができます。これは、これらのハードウェア設定をコンピュータプロファイルに追加することで実行できます。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピューティングプロファイルを選択します。
3. VMware コンピュートリソースを選択します。
4. **CPU** フィールドには、ホストに割り当てる CPU の数を入力します。
5. **1ソケットあたりのコア数** フィールドには各 CPU に割り当てるコアの数を入力します。
6. **メモリー** フィールドには、新規ホストに割り当てるメモリーの容量を MiB 単位で入力します。
7. **ファームウェア** チェックボックスで、ホストのファームウェアとして **BIOS** または **UEFI** を選択します。デフォルトでは、これは **自動** に設定されています。
8. **クラスター** フィールドには、VMware 環境のターゲットホストクラスター名を入力します。
9. **リソースプール** リストから、ホストに対して利用可能なリソース割り当てを選択します。
10. **フォルダー** リストで、ホストを整理するフォルダーを選択します。
11. **ゲスト OS** リストから、VMware vSphere で使用するオペレーティングシステムを選択します。
12. **仮想ハードウェアのバージョン** リストから、仮想マシンに使用する基礎となる VMware ハードウェアの抽象化を定義します。
13. 仮想マシンの電源がオンの状態でメモリーを追加する場合は、**メモリーのホット追加** チェックボックスを選択します。
14. 仮想マシンの電源がオンの状態で CPU を追加する場合は、**CPU のホット追加** チェックボックスを選択します。
15. CD-ROM ドライブを追加する場合は、**CD-ROM ドライブ** のチェックボックスを選択します。
16. **起動順序** リストから、仮想マシンが起動を試行する順序を定義します。
17. オプション: **アノテーションメモ** フィールドに、任意の説明を入力します。
18. イメージベースのプロビジョニングを使用する場合は、イメージリストから **イメージ** を選択します。
19. **SCSI コントローラー** リストから、ホストのディスクアクセスの方法を選択します。
20. Eager Zero シックプロビジョニングを使用する必要がある場合は、**Eager Zero** チェックボックスを選択します。デフォルトでは、ディスクは Lazy Zero シックプロビジョニングを使用します。
21. **ネットワークインターフェイス** リストから、ホストのネットワークインターフェイスのネットワークパラメーターを選択します。少なくとも1つのインターフェイスが Capsule で管理されるネットワークを参照している必要があります。
22. オプション: インターフェイスの **追加** をクリックして、別のネットワークインターフェイスを作成します。
23. **Submit** をクリックしてコンピュータプロファイルを保存します。

CLI 手順

1. コンピュートプロファイルを作成します。

```
# hammer compute-profile create --name "My_Compute_Profile"
```

2. コンピューティングプロファイルへの VMware の詳細の追加

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

11.6. VMWARE でのホストの作成

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

ネットワークベースのプロビジョニングでは、ホストが PXE プロビジョニングサービスにアクセスできるように、新規ホストは VMware vSphere 仮想マシン上にある Satellite Server の統合 Capsule か、外部の Capsule Server にアクセスできる必要があります。この新しいホストエントリーにより、VMware vSphere サーバーが仮想マシンを作成して起動するようにトリガーします。仮想マシンが仮想ネットワークで定義済みの Capsule Server を検出した場合には、仮想マシンは PXE 機能を使用してブートして、選択したオペレーティングシステムのインストールを開始します。

DHCP の競合

プロビジョニング目的で VMware vSphere サーバーで仮想ネットワークを使用する場合は、DHCP 割り当てを提供しない仮想ネットワークを必ず選択します。これにより、新規ホストの起動時に、Satellite Server と DHCP が競合してしまうためです。

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

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、VMware vSphere 接続を選択します。

7. **コンピュータプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
8. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
9. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。VMware は、プロビジョニング中にホストに MAC アドレスを割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
10. インターフェイスウィンドウで、コンピュータプロファイルの設定が入力された VMware 固有のフィールドを確認します。必要に応じてこれらの設定を変更します。
11. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. 必要なプロビジョニング方法を選択します。
 - ネットワークベースのプロビジョニングの場合、**ネットワークベース** をクリックします。
 - イメージベースのプロビジョニングの場合、**イメージベース** をクリックします。
 - ブートディスクベースのプロビジョニングの場合、**ブートディスクベース** をクリックします。
14. **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
15. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュータプロファイルからの情報が入力されていることを確認します。要件に合わせてこれらの設定を変更します。
16. **パラメーター** タブをクリックして、存在するパラメーターでアクティベーションキーが提供されていることを確認します。パラメーターが存在しない場合は、**+ Add Parameter** をクリックします。フィールド **Name** に **kt_activation_keys** と入力します。値 フィールドに、コンテンツホストの登録に使用するアクティベーションキーの名前を入力します。
17. **Submit** をクリックして、VMware でホストをプロビジョニングします。

CLI手順

- **hammer host create** コマンドでネットワークからホストを作成し、**--provision-method build** を指定してネットワークベースのプロビジョニングを使用します。

```
# hammer host create \  
--build true \  

```

```

--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=
true" \
--compute-resource "My_VMware" \
--enabled true \
--hostgroup "My_Host_Group" \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=my
network" \
--location "My_Location" \
--managed true \
--name "My_Host" \
--organization "My_Organization" \
--provision-method build \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"

```

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

```

# hammer host create \
--compute-
attributes="cpus=1,corespersocket=2,memory_mb=1024,cluster=MyCluster,path=MyVMs,start=
true" \
--compute-resource "My_VMware" \
--enabled true \
--hostgroup "My_Host_Group" \
--image "My_VMware_Image" \
--interface
"managed=true,primary=true,provision=true,compute_type=VirtualE1000,compute_network=my
network" \
--location "My_Location" \
--managed true \
--name "My_Host" \
--organization "My_Organization" \
--provision-method image \
--volume="size_gb=20G,datastore=Data,name=myharddisk,thin=true"

```

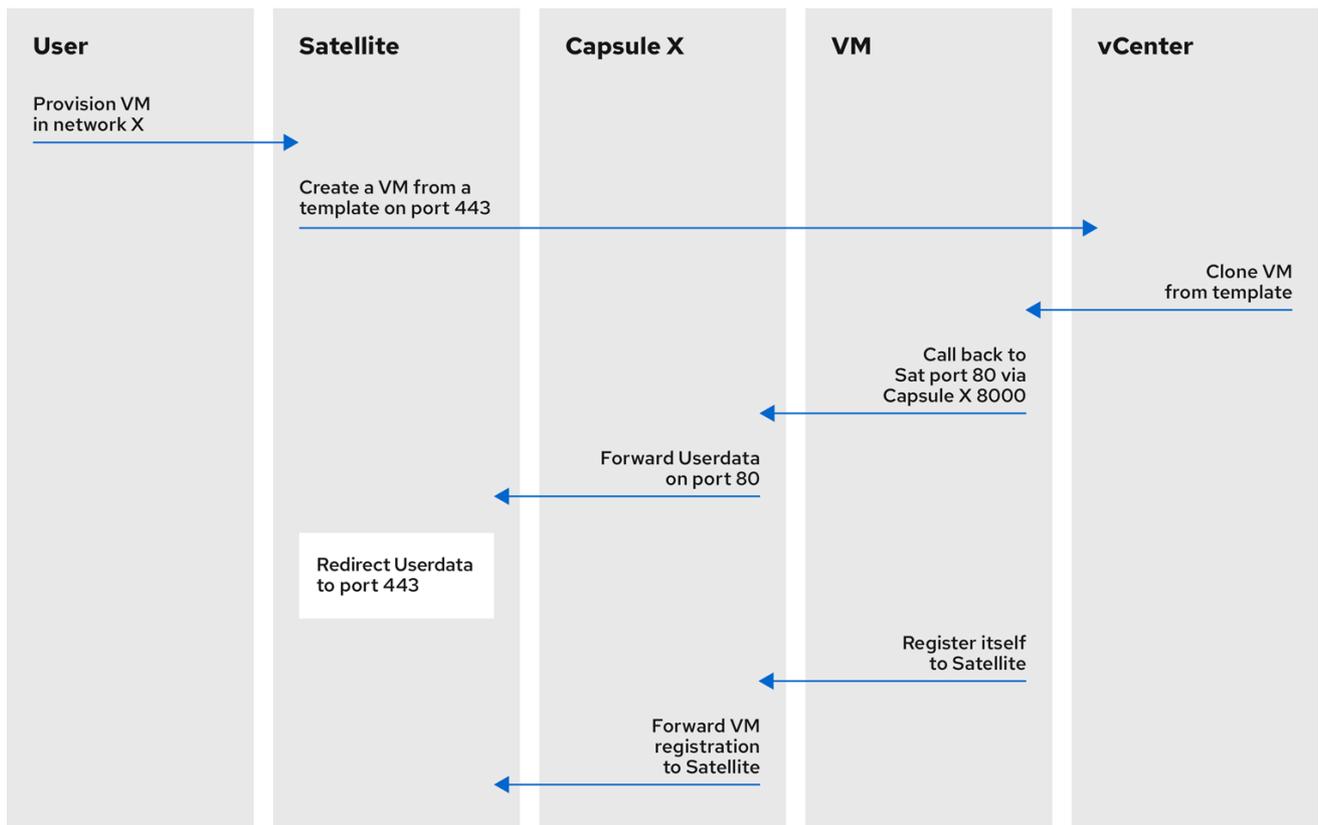
このコンピュータリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

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

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

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

図11.1 VMware cloud-init プロビジョニングの概要



278_Satellite_0922

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

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

前提条件

- 必要な接続ができるように、ポートとファイアウォールが設定されている。**cloud-init** サービスがあるので、仮想マシンを Capsule に登録している場合でさえも、仮想マシンは常に Satellite に対してコールバックします。ポートとファイアウォールの要件に関する詳細は、[オ](#)

オンラインネットワーク環境での [Satellite Server のインストール](#) のポートとファイアウォールの要件 および [Capsule Server のインストール](#) のポートとファイアウォールの要件を参照してください。

- Satellite Server の代わりに Capsule Server を使用する場合は、Capsule Server が適切に設定されていることを確認してください。詳細は、[Capsule Server のインストール](#) の [ホストの登録とプロビジョニングのための Capsule の設定](#) を参照してください。
- 以下の設定ファイルをバックアップします。
 - `/etc/cloud/cloud.cfg.d/01_network.cfg`
 - `/etc/cloud/cloud.cfg.d/10_datasource.cfg`
 - `/etc/cloud/cloud.cfg`

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

1. Satellite Web UI で、**Hosts > Templates > Provisioning Templates** に移動します。
2. **CloudInit default** テンプレートを検索し、その名前をクリックします。
3. **Association** タブをクリックします。
4. テンプレートを適用するすべてのオペレーティングシステムを選択し、**Submit** をクリックします。
5. **UserData open-vm-tools** テンプレートに対して上記の手順を繰り返します。
6. **Hosts > Provisioning Setup > Operating Systems** に移動します。
7. プロビジョニングに使用するオペレーティングシステムを選択します。
8. **Templates** タブをクリックします。
9. **Cloud-init template** リストから **Cloudinit default** を選択します。
10. **User data** テンプレート リストから **UserData open-vm-tools** を選択します。
11. **Submit** をクリックして変更を保存します。

cloud-init テンプレートを使用するためのイメージの準備

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

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

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

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

手順

1. イメージの作成に使用する仮想マシンに、必要なパッケージをインストールします。

```
# dnf install cloud-init open-vm-tools perl-interpreter perl-File-Temp
```

2. **cloud-init** によるネットワーク設定を無効にします。

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

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

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

Capsule Server を通じてプロビジョニングする場合は、**seedfrom** オプションで Capsule Server の URL (**https://capsule.example.com:9090/userdata/** など) を使用します。

4. **cloud-init** で使用するモジュールを設定します。

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

cloud_config_modules:
  - runcmd

cloud_final_modules:
  - scripts-per-once
  - scripts-per-boot
  - scripts-per-instance
  - scripts-user
  - phone-home

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

5. イメージの CA 証明書を有効にします。

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

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

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

Capsule Server を通じてプロビジョニングする場合は、Capsule Server (<https://capsule.example.com/pub/katello-server-ca.crt> など) からファイルをダウンロードします。

7. 証明書のレコードを更新します。

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

8. イメージをクリーンアップします。

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

9. ログスペースの削減、古いログの削除、ログの切り詰めを実行します。

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

10. **udev** ハードウェアルールを削除します。

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

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

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

12. SSH ホストキーを削除します。

```
# rm -f /etc/ssh/ssh_host_*
```

13. root ユーザーの SSH 履歴を削除します。

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

14. root ユーザーの Shell 履歴を削除します。

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

15. この仮想マシンからイメージを作成します。

16. [イメージを Satellite に追加します。](#)

11.8. VMWARE 上の VM の削除

VMware で実行されている VM は、Satellite 内から削除できます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. VMware プロバイダーを選択します。
3. **Virtual Machines** タブで、**Actions** メニューから **Delete** をクリックします。これにより、VMware コンピューティングリソースから仮想マシンが削除されますが、Satellite 内で関連付けられたホストは保持されます。孤立したホストを削除する場合は、**ホスト > すべてのホスト** に移動し、ホストを手動で削除します。

11.9. VMWARE から SATELLITE への仮想マシンのインポート

VMware で実行されている既存の仮想マシンを Satellite にインポートできます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. VMware コンピューティングリソースを選択します。
3. **Virtual Machines** タブで、**Actions** メニューから **Import as managed Host** または **Import as unmanaged Host** をクリックします。以下のページは、コンピュートリソースをすでに選択した状態でホストを作成する場合と同じです。詳細は、ホストの管理の Satellite でのホストの作成を参照してください。
4. **Submit** をクリックして仮想マシンを Satellite にインポートします。

11.10. コンピュートリソースのキャッシング

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

11.10.1. コンピュートリソースのキャッシングの有効化

コンピュートリソースのキャッシュを有効または無効にするには、以下を実行します。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. 更新する VMware サーバーの右側にある **編集** ボタンをクリックします。
3. **キャッシュを有効にする** のチェックボックスを選択します。

11.10.2. コンピュートリソースのキャッシュのリフレッシュ

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. コンピュートリソースのキャッシュをリフレッシュする VMware サーバーを選択し、**Refresh Cache** をクリックします。

CLI 手順

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

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

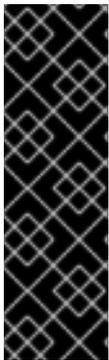
hammer compute-resource list を使用して、コンピュートリソースのキャッシュをリフレッシュする VMware サーバーの ID を判断します。

第12章 OPENSIFT VIRTUALIZATION での仮想マシンのプロビジョニング

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

Satellite を使用すると、OpenShift Virtualization のコンピュートリソースを作成し、Satellite を使用して Kubernetes 仮想マシンをプロビジョニングおよび管理できます。

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



重要

OpenShift Virtualization コンピュートリソースは、テクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポートのサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。これらの機能により、近日発表予定の製品機能をリリースに先駆けてご提供でき、お客様は開発プロセス時に機能をテストして、フィードバックをお寄せいただくことができます。Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

前提条件

- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。
- OpenShift Container Platform 仮想クラスターの **cluster-admin** パーミッションを持つ OpenShift Virtualization ユーザー。詳細は、OpenShift Container Platform ドキュメントの [認証ガイドの RBAC を使用したパーミッションの定義および適用](#) を参照してください。
- OpenShift Virtualization サーバーでネットワークを管理する Capsule Server。Capsule Server との競合を避けるために他の DHCP サービスがこのネットワーク上で実行されていないことを確認します。Capsule Server のネットワークサービス設定の詳細は、[ホストのプロビジョニングのネットワークの設定](#) を参照してください。

関連情報

- 管理者以外のユーザーがホストをプロビジョニングするために必要なパーミッションのリストは、[付録E ホストをプロビジョニングするために必要なパーミッション](#) を参照してください。

12.1. SATELLITE SERVER への OPENSIFT VIRTUALIZATION 接続の追加

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

手順

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

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

2. HTTP と HTTPS 認証に使用するベアータークンを生成します。OpenShift Virtualization サーバーで、トークンが含まれるシークレットを一覧表示します。

```
# kubectl get secrets
```

3. シークレットのトークンをリストします。

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

このトークンを書き留めてください。この手順の後半で使用します。

4. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、**Create Compute Resource** をクリックします。
5. **Name** フィールドに、新しいコンピュートリソースの名前を入力します。
6. **Provider** リストから **OpenShift Virtualization** を選択します。
7. **Description** フィールドに、コンピュートリソースの説明を入力します。
8. **Hostname** フィールドに、使用する OpenShift Virtualization サーバーの FQDN、ホスト名、または IP アドレスを入力します。
9. **API Port** フィールドには、Satellite から OpenShift Virtualization へのプロビジョニング要求に使用するポート番号を入力します。
10. **Namespace** フィールドには、使用する OpenShift Virtualization 仮想クラスターのユーザー名を入力します。
11. **Token** フィールドに、HTTP および HTTPS 認証用のベアータークンを入力します。
12. オプション: **X509 認証局** フィールドに、API サーバー呼び出しのクライアントの証明書認証を有効にする証明書を入力します。

第13章 RED HAT OPENSTACK PLATFORM でのクラウドインスタンスのプロビジョニング

Red Hat OpenStack Platform は、プライベートまたはパブリックの Infrastructure-as-a-Service (IaaS) クラウドを構築するための基盤を提供します。これにより、スケーラビリティが極めて高く、耐障害性に優れたプラットフォームをクラウド対応のワークロード開発に利用できます。Satellite では、Red Hat OpenStack Platforms REST API と対話し、クラウドインスタンスを作成して、電源管理の状態を制御することができます。

前提条件

- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#)を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#)を参照してください。
- OpenStack 環境でネットワークを管理する Capsule Server。詳細は、[ホストのプロビジョニングのネットワークの設定](#)を参照してください。
- イメージベースのプロビジョニング用に OpenStack Image Storage (glance) サービスに追加されたイメージ。詳細は、[Red Hat OpenStack Platform Instances and Images Guide](#)を参照してください。

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. **Create Compute Resource** をクリックします。
3. **Name** フィールドに、新しいコンピュートリソースの名前を入力します。
4. **プロバイダー** リストから **RHEL OpenStack Platform** を選択します。
5. オプション: **Description** フィールドには、コンピュートリソースの説明を入力します。
6. **URL** フィールドには、**http://openstack.example.com:5000/v2.0/tokens** または **http://openstack.example.com:5000/v3/auth/tokens** などの **tokens** リソースにおける OpenStack 認証 keystone サービスの API の URL を入力します。
7. **Username** フィールドおよび **Password** フィールドに、環境にアクセスするための Satellite のユーザー認証を入力します。
8. オプション: **Project (Tenant) name** フィールドに、Satellite Server が管理するテナント (v2) またはプロジェクト (v3) の名前を入力します。
9. **User domain** フィールドに v3 認証のユーザードメインを入力します。
10. **Project domain name** フィールドに v3 認証のプロジェクトドメイン名を入力します。

11. **Project domain ID** フィールドに、v3 認証のプロジェクトドメイン ID を入力します。
12. オプション: **Allow external network as main network** を選択して、ホストのプライマリーネットワークとして外部ネットワークを使用します。
13. オプション: **Test Connection** をクリックして、Satellite がコンピュートリソースに接続できることを確認します。
14. **ロケーション** および **組織** タブをクリックして、使用するロケーションと組織が現在のコンテキストに自動的に設定されていることを確認します。他のコンテキストは、これらのタブに追加します。
15. **Submit** をクリックして Red Hat OpenStack Platform の接続を保存します。

CLI 手順

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

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

13.2. SATELLITE SERVER への RED HAT OPENSTACK PLATFORM イメージの追加

イメージベースのプロビジョニングを使用してホストを作成するには、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

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

手順

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

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

CLI 手順

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

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

13.3. RED HAT OPENSTACK PLATFORM の詳細のコンピュートプロファイルへの追加

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Red Hat OpenStack Platform コンピュートリソースの名前をクリックします。
4. **Flavor** リストから、ホストに使用する Red Hat OpenStack Platform のハードウェアのプロファイルを選択します。
5. **Availability zone** リストから Red Hat OpenStack Platform 環境内で使用するターゲットクラスターを選択します。
6. **Image** リストから、イメージベースのプロビジョニングに使用するイメージを選択します。
7. **テナント** リストから、Red Hat OpenStack Platform インスタンスのテナントまたはプロジェクトを選択します。
8. **セキュリティーグループ** リストから、ポートおよび IP アドレスのクラウドベースのアクセスルールを選択します。
9. **内部ネットワーク** から、ホストが参加するプライベートネットワークを選択します。

10. **Floating IP ネットワーク** から、ホストが参加する外部ネットワークを選択して、Floating IP アドレスを割り当てます。
11. **ボリュームからの起動** から、イメージからボリュームが作成されるかを設定します。これが選択されていない場合には、インスタンスはイメージを直接起動します。
12. **新規起動ボリュームサイズ (GB)** フィールドには、新規起動ボリュームのサイズ (GB) を入力します。
13. **Submit** をクリックしてコンピュータプロファイルを保存します。

CLI 手順

- コンピュータプロファイルに Red Hat OpenStack Platform の詳細を設定します。

```
# hammer compute-profile values create
--compute-resource "My_Laptop" \
--compute-profile "My_Compute_Profile" \
--compute-attributes
"availability_zone=My_Zone,image_ref=My_Image,flavor_ref=m1.small,tenant_id=openstack,
security_groups=default,network=My_Network,boot_from_volume=false"
```

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

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

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、Red Hat OpenStack Platform 接続を選択します。
7. **コンピュータプロファイル** リストから、仮想マシンの設定を自動的に投入するために使用するプロファイルを選択します。
8. **ライフサイクル環境** リストから、環境を選択します。
9. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。

10. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。Red Hat OpenStack Platform は、プロビジョニング中に MAC アドレスをホストに割り当てます。
 - **ホスト タブの名前は DNS 名** になります。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
11. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. コンピュートプロファイルから自動的にデータが投入されるイメージを変更するには、新規ホストの root ボリュームをベースとする別のイメージを **イメージ** リストから選択します。
14. **プロビジョニングテンプレート** で **解決** をクリックし、新規ホストから使用する適切なプロビジョニングテンプレートを特定できることを確認します。
15. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュートプロファイルからの情報が入力されていることを確認します。必要に応じてこれらの設定を変更します。
16. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
17. **Submit** をクリックしてホストエントリーを保存します。

CLI 手順

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

```
# hammer host create \
--compute-
attributes="flavor_ref=m1.small,tenant_id=openstack,security_groups=default,network=mynetwork" \
--compute-resource "My_OpenStack_Platform" \
--enabled true \
--hostgroup "My_Host_Group" \
--image "My_OpenStack_Image" \
--interface "managed=true,primary=true,provision=true" \
--location "My_Location" \
--managed true \
--name "My_Host_Name" \
--organization "My_Organization" \
--provision-method image
```

このコンピュートリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

第14章 AMAZON EC2 でのクラウドインスタンスのプロビジョニング

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

14.1. AMAZON EC2 プロビジョニングの要件

Amazon EC2 プロビジョニングの要件には以下が含まれます。

- EC2 環境でネットワークを管理する Capsule Server。ホストと Capsule Server 間のネットワークのセキュリティーを確保するために、Virtual Private Cloud (VPC) を使用します。
- イメージベースのプロビジョニング用の Amazon Machine Image (AMI)。
- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使うことができます。詳細は、[コンテンツの管理のリポジトリの同期](#) を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#) を参照してください。

14.2. AMAZON EC2 プラグインのインストール

Amazon EC2 プラグインをインストールして、EC2 コンピューティングリソースプロバイダーを Satellite にアタッチします。これにより、ホストを管理して EC2 にデプロイできます。

手順

1. EC2 コンピューティングリソースプロバイダーを Satellite Server にインストールします。

```
# satellite-installer --enable-foreman-compute-ec2
```

2. オプション: Satellite Web UI で、**Administer > About** に移動し、**compute resources** タブを選択して、Amazon EC2 プラグインのインストールを確認します。

14.3. SATELLITE SERVER への AMAZON EC2 接続の追加

この手順を使用して、Satellite Server のコンピュートリソースに Amazon EC2 接続を追加します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

前提条件

- この手順を実行する AWS EC2 ユーザーには、**AmazonEC2FullAccess** パーミッションが必要です。これらのパーミッションは AWS から割り当てることができます。

時間設定および Amazon Web Services

Amazon Web Services は、認証プロセスの一環として時間の設定を使用します。Satellite Server の時間が正しく同期されていることを確認します。さらに、**ntpd** または **chronyd** などの NTP サービスが Satellite Server で正しく実行されていることを確認します。Amazon Web Services に正しい時間を指定できないと、認証に失敗する可能性があります。

Satellite での時間の同期に関する詳細は、[オンラインネットワーク環境での Satellite Server のインストールの時間の同期](#) を参照してください。

手順

1. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、コンピュートリソースのウィンドウで **Create Compute Resource** をクリックします。
2. **名前** フィールドには、Amazon EC2 コンピュートリソースを識別するための名前を入力します。
3. **プロバイダー** のリストから **EC2** を選択します。
4. **説明** フィールドには、今後使用する時にリソースを特定できるように名前を入力します。
5. オプション: **HTTP プロキシ** リストから、HTTP プロキシを選択して、外部の API サービスに接続します。HTTP プロキシを Satellite に追加してから、リストからプロキシを選択する必要があります。詳細は、[「コンピュートリソースと HTTP プロキシとの連携」](#) を参照してください。
6. **アクセスキー** と **シークレットキー** フィールドで、Amazon EC2 アカウントのアクセスキーを入力します。詳細情報は、Amazon ドキュメントの Web サイトで [Managing Access Keys for your AWS Account](#) を参照してください。
7. オプション: **Load Regions** をクリックして、**Region** リストにデータを入力します。
8. **リージョン** リストから、使用する Amazon EC2 リージョンとデータセンターを選択します。
9. **ロケーション** タブをクリックして、使用するロケーションが選択されていることを確認します。あるいは、別のロケーションを追加します。
10. **組織** タブをクリックして、使用する組織が選択されていることを確認します。あるいは、別のロケーションを追加します。
11. **Submit** をクリックして Amazon EC2 接続を保存します。
12. 新規コンピュートリソースを選択して、**SSH キー** タブをクリックし、**ダウンロード** をクリックして、SSH 認証に使用するために SSH キーのコピーを保存します。[BZ1793138](#) が解決するまで、Amazon EC2 コンピュートリソースのコピーの作成直後に SSH キーのコピーをダウンロードします。後で SSH キーが必要な場合には、[「SSH での Amazon EC2 インスタンスへの接続」](#) の手順を実行してください。

CLI 手順

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

```
# hammer compute-resource create \  
--description "Amazon EC2 Public Cloud" \  
--locations "My_Location" \  
--name "My_EC2_Compute_Resource" \  
--organizations "My_Organization" \  
--password "My_Secret_Key" \  
--provider "EC2" \  
--region "My_Region" \  
--user "My_User_Name"
```

14.4. コンピュートリソースと HTTP プロキシとの連携

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

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

手順

1. Satellite Web UI で、**Infrastructure > HTTP Proxies** に移動して、**New HTTP Proxy** を選択します。
2. **名前** フィールドで、HTTP プロキシの名前を入力します。
3. **URL** フィールドに、ポート番号を含む HTTP プロキシの URL を入力します。
4. オプション: HTTP プロキシに認証が必要な場合には、HTTP プロキシに対する認証に使用するユーザーとパスワードを入力します。
5. **テスト接続** をクリックして Satellite から HTTP プロキシに対する接続できることを確認します。
6. **ロケーション** タブで、ロケーションを追加します。
7. **Organization** タブをクリックして、組織を追加します。
8. **Submit** をクリックします。

14.5. AMAZON EC2 のイメージの作成

Satellite 内から Amazon EC2 のイメージを作成できます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. Amazon EC2 プロバイダーを選択します。
3. **イメージの作成** をクリックします。
 - **Name** フィールドに、EC2 イメージの意味のある一意の名前を入力します。
 - **Operating System** リストから、イメージに関連付けるオペレーティングシステムを選択します。
 - **Architecture** リストから、イメージに関連付けるアーキテクチャーを選択します。
 - **Username** フィールドに、マシンへの SSH 接続に必要なユーザー名を入力します。
 - **Image ID** フィールドに、Amazon またはオペレーティングシステムベンダーから提供されたイメージ ID を入力します。

- オプション: **User Data** チェックボックスを選択して、ユーザーデータ入力をサポートを有効にします。
- オプション: このイメージの作成時に使用する **Fog** の **Iam Role** を設定します。
- **Submit** をクリックして、変更を **Satellite** に保存します。

14.6. SATELLITE SERVER への AMAZON EC2 イメージの追加

Amazon EC2 はイメージベースのプロビジョニングを使用して新規ホストを作成します。イメージの詳細を **Satellite Server** に追加する必要があります。これにはアクセスの詳細およびイメージのロケーションが含まれます。

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

手順

1. **Satellite Web UI** で、**Infrastructure > Compute Resources** に移動し、**Amazon EC2 接続** を選択します。
2. **Images** タブをクリックし、**Create Image** をクリックします。
3. **名前** フィールドには、今後使用する時にイメージを特定できるように名前を入力します。
4. **オペレーティングシステム** リストから、追加するイメージに対応するオペレーティングシステムを選択します。
5. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための **SSH** ユーザー名を入力します。通常、これは **root** ユーザーになります。
7. **パスワード** フィールドには、イメージにアクセスするための **SSH** パスワードを入力します。
8. **イメージ ID** フィールドには、イメージの **Amazon Machine Image (AMI)** ID を入力します。通常、この形式は **ami-xxxxxxx** になります。
9. オプション: イメージが **cloud-init** データなどのユーザーデータ入力をサポートする場合には、**ユーザーデータ** チェックボックスを選択します。ユーザーデータを有効にすると、**Finish** スクリプトは自動的に無効になります。これは、逆の場合にも当てはまります。**Finish** スクリプトを有効にすると、ユーザーデータが無効になります。
10. オプション: **IAM ロール** リストから、イメージの作成に使用する **Amazon** のセキュリティロールを入力します。
11. **Submit** をクリックしてイメージの詳細を保存します。

CLI 手順

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

```
# hammer compute-resource image create \
--architecture "My_Architecture" \
--compute-resource "My_EC2_Compute_Resource" \
--name "My_Amazon_EC2_Image" \
```

```
--operatingsystem "My_Operating_System" \
--user-data true \
--username root \
--uuid "ami-My_AMI_ID"
```

14.7. AMAZON EC2 の詳細のコンピュータプロファイルへの追加

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

手順

1. Satellite Web UI で **インフラストラクチャー > コンピュータプロファイル** に移動して、プロファイルの名前をクリックして EC2 接続の名前をクリックします。
2. **フレーバー** リストから、ホストに使用する EC2 のハードウェアのプロファイルを選択します。
3. **Image** リストから、イメージベースのプロビジョニングに使用するイメージを選択します。
4. **アベイラビリティゾーン** リストから、EC2 リージョン内で使用するターゲットクラスターを選択します。
5. **サブネット** リストから EC2 インスタンスのサブネットを追加します。新規ホストのプロビジョニング用の VPC がある場合は、そのサブネットを使用します。
6. **セキュリティグループ** リストから、ホストに適用するポートおよび IP アドレスのクラウドベースのアクセスルールを定義します。
7. **管理 IP** リストから、**パブリック IP** または **プライベート IP** のいずれかを選択します。
8. **Submit** をクリックしてコンピュータプロファイルを保存します。

CLI 手順

- コンピュータプロファイルに Amazon EC2 の詳細を設定します。

```
# hammer compute-profile values create
--compute-resource "My_Laptop" \
--compute-profile "My_Compute_Profile" \
--compute-attributes "flavor_id=1,availability_zone=
My_Zone,subnet_id=1,security_group_ids=1,managed_ip=public_ip"
```

14.8. AMAZON EC2 でのイメージベースのホストの作成

Amazon EC2 プロビジョニングプロセスでは、Amazon EC2 サーバーで既存イメージからホストを作成します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。

4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、EC2 接続を選択します。
7. **コンピュートプロファイル** リストから、仮想マシンベースの設定を自動的に投入するために使用するプロファイルを選択します。
8. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
9. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - **Satellite** は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。EC2 は、プロビジョニング中にホストに MAC アドレスを割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - **Satellite** が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
10. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
11. **オペレーティングシステム** タブをクリックして、全フィールドに値が入力されていることを確認します。
12. **仮想マシン** タブをクリックして、全フィールドに値が投入されていることを確認します。
13. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
14. **Submit** をクリックして変更を保存します。

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

CLI 手順

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

```
# hammer host create \
--compute-attributes="flavor_id=m1.small,image_id=TestImage,availability_zones=us-east-1a,security_group_ids=Default,managed_ip=Public" \
--compute-resource "My_EC2_Compute_Resource" \
--enabled true \
--hostgroup "My_Host_Group" \
--image "My_Amazon_EC2_Image" \
--interface "managed=true,primary=true,provision=true,subnet_id=EC2" \
```

```
--location "My_Location" \  
--managed true \  
--name "My_Host_Name_" \  
--organization "My_Organization" \  
--provision-method image
```

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

14.9. SSH での AMAZON EC2 インスタンスへの接続

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

手順

1. Satellite Server ベースシステムで、コンピュータリソースリストの場所を確認するには、以下のコマンドを入力し、使用するコンピュータリソースの ID を書き留めます。

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

2. ユーザーを **postgres** ユーザーに切り替えます。

```
# su - postgres
```

3. **postgres** シェルを開始します。

```
$ psql
```

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

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

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

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

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

7. **.pem** ファイルを作成し、ファイルにキーを貼り付けます。

```
# vim Keyname.pem
```

8. **.pem** ファイルへのアクセスを制限するよう確認します。

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

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

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

14.10. AMAZON WEB SERVICE EC2 環境向け FINISH テンプレートの設定

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

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

手順

1. Satellite Web UI で、**Hosts > Provisioning Templates** に移動します。
2. **プロビジョニングテンプレート** ページの検索フィールドに **Kickstart default finish** を入力し、**検索** をクリックします。
3. **Kickstart default finish** テンプレートで、**クローン** を選択します。
4. **名前** フィールドに、テンプレート向けに独自の名前を入力します。
5. テンプレートで、**subscription-manager register** コマンドおよび **yum** コマンド以外の root 権限が必要な各コマンドを **sudo** で指定します。または、以下の行を追加してテンプレート全体を sudo ユーザーとして実行します。

```
sudo -s << EOS
__Template__Body__
EOS
```

6. **関連付け** タブをクリックし、使用する Red Hat Enterprise Linux オペレーティングシステムとテンプレートを関連付けします。
7. **ロケーション** タブをクリックして、ホストがある場所を追加します。
8. **Organizations** タブをクリックして、ホストが属する組織を追加します。
9. 必要なカスタマイズまたは変更を追加したら、**Submit** をクリックしてテンプレート保存します。
10. Satellite Web UI で、**Hosts > Operating systems** に移動し、ホストに使用するオペレーティングシステムを選択します。
11. **Templates** タブをクリックし、**Finish Template** リストから、finish テンプレートを選択します。
12. Satellite Web UI で、**Hosts > Create Host** に移動します。
13. **Name** フィールドには、ホストの名前を入力します。
14. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。

15. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
16. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
17. **Parameters** タブをクリックし、**Host parameters** に移動します。
18. **Host parameters** で、**Add Parameter** を 2 回クリックし、新しいパラメーターフィールドを 2 つ追加します。以下のパラメーターを追加します。
 - a. **Name** フィールドで、**activation_keys** を入力します。対応する **Value** フィールドで、アクティベーションキーを入力します。
 - b. **Name** フィールドで、**remote_execution_ssh_user** を入力します。対応する **Value** フィールドで、**ec2-user** を入力します。
19. **Submit** をクリックして変更を保存します。

14.11. AMAZON EC2 での仮想マシンの削除

Amazon EC2 で実行している仮想マシンを、Satellite 内から削除できます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. Amazon EC2 プロバイダーを選択します。
3. **Virtual Machines** タブで、**Actions** メニューから **Delete** をクリックします。これにより、Amazon EC2 コンピューティングリソースから仮想マシンが削除されますが、Satellite 内の関連するホストは保持されます。孤立したホストを削除する場合は、**Hosts > All Hosts** に移動し、ホストを手動で削除します。

14.12. AMAZON WEB SERVICES および SATELLITE に関する詳細情報

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

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

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

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

Satellite は、Google Compute Engine (GCE) と対話して、新規仮想マシンの作成、電源管理の状態の制御などを実行できます。

GCE ホストの作成には、Satellite では Red Hat がサポートするゴールデンイメージのみを使用できません。

前提条件

- Satellite でドメインとサブネットを設定する。ネットワーク要件の詳細は、[3章 ネットワークの設定](#)を参照してください。
- Red Hat Enterprise Linux の同期済みのコンテンツリポジトリを使用できる。詳細は、[コンテンツの管理のリポジトリの同期](#)を参照してください。
- ホスト登録用のアクティベーションキーを提供します。詳細は、[コンテンツの管理のアクティベーションキーの作成](#)を参照してください。
- GCE プロジェクトで、必要な IAM Compute ロールでサービスアカウントを設定します。詳細は、GCE ドキュメントの [Compute Engine IAM roles](#) を参照してください。
- GCE プロジェクト全体のメタデータで、**enable-oslogin** を **FALSE** に設定します。詳細は、GCE ドキュメントの [Enabling or disabling OS Login](#) を参照してください。
- オプション: GCE ホストで Puppet を使用するには、**管理 > 設定 > Puppet** に移動して、**証明書での UUID の使用** 設定を有効にして、Puppet が一貫性のある Puppet 証明書 ID を使用するよう設定します。
- ニーズに合わせて **finish** または **user_data** プロビジョニングテンプレートと使用するオペレーティングシステムを関連付けます。詳細は、ホストのプロビジョニングの [プロビジョニングテンプレート](#) を参照してください。

15.1. SATELLITE SERVER への GOOGLE GCE 接続の追加

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

手順

1. Google GCE で、サービスアカウントキーを JSON 形式で生成します。詳細は、GCE ドキュメントの [サービスアカウントキーの作成と管理](#)を参照してください。
2. Satellite Web UI で **Infrastructure > Compute Resources** に移動して、**Create Compute Resource** をクリックします。
3. **名前** フィールドには、コンピュートリソースの名前を入力します。
4. **プロバイダー** のリストから **Google** を選択します。
5. オプション: **説明** フィールドに、リソースの説明を入力します。
6. **JSON key** フィールドで、**Choose File** をクリックし、ローカルマシンからアップロードするサービスアカウントキーを見つけます。

7. **ゾーンのロード** をクリックして GCE 環境からゾーンリストを生成します。
8. **ゾーン** リストから使用する GCE ゾーンを選択します。
9. **Submit** をクリックします。

CLI 手順

1. Google GCE で、サービスアカウントキーを JSON 形式で生成します。詳細は、GCE ドキュメントの [サービスアカウントキーの作成と管理](#) を参照してください。
2. ファイルをローカルマシンから Satellite Server にコピーします。

```
# scp My_GCE_Key.json root@satellite.example.com:/etc/foreman/My_GCE_Key.json
```

3. Satellite Server で、サービスアカウントキーの所有者を **foreman** ユーザーに変更します。

```
# chown root:foreman /etc/foreman/My_GCE_Key.json
```

4. Satellite Server で、サービスアカウントキーのアクセス許可を設定して、ファイルが読み取り可能であることを確認します。

```
# chmod 0640 /etc/foreman/My_GCE_Key.json
```

5. Satellite Server で、サービスアカウントキーの SELinux コンテキストを復元します。

```
# restorecon -vv /etc/foreman/My_GCE_Key.json
```

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

```
# hammer compute-resource create \  
--key-path "/etc/foreman/My_GCE_Key.json" \  
--name "My_GCE_Compute_Resource" \  
--provider "gce" \  
--zone "My_Zone"
```

15.2. SATELLITE SERVER への GOOGLE COMPUTE ENGINE イメージの追加

イメージベースのプロビジョニングを使用してホストを作成するには、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動し、Google Chrome Engine 接続の名前をクリックします。
2. **イメージの作成** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。

4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。**root** ユーザーは SSH キーを使用して GCE インスタンスに接続できないので、**root** 以外のユーザーを指定してください。ユーザー名は文字で始まり、小文字と数字で設定されている必要があります。
7. **イメージ** リストの Google Chrome Engine コンピュートリソースからイメージを選択します。
8. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
9. **Submit** をクリックしてイメージの詳細を保存します。

CLI 手順

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

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

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

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピュートプロファイルウィンドウで、既存のコンピュートプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. GCE コンピュートリソースの名前をクリックします。
4. **マシンタイプ** リストからプロビジョニングに使用するマシンタイプを選択します。
5. **イメージ** リストから、プロビジョニングに使用するイメージを選択します。

6. **ネットワーク** リストから、プロビジョニングに使用する Google GCE ネットワークを選択します。
7. オプション: **一時的な外部 IP の関連付け** チェックボックスを選択して、動的で一時的な IP アドレスを割り当て、Satellite がホストと通信に使用できるようにします。ホストを再起動すると、このパブリック IP アドレスは変わります。永続的な IP アドレスが必要な場合には、Google GCE で静的なパブリック IP アドレスを予約し、ホストにアタッチします。
8. **サイズ (GB)** フィールドに、ホストで作成するストレージのサイズを入力します。
9. **Submit** をクリックしてコンピュータプロファイルを保存します。

CLI 手順

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

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

2. GCE の情報をコンピュータプロファイルに追加します。

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

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

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

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、Google Compute Engine 接続を選択します。
7. **コンピュータプロファイル** リストから、仮想マシンの設定を自動的に投入するために使用するプロファイルを選択します。
8. **ライフサイクル環境** リストから、環境を選択します。

9. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
10. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。Google Compute Engine サーバーは MAC アドレスをホストに割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。
 - **ドメイン** フィールドに必要なドメインの情報が入力されていること。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
11. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
13. **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。
14. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュープロファイルからの情報が入力されていることを確認します。必要に応じてこれらの設定を変更します。
15. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
16. **Submit** をクリックしてホストエントリーを保存します。

CLI 手順

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

```
# hammer host create \
--architecture x86_64 \
--compute-profile "My_Compute_Profile" \
--compute-resource "My_Compute_Resource" \
--image "My_GCE_Image" \
--interface "type=interface,domain_id=1,managed=true,primary=true,provision=true" \
--location "My_Location" \
--name "My_Host_Name" \
--operatingsystem "My_Operating_System" \
--organization "My_Organization" \
--provision-method "image" \
--puppet-ca-proxy-id My_Puppet_CA_Proxy_ID \
--puppet-environment-id My_Puppet_Environment_ID \
--puppet-proxy-id My_Puppet_Proxy_ID \
--root-password "My_Root_Password"
```

このコンピュータリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

15.5. GOOGLE GCE での VM の削除

Satellite Server の Google GCE で実行されている VM を削除できます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動します。
2. Google GCE プロバイダーを選択します。
3. **Virtual Machines** タブで、**Actions** メニューから **Delete** をクリックします。これにより、Google GCE コンピューティングリソースから仮想マシンが削除されますが、Satellite 内の関連するホストは保持されます。孤立したホストを削除する場合は、**ホスト > すべてのホスト** に移動し、ホストを手動で削除します。

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

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

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

前提条件

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

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

この手順を使用して、Satellite でコンピューティングリソースとして Microsoft Azure を追加します。使用する Microsoft Azure リージョンごとに個別のコンピューティングリソースを追加する必要があります。ことに注意してください。

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

手順

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

CLI 手順

- **hammer compute-resource create** を使用して Satellite に Azure コンピュートリソースを追加します。

```
# hammer compute-resource create \
--app-ident My_Client_ID \
--name My_Compute_Resource_Name \
--provider azurearm \
--region "My_Region" \
--secret-key My_Client_Secret \
--sub-id My_Subscription_ID \
--tenant My_Tenant_ID
```

--region オプションの値には、小文字を使用する必要があり、特殊文字を含めることはできません。



重要

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

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

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

イメージベースのプロビジョニングを使用してホストを作成するには、アクセスの情報およびイメージの場所など、イメージの情報を Satellite Server に追加する必要があります。

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources** に移動し、Microsoft Azure Resource Manager 接続の名前をクリックします。
2. **イメージの作成** をクリックします。
3. **名前** フィールドに、イメージの名前を入力します。
4. **オペレーティングシステム** リストから、イメージのベースオペレーティングシステムを選択します。
5. **Architecture** リストから、オペレーティングシステムのアーキテクチャーを選択します。
6. **ユーザー名** フィールドには、イメージにアクセスするための SSH ユーザー名を入力します。**root** ユーザーは使用できません。
7. オプション: **パスワード** フィールドに認証に使用するパスワードを入力します。
8. **Azure Image Name** フィールドでイメージ名を **prefix://UUID** 形式で入力します。
 - カスタムのイメージの場合は **custom** のプリフィックスを使用します。(例: **custom://image-name**)
 - 共有ギャラリーのイメージの場合には **gallery** のプリフィックスを使用します。(例: **gallery://image-name**)
 - パブリックまたは RHEL Bring Your Own Subscription (BYOS) イメージの場合には、**marketplace** のプリフィックスを使用します。(例: **marketplace://OpenLogicCentOS:7.5:latest**)
詳細は、[Find Linux VM images in the Azure Marketplace with the Azure CLI](#) を参照してください。
9. オプション: イメージで **cloud-init** データなどのユーザーデータ入力をサポートさせるには、**ユーザーデータ** チェックボックスを選択します。
10. **Submit** をクリックしてイメージの詳細を保存します。

CLI 手順

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

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

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

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

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

手順

1. Satellite Web UI で、**Infrastructure > Compute Profiles** に移動します。
2. コンピューティングプロファイルウィンドウで、既存のコンピューティングプロファイル名をクリックするか、**Create Compute Profile** をクリックするか、**Name** を入力して **Submit** をクリックします。
3. Azure コンピューティングリソースの名前をクリックします。
4. **リソースグループ** リストから、プロビジョニングするリソースグループを選択します。
5. **VM サイズ** リストからプロビジョニングする仮想マシンのサイズを選択します。
6. **プラットフォーム** リストから **Linux** を選択します。
7. **Username** フィールドに、認証に使用するユーザー名を入力します。コンピューティングプロファイルに入力するユーザー名は、イメージの作成時に使用したのと同じものでなければなりません。
8. ユーザーを認証するには、以下のオプションのいずれかを使用します。
 - パスワードを使用した認証は、**パスワード** フィールドにパスワードを入力します。
 - SSH キーを使用した認証は、**SSH Key** フィールドに SSH キーを入力します。
9. オプション **I**: 仮想マシンに、プレミアム仮想マシンディスクを使用させる場合には、**プレミアム OS ディスク** のチェックボックスを選択します。
10. **OS ディスクのキャッシュ** リスからディスクのキャッシュ設定を選択します。
11. オプション **C**: **カスタムスクリプトコマンド** フィールドに、仮想マシンのプロビジョニング時に仮想マシンで実行するコマンドを入力します。

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



重要

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

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

CLI 手順

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

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

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

```
# hammer compute-profile values create \  
--compute-  
attributes="resource_group=resource_group,vm_size=Standard_B1s,username=azure_us  
er,password=azure_password,platform=Linux,script_command=touch /var/tmp/text.txt" \  
--compute-profile "compute_profile_name" \  
--compute-resource azure_cr_name \  
--
```

```
interface="compute_public_ip=Dynamic,compute_network=mysubnetID,compute_private_ip=false" \  
--volume="disk_size_gb=5,data_disk_caching=None"
```

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

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

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

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

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

手順

1. Satellite Web UI で、**Hosts > Create Host** に移動します。
2. **Name** フィールドには、ホストの名前を入力します。
3. オプション: **Organization** タブをクリックし、要件に合わせて組織コンテキストを変更します。
4. オプション: **Location** タブをクリックし、要件に合わせてロケーションコンテキストを変更します。
5. **Host Group** リストから、ホストを割り当てるホストグループを選択します。そのホストグループがフォームに入力されます。
6. **デプロイ先** リストから、Microsoft Azure Resource Manager 接続を選択します。
7. **コンピュートプロファイル** リストから、仮想マシンの設定を自動的に投入するために使用するプロファイルを選択します。
8. **ライフサイクル環境** リストから、環境を選択します。
9. **Interfaces** タブをクリックし、ホストのインターフェイスで **Edit** をクリックします。
10. フィールドに値が投入されていることを確認します。特に以下に注意してください。
 - Satellite は新規ホストの IP アドレスを自動的に割り当てます。
 - **MAC アドレス** フィールドが空白であることを確認します。Microsoft Azure Resource Manager サーバーは、プロビジョニング中に MAC アドレスをホストに割り当てます。
 - **ホスト** タブの **名前** は **DNS 名** になります。

- **Azure サブネット** フィールドに必要な Azure サブネットの情報が入力されていること。
 - オプション: 仮想マシンで **IPv4 Subnet** リストから静的なプライベート IP アドレスを使用する場合には、Azure サブネットのアドレスと一致するように、**ネットワークアドレス** フィールドで、Satellite のサブネットを選択してください。**IPv4 アドレス** フィールドに、Azure サブネットの範囲内の IPv4 アドレスを入力します。
 - Satellite が、ホストの最初のインターフェイスに **Managed**、**Primary**、および **Provision** オプションを自動選択していることを確認します。選択されていない場合は、それらを選択してください。
11. **OK** をクリックして保存します。別のインターフェイスを追加するには、**インターフェイスの追加** をクリックします。**プロビジョニング** および **プライマリー** には、インターフェイスは1つしか選択できません。
 12. **オペレーティングシステム** タブをクリックして、全フィールドに値が自動的に含まれていることを確認します。
 13. **Provisioning Method** には **Image Based** が選択されているようにします。
 14. **イメージ** リストから、プロビジョニングに使用する Azure Resource Manager イメージを選択します。
 15. **Root パスワード** フィールドで、認証に使用する root パスワードを入力します。
 16. **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。
 17. **仮想マシン** タブをクリックして、設定には、ホストグループおよびコンピュートプロファイルからの情報が入力されていることを確認します。必要に応じてこれらの設定を変更します。
 18. **パラメーター** タブをクリックして、パラメーターが存在し、そのパラメーターでアクティベーションキーが指定されていることを確認します。存在しない場合には、アクティベーションキーを追加します。
 19. **Submit** をクリックしてホストエントリーを保存します。

CLI 手順

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

```
# hammer host create \
--architecture x86_64 \
--compute-profile "My_Compute_Profile" \
--compute-resource "My_Compute_Resource" \
--domain "My_Domain" \
--image "My_Azure_Image" \
--location "My_Location" \
--name "My_Host_Name" \
--operatingsystem "My_Operating_System" \
--organization "My_Organization" \
--provision-method "image"
```

このコンピュートリソースに対する他のホスト作成パラメーターの情報は、**hammer host create --help** コマンドを入力します。

16.5. MICROSOFT AZURE 上の VM の削除

Microsoft Azure で実行されている VM は、Satellite 内から削除できます。

手順

1. Satellite Web UI で、**Infrastructure > Compute Resources**に移動します。
2. Microsoft Azure プロバイダーを選択します。
3. **Virtual Machines** タブで、**Actions** メニューから **Delete** をクリックします。これにより、Microsoft Azure コンピューティングリソースから仮想マシンが削除されますが、Satellite 内で関連付けられたホストは保持されます。孤立したホストを削除する場合は、**ホスト > すべてのホスト** に移動し、ホストを手動で削除します。

付録A プロビジョニングのサンプル用の初期化スクリプト

Managing Content の例に従っていない場合は、次の初期化スクリプトを使用して、例をプロビジョニングするための環境を作成できます。

手順

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

```
#!/bin/bash

MANIFEST=$1

# Import the content from Red Hat CDN
hammer organization create \
--name "ACME" \
--label "ACME" \
--description "Our example organization for managing content."

hammer subscription upload \
--file ~/$MANIFEST \
--organization "ACME"

hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (RPMs)" \
--releasever "7Server" \
--basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" \
--organization "ACME"

hammer repository-set enable \
--name "Red Hat Enterprise Linux 7 Server (Kickstart)" \
--releasever "7Server" \
--basearch "x86_64" \
--product "Red Hat Enterprise Linux Server" \
--organization "ACME"

hammer repository-set enable \
--name "Red Hat Satellite Client 6 (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 lifecycle
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
Client 6 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"

```

2. スクリプトに実行権限を設定します。

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

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

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

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

付録B FIPS 準拠ホストのプロビジョニング

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

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

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

詳細は、[ホストの管理](#) の [ホストグループの作成](#) を参照してください。

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

B.1. プロビジョニングパスワードハッシュアルゴリズムの変更

FIPS に準拠するホストをプロビジョニングするには、プロビジョニングで使用するパスワードのハッシュアルゴリズムを SHA256 に設定します。この設定は、FIPS 準拠としてデプロイするオペレーティングシステムごとに、適用する必要があります。

手順

1. オペレーティングシステム ID を特定します。

```
# hammer os list
```

2. 各オペレーティングシステムのパスワードハッシュ値を更新します。

```
# hammer os update \  
--password-hash SHA256 \  
--title "My_Operating_System"
```

値をコンマ区切りで指定することはできない点に注意してください。

B.2. FIPS 有効化パラメーターの設定

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

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

```
# hammer hostgroup set-parameter \  
--hostgroup "My_Host_Group" \  
--name fips_enabled \  
--value "true"
```

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

B.3. FIPS モードの有効化の確認

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

手順

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

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

値が **1** の場合は、FIPS モードが有効であることが分かります。

付録C RED HAT SATELLITE 向けクラウドイメージの構築

このセクションを使用して、Red Hat Satellite にイメージを構築して登録します。

事前に設定済みの Red Hat Enterprise Linux KVM ゲスト QCOW2 イメージを使用することができます。

- [最新の RHEL 9 KVM ゲストイメージ](#)
- [最新の RHEL 8 KVM ゲストイメージ](#)

これらのイメージには **cloud-init** が含まれます。適切に機能させるには、ec2 互換のメタデータサービスを使用して SSH キーをプロビジョニングする必要があります。



注記

KVM ゲストイメージでは、以下の点に注意してください。

- KVM ゲストイメージでは **root** アカウントが無効になっていますが、**cloud-user** という名前の特別なユーザーに **sudo** アクセスが許可されています。
- このイメージには **root** パスワードは設定されていません。**root** パスワードは、**/etc/shadow** で 2 番目のフィールドに **!!** と記載することによりロックされます。

カスタム Red Hat Enterprise Linux イメージを作成する場合は、[カスタマイズされた Red Hat Enterprise Linux 9 イメージの設定](#) または [カスタマイズされた Red Hat Enterprise Linux 8 イメージの設定](#) を参照してください。

C.1. RED HAT ENTERPRISE LINUX のカスタムイメージの作成

前提条件

- Linux ホストマシンを使用して、イメージを作成する。この例では、Red Hat Enterprise Linux 7 Workstation を使用します。
- ワークステーションで **virt-manager** を使用して、この手順を実行する。リモートサーバーでイメージを作成した場合、**virt-manager** を使用してワークステーションからサーバーに接続します。
- Red Hat Enterprise Linux 7 または 6 の ISO ファイル ([Red Hat Enterprise Linux 7.4 Binary DVD](#) または [Red Hat Enterprise Linux 6.9 Binary DVD](#) を参照)。

Red Hat Enterprise Linux Workstation のインストールに関する詳細は、[Red Hat Enterprise Linux 7 インストールガイド](#) を参照してください。

カスタムのイメージを作成する前に、以下のパッケージをインストールします。

- **libvirt**、**qemu-kvm** およびグラフィカルツールをインストールします。

```
# dnf install virt-manager virt-viewer libvirt qemu-kvm
```

- 以下のコマンドラインツールをインストールします。

```
# dnf install virt-install libguestfs-tools-c
```



注記

以下の手順では、**libvirt** 環境をホストするワークステーションで、**[root@host]#** プロンプトを伴うコマンドはすべて入力します。

C.2. 登録時にサポートされるクライアント

Satellite は、登録用に以下のオペレーティングシステムとアーキテクチャーをサポートします。

サポートされているホストオペレーティングシステム

ホストは、以下のオペレーティングシステムを使用できます。

- Red Hat Enterprise Linux 9、8、7
- [ELS アドオン](#) を使用する Red Hat Enterprise Linux 6

サポートされるホストアーキテクチャー

ホストは、次のアーキテクチャーを使用できます。

- i386
- x86_64
- s390x
- ppc_64

C.3. ホスト登録の設定

Satellite Server または Capsule Server に登録できるようにホストを設定します。

前提条件

- ホストはサポートされているオペレーティングシステムを使用している必要がある。詳細は、「[登録時にサポートされるクライアント](#)」を参照してください。

手順

- 時間同期ツールが有効になっていて、ホスト上で実行されていることを確認します。

- Red Hat Enterprise Linux 7 以降の場合:

```
# systemctl enable --now chronyd
```

- Red Hat Enterprise Linux 6 の場合:

```
# chkconfig --add ntpd
# chkconfig ntpd on
# service ntpd start
```

C.4. ホストの登録

登録テンプレートを使用してホストを登録し、登録プロセス中にさまざまなインテグレーション機能とホストツールを設定できます。

前提条件

- ユーザーアカウントに **create_hosts** 権限を持つロールが割り当てられている。
- 登録するホストで root 権限がある。
- Satellite Server、Capsule Server、およびすべてのホストを同じ NTP サーバーと同期し、時間同期ツールを有効にして実行しておく。
- ホストのアクティベーションキーがある。詳細は、[コンテンツの管理のアクティベーションキーの管理](#) を参照してください。
- オプション: ホストを Red Hat Insights に登録する場合は、**rhel-8-for-x86_64-baseos-rpms** リポジトリと **rhel-8-for-x86_64-appstream-rpms** リポジトリを同期し、アクティベーションキーでできるようにする必要があります。これは、**insights-client** パッケージをインストールするために必要です。
- Satellite Server の代わりに Capsule Server を使用する場合は、Capsule Server が適切に設定されていることを確認してください。詳細は、[Capsule Server のインストールのホストの登録とプロビジョニングのための Capsule の設定](#) を参照してください。
- Satellite Server または Capsule Server が HTTP プロキシの背後にある場合は、接続に HTTP プロキシを使用するようにホストでサブスクリプションマネージャーを設定します。詳細は、[Red Hat ナレッジベースの How to access Red Hat Subscription Manager \(RHSM\) through a firewall or proxy](#) を参照してください。

手順

1. Satellite Web UI で、**Hosts > Register Host** に移動します。
2. オプション: 別の **Organization** を選択します。
3. オプション: 別の **Location** を選択します。
4. オプション: **Host Group** リストから、ホストと関連付けるホストグループを選択します。 **Host group** からの値を継承するフィールド: **Operating system**、**Activation Keys**、**Lifecycle environment**。
5. オプション: **Operating system** リストから、登録するホストのオペレーティングシステムを選択します。
6. オプション: **Capsule** リストから、ホストの登録に使用する Capsule を選択します。



注記

ロードバランサーの背後にある Capsule は、Satellite Web UI でホストのコンテンツソースとして選択された Capsule よりも優先されます。

7. **Activation Keys** フィールドで、ホストに割り当てるアクティベーションキーを1つ以上入力します。
8. オプション: 最初の呼び出しを非セキュアにする場合は、**Insecure** オプションを選択します。この最初の呼び出し中に、ホストは Satellite から CA ファイルをダウンロードします。ホスト

は、この CA ファイルを使用して Satellite に接続し、今後のすべての呼び出しは安全になります。

Red Hat は、セキュアでない呼び出しを回避することを推奨します。

Satellite とホスト間のネットワークに存在する攻撃者が、初回の安全ではない呼び出しから CA ファイルをフェッチする場合、攻撃者は登録したホストと JSON Web Tokens (JWT) 間の API 呼び出しの内容にアクセスできます。そのため、登録中の SSH キーのデプロイを選択した場合、攻撃者は SSH キーを使用してホストにアクセスできます。

代わりに、ホストを登録する前に、各ホストに手動で CA ファイルをコピーしてインストールできます。

そのためには、**Administer > Settings > Authentication** に移動し、**SSL CA file** 設定を特定することで、Satellite が CA ファイルを保存する場所を検出します。

CA ファイルをホストの `/etc/pki/ca-trust/source/anchors/` ディレクトリーにコピーし、以下のコマンドを入力します。

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

次に、以下のようなセキュアな **curl** コマンドを使用してホストを登録します。

```
# curl -sS https://satellite.example.com/register ...
```

以下は、**--insecure** オプションを指定した **curl** コマンドの例です。

```
# curl -sS --insecure https://satellite.example.com/register ...
```

9. **Advanced** タブを選択します。
10. **Setup REX** リストから、Satellite SSH 鍵をホストにデプロイするかどうかを選択します。**Yes** に設定すると、パブリック SSH キーが登録済みホストにインストールされます。継承された値は **host_registration_remote_execution** パラメーターに基づいています。たとえば、ホストグループ、オペレーティングシステム、または組織から継承できます。上書きされると、選択した値がホストパラメーターレベルに保存されます。
11. オプション: **Setup Insights** リストから、**insights-client** をインストールして、ホストを Insights に登録するかどうかを選択します。Insights ツールは、Red Hat Enterprise Linux でのみ利用できます。他のオペレーティングシステムには影響ありません。

登録されたマシンで以下のリポジトリーを有効にする必要があります。

- Red Hat Enterprise Linux 6: **rhel-6-server-rpms**
- Red Hat Enterprise Linux 7: **rhel-7-server-rpms**
- Red Hat Enterprise Linux 8: **rhel-8-for-x86_64-appstream-rpms**
Insights-client パッケージは、Red Hat Enterprise Linux 8 が最小インストールオプションでデプロイされた環境を除き、デフォルトで Red Hat Enterprise Linux 8 にインストールされます。

12. オプション: **Install packages** フィールドに、登録時にホストにインストールするパッケージを(スペースで区切って)リストします。これは、**host_packages** パラメーターで設定できます。

13. オプション: 登録時にホスト上のすべてのパッケージを更新するには、**Update packages** オプションを選択します。これは、**host_update_packages** パラメーターで設定できます。
14. オプション: **Repository** フィールドに、登録を実行する前に追加するリポジトリを入力します。たとえば、登録の目的で **subscription-manager** パッケージを利用できるようにすると便利です。Red Hat ファミリーのディストリビューションの場合、リポジトリの URL を入力します (例: `:http://rpm.example.com/`)。
15. オプション: **Repository GPG key URL** フィールドで、公開鍵を指定して、GPG 署名付きパッケージの署名を確認します。GPG 公開鍵ヘッダーを持つ ASCII 形式で指定する必要があります。
16. オプション: **Token lifetime (hours)** フィールドで、Satellite が認証に使用する JSON Web トークン (JWT) の有効期間を変更します。このトークンの期間は、生成された **curl** コマンドが機能する期間を定義します。期間は、0 - 999 999 時間または無制限に設定できます。Satellite は、ホストの認証に **curl** コマンドを生成するユーザーのパーミッションを適用する点に注意してください。ユーザーが追加のパーミッションを失ったり取得したりすると、JWT のパーミッションも変わってきます。そのため、トークン期間中にユーザーのパーミッションを削除、ブロック、または変更しないでください。

JWT の範囲は登録エンドポイントのみに制限されているため、その他の場所では使用できません。
17. オプション: **リモート実行インターフェイス** フィールドに、ホストが SSH 接続に使用する必要があるネットワークインターフェイスの識別子を入力します。このフィールドが空の場合、Satellite はデフォルトのネットワークインターフェイスを使用します。
18. オプション: **REX pull mode** リストから、Satellite リモート実行プルクライアントをデプロイするかどうかを選択します。

Yes に設定すると、リモート実行のプルクライアントは登録されたホストにインストールされます。継承された値は **host_registration_remote_execution_pull** パラメーターに基づいています。たとえば、ホストグループ、オペレーティングシステム、または組織から継承できません。上書きされると、選択した値はホストパラメーターレベルに保存されます。

登録したホストは、Red Hat Satellite Client 6 リポジトリにアクセスする必要があります。

プルモードの詳細は、[ホストの管理](#) の [リモート実行用のトランスポートモード](#) を参照してください。
19. オプション: **Lifecycle environment** を選択します。
20. オプション: サブスクリプションマネージャーのエラーを無視する場合は、**Ignore errors** オプションを選択します。
21. オプション: 登録前に **katello-ca-consumer rpm** を削除し、**--force** 引数で **subscription-manager** を実行する場合は、**Force** オプションを選択します。
22. **Generate** をクリックします。
23. 生成された **curl** コマンドをコピーします。
24. 登録するホストで、**curl** コマンドを **root** として実行します。

C.5. KATELLO エージェントのインストール

Satellite クライアントをリモートで更新するには、Katello エージェントをインストールしてください。



注記

Katello エージェントは非推奨で、今後の Satellite のバージョンで削除されます。プロセスを移行し、リモート実行機能を使用してクライアントをリモートで更新してください。詳細は、[ホストの管理](#) の [Katello エージェントからリモート実行への移行](#) を参照してください。

katello-agent パッケージは、**goferd** service を提供する **gofer** パッケージによって異なります。

前提条件

- Satellite Server で、Satellite Client 6 リポジトリを有効にしておく。詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の [Satellite Client 6 リポジトリの有効化](#) を参照してください。
- Satellite Server で、Satellite Client 6 リポジトリを同期しておく。詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の [Satellite Client 6.12 リポジトリの同期](#) を参照してください。
- クライアントで Satellite Client 6 リポジトリを有効にしておく。

手順

1. **katello-agent** パッケージをインストールします。

```
# dnf install katello-agent
```

2. **goferd** サービスを開始します。

```
# systemctl start goferd
```

C.6. PUPPET AGENT の手動でのインストールと設定

Puppet エージェントを手動でホストにインストールし、設定できます。Satellite と Puppet を統合するには、設定済みの Puppet エージェントがホストに必要です。Puppet の詳細は、[Red Hat Satellite で Puppet インテグレーションを使用した設定の管理](#) を参照してください。

前提条件

- Satellite で Puppet が有効になっている。詳細は、[Red Hat Satellite で Puppet インテグレーションを使用して設定を管理する](#) の [Satellite で Puppet インテグレーションを有効にする](#) を参照してください。
- ホストに Puppet 環境が割り当てられている。
- **Satellite Client 6** リポジトリを有効化して Satellite Server に同期し、ホスト上で有効化している必要がある。詳細は、[コンテンツの管理](#) の [コンテンツのインポート](#) を参照してください。

手順

1. **root** ユーザーで、ホストにログインします。
2. Puppet エージェントパッケージをインストールします。

- Red Hat Enterprise Linux 8 以降を実行しているホストの場合:

```
# dnf install puppet-agent
```

- Red Hat Enterprise Linux 7 以前を実行しているホストの場合:

```
# yum install puppet-agent
```

3. 次のスクリプトを使用して、Puppet エージェントを現在のシェルの **PATH** に追加します。

```
./etc/profile.d/puppet-agent.sh
```

4. Puppet エージェントを設定します。ホストの所属先の Puppet 環境名に **environment** パラメーターを設定します。

```
# puppet config set server satellite.example.com --section agent  
# puppet config set environment My_Puppet_Environment --section agent
```

5. Puppet エージェントサービスを開始します。

```
# puppet resource service puppet ensure=running enable=true
```

6. ホストの証明書を作成します。

```
# puppet ssl bootstrap
```

7. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。

8. 必要な Capsule Server の **Actions** コラムの一覧から、**Certificates** を選択します。

9. 必要なホストの右にある **Sign** をクリックして、Puppet エージェントの SSL 証明書に署名します。

10. ホスト上で、Puppet エージェントを再度実行します。

```
# puppet ssl bootstrap
```

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

手順

1. システムを更新します。

```
# yum update
```

2. **cloud-init** パッケージをインストールします。

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

3. **/etc/cloud/cloud.cfg** 設定ファイルを開きます。

```
# vi /etc/cloud/cloud.cfg
```

4. **cloud_init_modules** のタイトルの下に、以下を追加します。

```
- resolv-conf
```

resolv-conf オプションは、インスタンスの初回起動時に **resolv.conf** を自動的に設定します。このファイルには、**nameservers**、**domain**、その他のオプションなどのインスタンスに関連した情報が記載されています。

5. **/etc/sysconfig/network** ファイルを開きます。

```
# vi /etc/sysconfig/network
```

6. 以下の行を追加し、EC2 メタデータサービスへのアクセスで問題が発生するのを回避します。

```
NOZEROCONF=yes
```

7. 仮想マシンの登録を解除して、作成されるイメージをベースにクローン作成されるインスタンスすべてに同じサブスクリプション情報が含まれないようにします。

```
# subscription-manager repos --disable=*  
# subscription-manager unregister
```

8. インスタンスの電源をオフにします。

```
# poweroff
```

9. Red Hat Enterprise Linux Workstation で、ルートユーザーとしてターミナルに接続し、**/var/lib/libvirt/images/** ディレクトリーに移動します。

```
# cd /var/lib/libvirt/images/
```

10. **virt-sysprep** コマンドでイメージのリセットおよびクリーニングをして、問題なくインスタンスの作成に使用できるようにします。

```
# virt-sysprep -d rhel7
```

11. **virt-sparsify** コマンドを使用してイメージのサイズを縮小します。このコマンドにより、ディスクイメージ内の空き容量は、ホスト内の空き容量に戻ります。

```
# virt-sparsify --compress rhel7.qcow2 rhel7-cloud.qcow2
```

これにより、コマンドを実行する場所に新しい **rhel7-cloud.qcow2** ファイルが作成されます。

C.8. RED HAT ENTERPRISE LINUX 6 イメージの完了

手順

1. システムを更新します。

```
# yum update
```

2. **cloud-init** パッケージをインストールします。

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

3. **/etc/cloud/cloud.cfg** 設定ファイルを編集して、**cloud_init_modules** の下に以下を追加します。

```
- resolv-conf
```

resolv-conf オプションは、インスタンスの初回起動時に **resolv.conf** 設定ファイルを自動的に設定します。このファイルには、**nameservers**、**domain**、その他のオプションなどのインスタンスに関連した情報が記載されています。

4. ネットワークの問題が発生するのを防ぐために、以下のように **/etc/udev/rules.d/75-persistent-net-generator.rules** ファイルを作成します。

```
# echo "#" > /etc/udev/rules.d/75-persistent-net-generator.rules
```

これにより、**/etc/udev/rules.d/70-persistent-net.rules** ファイルが作成されるのを防ぎます。**/etc/udev/rules.d/70-persistent-net.rules** が作成されてしまうと、スナップショットからのブート時にネットワークが適切に機能しなくなる可能性があります(ネットワークインターフェイスが eth0 ではなく eth1 として作成され、IP アドレスが割り当てられません)。

5. **/etc/sysconfig/network** に以下の行を追加し、EC2 メタデータサービスへのアクセスで問題が発生するのを回避します。

```
NOZEROCONF=yes
```

6. 仮想マシンの登録を解除して、作成されるイメージをベースにクローン作成されるインスタンスすべてに同じサブスクリプション情報が含まれないようにします。

```
# subscription-manager repos --disable=*
# subscription-manager unregister
# yum clean all
```

7. インスタンスの電源をオフにします。

```
# poweroff
```

8. Red Hat Enterprise Linux Workstation でルートとしてログインし、**virt-sysprep** コマンドを使用してイメージのリセットとクリーニングをし、問題なくインスタンスの作成に使用できるようにします。

```
# virt-sysprep -d rhel6
```

9. **virt-sparsify** コマンドを使用してイメージのサイズを縮小します。このコマンドにより、ディスクイメージ内の空き容量は、ホスト内の空き容量に戻ります。

```
# virt-sparsify --compress rhel6.qcow2 rhel6-cloud.qcow2
```

これにより、コマンドを実行する場所に新しい **rhel6-cloud.qcow2** ファイルが作成されます。



注記

インスタンスに適用されているフレーバーのディスクスペースに応じて、イメージをベースとするインスタンスのパーティションを手動でリサイズする必要があります。

C.8.1. 次のステップ

- Satellite とプロビジョニングするすべてのイメージで、この手順を繰り返します。
- 後で使用できるように、保存場所にイメージを移動します。

C.9. 次のステップ

- Satellite とプロビジョニングするすべてのイメージで、この手順を繰り返します。
- 後で使用できるように、保存場所にイメージを移動します。

付録D ホストパラメーター階層

ホストのプロビジョニング時に、ホストパラメーターにアクセスできます。ホストは、優先度の高い順に、次の場所からパラメーターを継承します。

パラメーターレベル	Satellite Web UI で設定
グローバルに定義されたパラメーター	Configure > Global parameters
組織レベルのパラメーター	Administer > Organizations
ロケーションレベルのパラメーター	Administer > Locations
ドメインレベルのパラメーター	Infrastructure > Domains
サブネットレベルのパラメーター	Infrastructure > Subnets
オペレーティングシステムレベルのパラメーター	Hosts > Operating Systems
ホストグループレベルのパラメーター	Configure > Host Groups
ホストパラメーター	Hosts > All Hosts

付録E ホストをプロビジョニングするために必要なパーミッション

次のリストは、管理者以外のユーザーがホストをプロビジョニングするために必要なパーミッションの概要を示しています。

リソース名	パーミッション	詳細
アクティベーションキー	view_activation_keys	
Ansible ロール	view_ansible_roles	Ansible を使用する場合は必須。
アーキテクチャー	view_architectures	
コンピュートプロファイル	view_compute_profiles	
コンピュートリソース	view_compute_resources、 create_compute_resources、 destroy_compute_resources、 power_compute_resources	ベアメタルホストをプロビジョニングするために必須。
	view_compute_resources_vms、 create_compute_resources_vms 、 destroy_compute_resources_vms 、 power_compute_resources_vms	仮想マシンをプロビジョニングするために必須。
Content Views	view_content_views	
ドメイン	view_domains	
Environment	view_environments	
ホスト	view_hosts、 create_hosts、 edit_hosts、 destroy_hosts、 build_hosts、 power_hosts、 play_roles_on_host	
	view_discovered_hosts、 submit_discovered_hosts、 auto_provision_discovered_hosts 、 provision_discovered_hosts、 edit_discovered_hosts、 destroy_discovered_hosts	Discovery サービスが有効になっている場合は必須。
Hostgroup	view_hostgroups、 create_hostgroups、 edit_hostgroups、 play_roles_on_hostgroup	

リソース名	パーミッション	詳細
イメージ	view_images	
ライフサイクル環境	view_lifecycle_environments	
ロケーション	view_locations	
Medium	view_media	
オペレーティング・システム	view_operatingsystems	
Organization	view_organizations	
パラメーター	view_params、 create_params、 edit_params、 destroy_params	
製品とリポジトリ	view_products	
Provisioning Template (プロビジョニングテンプレート)	view_provisioning_templates	
Ptable	view_ptables	
Capsule	view_smart_proxies、 view_smart_proxies_puppetca	
	view_openscap_proxies	OpenSCAP プラグインが有効な場合に必須。
サブネット	view_subnets	

関連情報

- Red Hat Satellite の管理の [ロールの作成](#)
- Red Hat Satellite の管理の [ロールへのパーミッションの追加](#)
- Red Hat Satellite の管理の [ユーザーへのロールの割り当て](#)