



Red Hat Satellite 6.2

ホスト設定ガイド

Red Hat Satellite 6 環境におけるホストの管理ガイド
エディション 1.0

Red Hat Satellite Documentation
Team

Red Hat Satellite 6.2 ホスト設定ガイド

Red Hat Satellite 6 環境におけるホストの管理ガイド エディション 1.0

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

法律上の通知

Copyright © 2016 Red Hat.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

『Red Hat Satellite 6 ホスト設定ガイド』では、Red Hat Satellite 環境でホストを設定し、これを使用する方法について説明します。この作業を続行する前に、Red Hat Satellite 6 Server と必要なすべての Capsule Server が正常にインストールされている必要があります。

目次

第1章 Red Hat Satellite コンテンツダッシュボードの使用	3
1.1. タスクの管理	4
第2章 検索およびブックマーク機能	6
2.1. 検索クエリーの構築	6
2.2. フリーテキスト検索の使用	8
2.3. ブックマーク機能	8
第3章 コンテンツビューの使用	10
3.1. コンテンツビューの作成	10
3.2. リポジトリのコンテンツビューへの追加	11
3.3. コンテンツのフィルタリング	11
3.4. コンテンツビューの公開	13
第4章 エラータの表示および適用	14
4.1. 利用可能なエラータの検査	14
4.2. エラータのコンテンツホストへの適用	16
4.3. エラータ通知のサブスクライブ	17
第5章 セキュリティーコンプライアンス管理	18
5.1. SCAP について	18
5.2. インストール	19
5.3. コンプライアンスポリシーの管理	21
5.4. コンプライアンスのモニター	23
第6章 コンテナの使用	31
6.1. コンテナの管理	31
6.2. リポジトリの管理	36
6.3. 外部レジストリーのインポート	37
6.4. イメージのコンピュートリソースへのインポート	37
6.5. コンテナタグの使用	38
第7章 アクティベーションキーの設定	39
7.1. アクティベーションキーの作成	39
7.2. アクティベーションキーのサブスクリプションプロパティの定義	39
7.3. アクティベーションキーの編集	41
7.4. アクティベーションキーの削除	43
7.5. アクティベーションキーを使用したホストの自動登録	43
第8章 GPG キーの設定	45
8.1. GPG キーの作成	45
8.2. GPG キーの削除	45
第9章 プロビジョニング環境の設定	46
9.1. ホストグループの作成	46
9.2. パラメーター	47
9.3. プロビジョニング設定の構成	52
9.4. ホスト情報の保存および維持	62
第10章 ホストの設定	67
10.1. ホストの参照	67
10.2. ホストの状態タイプ	67
10.3. ホストの概要	67
10.4. ホストの作成	69
10.5. ホストの削除	71

10.5. ホスト登録の設定	69
10.6. 登録	71
10.7. ホストのグループの変更	76
10.8. ホストの環境の変更	76
10.9. ホストの管理	76
10.10. ホストの特定組織への割り当て	77
10.11. ホストの特定ロケーションへの割り当て	77
10.12. ブートストラップスクリプトによる新規ホストの Satellite 6 への追加	78
10.13. 追加のネットワークインターフェースの設定	80
10.14. ホストの削除	84
第11章 Satellite でのベアメタルホストの検出	86
11.1. Satellite Discovery プラグインの設定	86
11.2. Satellite Capsule Server Discovery プラグインの設定	89
11.3. 検出されたホストのプロビジョニング	90
11.4. Discovery イメージの拡張	93
11.5. Satellite 検出のトラブルシューティング	94
第12章 Satellite ホストでのジョブの実行	96
12.1. リモートコマンドのセキュアな接続の確立	96
12.2. リモートコマンドの設定および実行	97
12.3. グローバル設定	102
12.4. リモート実行用のパーミッションの委任	103
第13章 ホストコレクションの設定	105
13.1. ホストコレクションの作成	105
13.2. ホストのホストコレクションへの追加	105
13.3. コンテンツのホストコレクションへの追加	105
13.4. ホストコレクションからのコンテンツの削除	107
13.5. ホストコレクションのライフサイクル環境またはコンテンツビューの変更	107
13.6. ホストコレクションからのホストの削除	107
13.7. ホストコレクションの削除	108
13.8. ホストコレクションのクローン作成	108
13.9. ホストコレクションの詳細の確認	108
付録A テンプレート作成の参照	109
A.1. ERB テンプレートの作成	109
A.2. ERB テンプレートのトラブルシューティング	110
A.3. Satellite 固有の関数および変数	110

第1章 Red Hat Satellite コンテンツダッシュボードの使用




Red Hat Satellite コンテンツダッシュボードはサブスクリプションと現在登録されているホストの状態についての概要や、プロモーションおよび同期の概要、さらに最新の通知一覧を表示します。

モニター → ダッシュボード に移動してコンテンツダッシュボードにアクセスします。ダッシュボードの配置は、セクションのタイトルをクリックし、セクションを別の位置にドラッグすることで変更することができます。以下のセクションを利用できます。

コンテンツホストのサブスクリプションの状態

Satellite に登録されているホストによって現在使用されているサブスクリプションの概要です。サブスクリプションとはご購入いただいた証明書を指します。このサブスクリプションでホストのソフトウェア、アップグレード、およびセキュリティー修正などが利用できるようになります。以下の表はサブスクリプションの状態の種類を示しています。

表1.1 ホストのサブスクリプションの状態

状態	説明	アイコン
無効	製品がインストールされていて、サブスクリプションが適切に使用されていないホストです。これらのホストには早急な対応が必要です。	
部分的	サブスクリプションが使用されていて、有効なエンタイトルメントを持つホストですが、それらのエンタイトルメントは完全には使用されていません。これらのホストが予定通りに設定されていることを確認するために、これらのホストをモニターする必要があります。	
有効性 (Valid)	有効なエンタイトルメントを有し、それらのエンタイトルメントを完全に使用しているホストです。	

サブスクリプションタイプを選択し、選択したタイプのサブスクリプションに関連付けられたコンテンツホストを表示します。

最新イベント

管理情報、製品、サブスクリプションの変更およびエラーに関するホストが生成するメッセージの一覧です。

すべてのユーザーに送信されるグローバル通知や、異常なアクティビティまたはエラーを検出するためにこのセクションをモニターします。

同期の概要

Satellite およびそれらの同期の状態で有効にされているすべての製品またはリポジトリの概要です。同期待ちになっている製品、同期されていない製品、同期が行われた製品はすべてこのセクションに一覧表示されます。製品名をクリックすると同期の状態が表示されます。

ホストコレクション

Satellite 内のすべてのホストコレクションとそれらの状態の一覧で、各ホストコレクション内のコンテンツホストの数なども含まれます。ホストコレクションを表示するには、そのホストコレクションの名前をクリックします。

現在のサブスクリプション合計

アクティブなサブスクリプションの数、次の 120 日で期限の切れるサブスクリプションの数、および最近期限切れになったサブスクリプションの数を表示する現在のサブスクリプション合計の概要です。数をクリックして選択したサブスクリプションを一覧表示します。

コンテンツビューの概要

Satellite におけるすべてのコンテンツビューおよびそれらの公開状態の一覧です。

エラータの概要

Satellite に登録されているホストで利用できるすべてのエラータの一覧です。

タスクのステータス

ステータスと結果別に分類される現在のすべてのタスクの概要です。タスク数をクリックして対応するタスクの一覧に移動します。

最新の警告/エラータスク

警告またはエラーにより停止している最新タスクの一覧です。タスクをクリックして詳細を確認してください。



注記

Satellite Web UI で表示される日付の形式を変更することはできません。

1.1. タスクの管理

Red Hat Satellite は、同期されたりリポジトリ、適用されたエラータ、公開されたコンテンツビューなどの計画されたタスクまたは実行されたタスクのすべての詳細なログを保持します。ログを確認するには、**モニター** → **タスク** に移動します。このページから、特定のタスクを検索し、(該当する場合は) エラーを出したタスクを再開できます。

タスクは Dynflow エンジンを使用して管理されます。リモートタスクには、必要に応じて調整できるタイムアウトが設定されます。

手順1.1 タイムアウト設定を調整するには、以下を実行します。

1. **管理** → **設定** に移動します。
2. 検索ボックスに **%_timeout** を入力し、**検索** をクリックします。検索では、説明を含む 4 つの設定が返されるはずです。
3. **値** 列で、数字の横にあるアイコンをクリックして編集します。
4. 秒単位の必要な値を入力して **保存** をクリックします。



注記

低帯域幅の場合は **%_finish_timeout** 値の編集が役に立つ場合があります。待ち時間が長い場合は **%_accept_timeout** 値が役立つことがあります。

タスクが初期化されると、Candlepin または Pulp などのタスクで使用されるすべてのバックエンドサービスについて正常に機能するかどうかチェックされます。チェックにパスしない場合は、次のようなエラーを受信します。


```
There was an issue with the backend service candlepin: Connection refused – connect(2).
```

バックエンドサービスのチェック機能に問題が生じている場合は、以下のように無効にすることができます。

手順**1.2** サービスのチェックを無効にするには、以下を実行します。

1. **管理** → **設定** に移動します。
2. 検索ボックスに **check_services_before_actions** を入力し、**検索** をクリックします。
3. **値** 列のアイコンをクリックし、値を編集します。
4. ドロップダウンメニューから、**false** を選択します。
5. **保存** をクリックします。

第2章 検索およびブックマーク機能

Satellite Web UI は Web UI の大半のページで利用できる強力な検索機能を特長としています。この機能によって Satellite Server が管理するあらゆる種類のリソースを検索できます。検索では、フリーテキストと構文ベースのクエリーの両方を使用でき、クエリーは詳細な予測入力を使用して実行されます。検索クエリーは今後の再利用に備えてブックマークとして保存することができます。

2.1. 検索クエリーの構築

検索クエリーの入力を開始すると、現在のクエリーを補完する有効なオプションの一覧が表示されます。一覧からオプションを選択するか、予測機能を使用してクエリーを構築するか、または入力を継続するかのいずれかのオプションを選択できます。検索エンジンがフリーテキストを解釈する方法については、[「フリーテキスト検索の使用」](#) を参照してください。

2.1.1. クエリーの構文

parameter operator value

検索に利用できるフィールド、リソースおよびクエリーが解釈される方法はコンテキスト、つまり検索を実行するページによって異なります。たとえば、「ホスト」ページの「ホストグループ」フィールドは「ホストグループ」ページの「名前」フィールドに相当します。またフィールドのタイプにより、利用可能な演算子および許可される値が決まります。すべての演算子の一覧については、[「演算子」](#) を参照してください。値の形式についての説明は、[「値」](#) を参照してください。

2.1.2. 演算子

パラメーターと値間で使用できるすべての演算子は以下の表に一覧表示されています。予測に基づいて構築されるクエリーで表示される可能性のある他の記号および特殊文字 (コロンなど) には特別な意味がなく、フリーテキストとして処理されます。

表2.1 検索で利用できる比較演算子

演算子	ショートネーム	説明	例
=	EQUALS	数値、時間的な値 (temporal value)	hostgroup = RHEL7
!=	NOT EQUALS	またはテキストの値を受け入れます。テキストの場合、大文字と小文字が区別された完全一致が返されます。	
~	LIKE	テキストまたは時間的な値 (temporal value) を受け入れます。大文字と小文字を区別しない一致を返します。1 文字の場合の _、ゼロを含む任意の数の文字の場合の % または たは * などのワイルドカードを受け入れます。ワイルドカードが指定されない場合、文字列はワイルドカードで囲まれている場合の様に処理されます (例: %rhel7%)。	hostgroup ~ rhel%
!~	NOT LIKE		

演算子	ショートネーム	説明	例
>	GREATER THAN	数値、または時間的な値 (temporal value) を受け入れます。時間的な値の場合、演算子 > is は「later than (次の日付より後)」として、< は「earlier than (次の日付より前)」として解釈されます。どちらの演算子も EQUALS: >= <= と組み合わせることができます。	registered_at > 10-January-2017
<	LESS THAN		検索は、指定された日付の後、つまり 2017 年 1 月 10 日から現在までの間に登録されたホストを返します。 registered_at <= Yesterday 検索は、昨日または昨日よりも前に登録されたホストを返します。
^	IN	SQL の場合と同様に、値の一覧に対して式を比較します。値が含まれる一致または値の含まれない一致をそれぞれ返します。	release_version !^ 7
!^	NOT IN		
HAS		存在する値、または存在しない値をそれぞれ返します。	has hostgroup
NOT HAS			「Puppet クラス」ページでは、検索は 1 つ以上のホストグループに割り当てられるクラスを返します。 not has hostgroup ホストの概要が示される「ダッシュボード」では、検索はホストグループが割り当てられていないすべてのホストを返します。

記述された構文に従う単純なクエリーを組み合わせ、論理演算子の AND、OR および NOT を使用してより複雑なクエリーにすることができます。演算子の代替表記も使用できます。

表2.2 検索で使える論理演算子

演算子	代替表記			例
and	&	&&	<空白文字>	class = motd AND environment ~ production
or				errata_status = errata_needed errata_status = security_needed
not	-	!		hostgroup ~ rhel7 not status.failed

2.1.3. 値

テキストの値

空白文字を含むテキストは引用符で囲む必要があります。そうでないと、空白は AND 演算子として解釈されます。

例:

hostgroup = "Web servers"

検索は、「Web Servers」という名前の割り当て済みのホストグループと共にホストを返します。

hostgroup = Web servers

検索は、%servers% に一致するフィールドを持つホストグループ Web のホストを返します。

時間的な値 (temporal value)

以下を含め、数多くの日付/時刻形式を使用できます。

- ✧ "10 January 2017"
- ✧ "10 Jan 2017"
- ✧ 10-January-2017
- ✧ 10/January/2017
- ✧ "January 10, 2017"
- ✧ 「Today」、「Yesterday」など。



警告

02/10/2017 または 10-02-2017 などのあいまいな日付形式を使用しないようにしてください。

2.2. フリーテキスト検索の使用

フリーテキストを入力する際、複数のフィールドにまたがって検索が実行されます。たとえば、「64」と入力する場合、検索は名前、IP アドレス、MAC アドレスおよびアーキテクチャーにこの数字が含まれるすべてのホストを返します。



注記

複数の語句からなるクエリは引用符で囲む必要があります。そうしない場合は、空白文字は AND 演算子として解釈されます。

検索はすべてのフィールドで実行されるため、フリーテキストの検索結果は非常に正確になりますが、多数のホストで実行する場合などには検索スピードが遅くなる可能性があります。このため、可能な限りフリーテキストを使用せず、より具体的で、構文ベースのクエリを使用することが推奨されます。

2.3. ブックマーク機能

検索クエリを保存するには、**検索** ボタンの横にあるメニューの **この検索をブックマーク** 項目を使用します。保存されたブックマークは同じメニューに各自が選択した名前で表示されます。作成時に、ブックマークは「公開」マークを付けることができます。これによりブックマークはすべてのユーザーに対して表示されますが、「非公開 (private)」にすると作成したユーザーにしか表示されなくなります。

各ページにはそれぞれのブックマークがあります。一部のページには、すべてのアクティブなホスト、無効にされたホスト、または最近エラーが報告されたホストなどの最も一般的な検索で利用できるデフォルトのブックマークがあります。すべての既存のブックマークを表示し、これらを管理するには、**管理 → ブックマーク** に移動します。

第3章 コンテンツビューの使用

コンテンツビューは、オプションのフィルター機能と共に 1 つ以上のリポジトリ (yum、puppet、またはコンテナ) を含む管理対象のコンテンツセレクションです。フィルターは含めることも除外することもでき、これらのフィルターによってライフサイクルの管理用にコンテンツのシステムビューを調整することができます。また、フィルターを使用してクライアントシステムで利用できるようにコンテンツをカスタマイズすることができます。

yum、Puppet、およびコンテナのリポジトリを別々のコンテンツビューに保持すると、1 つのリポジトリに対する更新には関連するコンテンツビューの再公開のみが必要になるという利点があります。複合コンテンツビューを使用して公開されたコンテンツビューを組み合わせることで管理を容易にすることができます。

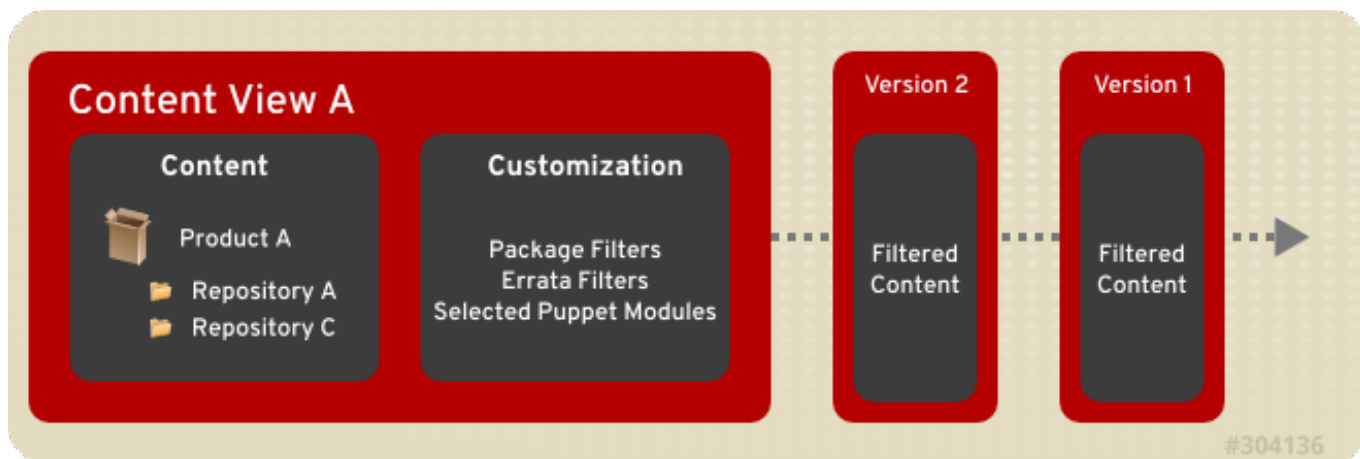


図3.1 上記の図は、コンテンツビューの新規バージョンの作成について詳細に説明しています。コンテンツビューのこれらのバージョンは、アプリケーションライフサイクル期間に環境パス上でプロモートされます。

公開されたコンテンツビューがライフサイクル環境と共に使用されます。

3.1. コンテンツビューの作成

管理者権限を持つユーザーはライフサイクル環境で利用できるコンテンツビューを作成することができます。

手順3.1 コンテンツビューを作成するには、以下を実行します。

1. Satellite 管理者としてログインします。
2. コンテンツ → コンテンツビュー をクリックします。
3. **新規ビューの作成** をクリックします。
4. コンテンツビューの **名前** を指定します。**ラベル** フィールドには、**名前** フィールドに記入すると同時に情報が読み込まれます。オプションとして、コンテンツビューの説明を記載します。
5. **複合ビュー?** チェックボックスを選択すると、一連の公開されたコンテンツビューを 1 つに組み込み、コンテンツビューを選択できます。

**注記**

複合ビュー? を選択すると、フィルターが上書きされます。さらに、公開されたコンテンツビューのグループを選択したり、これらのビューを 1 つの複合ビューにバンドルしたりできるようになります。

6. **保存** をクリックします。

3.2. リポジトリのコンテンツビューへの追加

リポジトリは、コンテンツ用のストレージを提供します。たとえば、YUM リポジトリ、Puppet リポジトリ、または Docker リポジトリなどがあります。

コンテンツビュー ページには、検索可能なコンテンツビューの一覧が含まれます。**検索** フィールドを使用してコンテンツビューを検索する際、部分的な文字列検索を実行するにはアスタリスク (*) を使用します。

たとえば、**RHEL7_Base** という名前のコンテンツビューを検索する場合、**RHEL7** と入力しても結果が返されません。その代わりに **RHEL7*** と入力します。または、***Base*** と入力してもコンテンツビュー **RHEL7_Base** を検索できます。

手順3.2 リポジトリをコンテンツビューに関連付けるには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** をクリックして、リポジトリを追加するコンテンツビューを選択します。
2. 保存する必要のあるコンテンツのタイプに応じて以下を実行します。
 - ※ Yum リポジトリを追加するには、**Yum コンテンツ** をクリックし、ドロップダウンメニューから **リポジトリ** を選択します。サブメニューの **追加** をクリックします。
 - ※ Puppet リポジトリを追加するには、**Puppet モジュール** をクリックしてから **新規モジュールの追加** をクリックします。
 - ※ Docker リポジトリを追加するには、**Docker コンテンツ** をクリックしてからサブメニューの **追加** をクリックします。
3. 追加するリポジトリを選択し、**リポジトリの追加** をクリックします。

3.3. コンテンツのフィルタリング

フィルターは、パッケージが後続の環境にプロモートされるのを防ぐメカニズムを提供します。フィルターでパッケージ名または正規表現がフィルターを使用すると、パッケージをブラックリストに記載するためのルールが作成されます。次に、そのフィルターが製品全体か、または製品内の個別のリポジトリに関連付けられます。

3.3.1. フィルターの作成

以下の手順では、パッケージのフィルターを作成する方法を説明します。

手順3.3 フィルターを作成するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** に移動して、フィルターするコンテンツビューを選択します。
2. **Yum コンテンツ** → **フィルター** をクリックしてから **新規フィルター** をクリックします。

3. **名前** フィールドに新規フィルターの名前を指定し、**コンテンツタイプ** ドロップダウンメニューからコンテンツタイプを選択します。**包含タイプ** ドロップダウンメニューを選択し、選択したコンテンツをフィルターに組み込むか、または除外するかを選択します。オプションとして **説明** フィールドに説明を入力します。
4. **保存** をクリックして新規フィルターを保存します。

3.3.2. コンテンツのフィルターへの追加

以下の手順では、コンテンツをパッケージフィルターに追加する方法を示します。

手順3.4 コンテンツをフィルターに追加するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** に移動して、フィルターするコンテンツビューを選択します。
2. **Yum コンテンツ** → **フィルター** をクリックし、編集するフィルターの名前をクリックします。選択したフィルターのタイプに応じて以下のアクションを実行します。
 - a. フィルターが複数のパッケージ用に作成されている場合、**パッケージ** サブタブでパッケージ名を指定し、ドロップダウンメニューから **詳細** の値を選択します。**追加** をクリックしてパッケージをフィルターに追加します。
 - b. フィルターが複数のパッケージグループ用に作成されている場合、**追加** サブタブをクリックし、必要なパッケージグループを選択します。**パッケージグループの追加** をクリックします。
 - c. フィルターがエラータ用に作成されている場合、**追加** サブタブをクリックします。エラータイプ (**セキュリティー**、**機能強化**、または **バグ修正**) を選択し、開始日と終了日を指定します。**エラータの追加** をクリックします。
 - d. フィルターが「エラータ - 日付およびタイプ」用に **エラータの日付範囲** サブタブで作成されている場合、エラータイプ (**セキュリティー**、**機能強化**、または **バグ修正**) を選択し、開始日および終了日を指定します。**保存** をクリックします。
3. **影響するリポジトリ** サブタブで、フィルターがリポジトリのすべてまたはサブセットに影響を与えるようにするかどうかを選択します。リポジトリのサブセットを選択する場合、必要なリポジトリを選択してから **リポジトリの更新** をクリックします。
4. **新規バージョンの公開** をクリックします。必要な場合はコマンドを入力し、**保存** をクリックします。

3.3.3. フィルターからのコンテンツの削除

以下の手順では、パッケージフィルターからコンテンツを削除する方法について示します。

手順3.5 コンテンツをフィルターから削除するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** に移動して、フィルターするコンテンツビューを選択します。
2. **Yum コンテンツ** → **フィルター** をクリックし、編集するフィルターの名前をクリックします。選択したフィルターのタイプに応じて以下のアクションを実行します。
 - a. フィルターがパッケージ用に作成されている場合、**パッケージ** サブタブをクリックし、削除するパッケージの横にある **パッケージ名 (Package Name)** チェックボックスを選択します。**パッケージの削除** をクリックして、パッケージをフィルターから削除します。

- b. フィルターが複数のパッケージグループ用に作成されている場合、**一覧/削除** サブタブをクリックしてから削除するパッケージグループの横にある **名前** チェックボックスを選択します。**パッケージグループの削除** をクリックして、パッケージグループをフィルターから削除します。
 - c. フィルターがエラータ用に作成されている場合、**一覧/削除** サブタブをクリックし、削除するエラータの横にある **エラータ ID** チェックボックスを選択します。**エラータの削除** をクリックし、エラータをフィルターから削除します。
 - d. フィルターが「エラータ - 日付およびタイプ」用に **エラータの日付範囲** サブタブで作成されている場合、エラータタイプ (**セキュリティ**、**機能強化**、または **バグ修正**) にチェックを付け、開始日および終了日を指定します。**保存** をクリックします。
3. **影響するリポジトリ** サブタブで、フィルターがリポジトリのすべてまたはサブセットに影響を与えるようにするかどうかを選択します。リポジトリのサブセットを選択する場合、必要なリポジトリを選択してから **リポジトリの更新** をクリックします。
4. **新規バージョンの公開** をクリックします。必要な場合はコマンドを入力し、**保存** をクリックします。

3.3.4. フィルターの削除

以下の手順では、フィルターを削除する方法について示します。

手順3.6 フィルターを削除するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** に移動して、フィルターするコンテンツビューを選択します。
2. **Yum** **コンテンツ** → **フィルター** をクリックしてから、削除するパッケージフィルターの名前の横にあるチェックボックスを選択します。
3. **フィルターの削除** をクリックします。

3.4. コンテンツビューの公開

コンテンツビューが作成されたら、コンテンツビューを表示し、ホストで 사용할 ことができるようにするために公開する必要があります。コンテンツビューの定義を公開する前に、コンテンツビューの定義に必要な製品、リポジトリおよびフィルターが含まれていることを確認します。

手順3.7 コンテンツビュー定義を公開するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** をクリックします。
2. 公開するコンテンツビューをクリックします。
3. **新規バージョンの公開** をクリックします。
4. コメントを入力します。
5. **保存** をクリックします。

第4章 エラータの表示および適用

Red Hat 製品のソフトウェアパッケージは、定期的また随時にリリースされるエラータと呼ばれる更新に基づいて提供されます。Red Hat Satellite は、エラータを検査し、フィルターするツールを提供し、詳細な更新管理を可能にします。これにより、関連する更新を選択し、コンテンツビューからそれらを選択したコンテンツホストに伝播できます。コンテンツビューについての詳細は、[3章 コンテンツビューの使用](#)を参照してください。



重要

「[Katello エージェントのインストール](#)」に説明されているように、*katello-agent* パッケージを Satellite Server にインストールします。このパッケージはエラータ管理に必要なサービスを提供します。

最新の更新を適用するために、Satellite コンテンツを正常に同期していることを確認します。モニター → ダッシュボード に移動して、エラータの同期の概要を確認します。

エラータには、更新で導入される変更について説明したアドバイザリーが含まれます。アドバイザリーには 3 つの種類があります (重要度の順に表示)。

- ✦ **セキュリティアドバイザリー:** パッケージに見つかったセキュリティ上の問題について説明します。問題のセキュリティ上の影響度については、**低度の影響**、**中程度の影響**、**重要な影響**、または **重大な影響** に分けられます。
- ✦ **バグ修正アドバイザリー:** パッケージのバグ修正について説明します。
- ✦ **製品の機能強化アドバイザリー:** パッケージに追加された機能強化および新機能について説明します。



注記

エラータには、それらに含まれる最も重要なアドバイザリータイプに応じてラベルが付けられます。そのため、**製品の機能強化アドバイザリー** というラベルが付けられたエラータには機能強化の更新のみが含まれ、**バグ修正アドバイザリー** エラータにはバグ修正と機能強化の両方が含まれ、**セキュリティアドバイザリー** にはこれら 3 つのタイプが含まれる場合があります。

Red Hat Satellite では、エラータと利用可能なコンテンツホストとの関係を説明する 2 つのキーワードが使われています。

- ✦ **適用可能:** エラータは 1 つ以上のコンテンツホストに適用されます。つまり、コンテンツホストにあるパッケージが更新されることを意味します。「適用可能」エラータはコンテンツホストがアクセスできる段階にはありません。
- ✦ **インストール可能:** エラータは 1 つ以上のコンテンツホストに適用され、コンテンツホストで利用できる状態です。インストール可能なエラータはコンテンツホストのライフサイクル管理とコンテンツビューにあります。インストールされていません。エラータは、コンテンツホストを管理するためのパーミッションがあるものの、より高いレベルでエラータ管理を行う権利を与えられていないユーザーがインストールできます。

4.1. 利用可能なエラータの検査

以下の手順では、利用可能なエラータを表示し、フィルターする方法や、選択したアドバイザリーのメタデータを表示する方法について説明します。

手順4.1 利用可能なエラータを検査するには、以下を実行します。

1. コンテンツ → エラータ に移動して、利用可能なエラータの一覧を表示します。
2. ページ上部のフィルターツールを使用して、表示されるエラータの数を制限します。
 - ※ ドロップダウンリストから検査するリポジトリを選択します。デフォルトでは **すべてのリポジトリ** が選択されます。
 - ※ **適用可能** チェックボックスがデフォルトで選択され、選択したリポジトリに適用されるエラータのみが表示されます。**インストール可能** チェックボックスを選択すると、インストール可能なマークが付けられたエラータのみが表示されます。
 - ※ エラータの表を検索するには、以下の形式で **検索** フィールドにクエリーを入力します。

```
parameter operator value
```

検索に使用できるパラメーターの一覧については、[表4.1「エラータ検索で利用できるパラメーター」](#)を参照してください。適用できる演算子の一覧については、『Server Administration Guide』の[Supported Operators for Granular Search](#)を参照してください。また、**and** および **or** 演算子を使用してクエリーを組み合わせることもできます。たとえば、*kernel* パッケージに関連するセキュリティーアドバイザリーのみを表示するには、以下を入力します。

```
type = security and package_name = kernel
```

Enter を押して検索を開始します。

3. 検査する必要があるエラータの **エラータ ID** をクリックします。
 - ※ **詳細** タブには、更新されたパッケージの説明や、更新によって提供される重要な修正および機能強化についての記載が含まれています。
 - ※ **コンテンツホスト** タブでは、[「エラータのコンテンツホストへの適用」](#)に説明されているように、エラータを選択したコンテンツホストに適用できます。
 - ※ **リポジトリ** タブには、エラータがすでに含まれているリポジトリが一覧表示されます。リポジトリは環境およびコンテンツビューでフィルターでき、リポジトリ名でそれらをフィルターできます。

表4.1 エラータ検索で利用できるパラメーター

パラメーター	説明	例
bug	Bugzilla 番号での検索。	bug = 1172165
cve	CVE 番号での検索。	cve = CVE-2015-0235
id	エラータ ID での検索。自動提案システムにより、入力時に利用可能な ID の一覧が表示されます。	id = RHBA-2014:2004
issued	発行日による検索。 "Feb16,2015" などの正確な日付を指定したり、 "Yesterday" 、または "1 hour ago" などのキーワードを使用したりできます。時間の範囲は "<" および ">" 演算子を使用して指定できます。	issued < "Jan 12, 2015"
package	完全なパッケージビルド名による検索。自動提案システムにより、入力時に利用可能なパッケージの一覧が表示されます。	package = glib2-2.22.5-6.el6.i686
package_name	パッケージ名による検索。自動提案システムにより、入力時に利用可能なパッケージの一覧が表示されます。	package_name = glib2

パラメーター	説明	例
severity	セキュリティー更新によって修正された問題の重大度による検索。 Critical 、 Important 、または Moderate を指定します。	severity = Critical
title	アドバイザリーのタイトルによる検索。	title ~ openssl
type	アドバイザリーのタイプによる検索。 security 、 bugfix 、または enhancement を指定します。	type = bugfix
updated	最終更新日付による検索。 issued パラメーターと同じ形式を使用できます。	updated = "6 days ago"

4.2. エラータのコンテンツホストへの適用

以下の手順では、1 つ以上のエラータをコンテンツホストに適用する方法を示しています。

手順4.2 単一エラータをコンテンツホストに適用するには、以下を実行します。

1. コンテンツ → エラータ に移動して、利用可能なエラータの一覧を表示します。
2. 適用する必要がある エラータ ID をクリックします。
3. コンテンツホスト タブで、更新する 1 つ以上のコンテンツホストを選択します。環境別に利用可能なコンテンツホストをフィルターし、名前ですれらを検索できます。ページの上にあるチェックボックスを選択すると、ライフサイクル環境にインストール可能なエラータをすでに含んでいるコンテンツホストのみが表示されます。
4. ホストに適用 をクリックします。
 - ※ エラータが **適用可能** な場合、コンテンツビューのマイナーバージョンが新たに作成されます。公開直後にエラータをコンテンツホストに適用 を選択する場合、Satellite は更新されたコンテンツビューのプロモート時に、エラータをコンテンツホストに自動的にインストールします。これを選択しない場合は、エラータがコンテンツホストにインストールできるように利用可能になります。インストール可能なエラータについては同じ手順で後で適用するか、または [手順4.4「インストール可能なエラータをコンテンツホストに適用するには、以下を実行します。」](#) で説明されているようにコンテンツホストごとに手動で適用できます。
 - ※ エラータが **インストール可能** の場合、つまりエラータが選択されたコンテンツホストのライフサイクル環境にすでにあるものの、まだインストールされていない場合は、コンテンツビューの新規バージョンは作成されません。
5. 確定します (Confirm) をクリックします。

手順4.3 複数のエラータをコンテンツホストに適用するには、以下を実行します。

1. コンテンツ → エラータ に移動して、利用可能なエラータの一覧を表示します。
2. エラータ ID フィールドの左側にあるチェックボックスを選択して適用するエラータを選択します。
3. エラータを適用 をクリックして、すべての選択したエラータを適用します。
4. 更新する 1 つ以上のコンテンツホストを選択します。環境別に利用可能なコンテンツホストをフィルターし、名前ですれらを検索できます。ページの上にあるチェックボックスを選択する場合、ライフサイクル環境にインストール可能なエラータをすでに含んでいるコンテンツホストのみが表示されます。

5. **次へ** をクリックします。選択したエラータの一部が **適用可能** である場合、コンテンツビューのマイナーバージョンが新たに作成されます。**公開直後にエラータをコンテンツホストに適用** を選択する場合、Satellite は更新されたコンテンツビューのプロモート時に、エラータをコンテンツホストに自動的にインストールします。インストール可能なエラータのみが選択されている場合、それらはコンテンツビューの新規バージョンを作成せずにインストールされます。

コンテンツホストのライフサイクル環境にインストール可能なエラータが含まれる場合、それらを [手順 4.4 「インストール可能なエラータをコンテンツホストに適用するには、以下を実行します。」](#) にあるように **コンテンツホスト** ページからインストールできます。エラータは、コンテンツホストを管理するパーミッションを持つものの、より高いレベルでエラータ管理を行う権利が与えられていないユーザーが適用できます。同様に [「エラータのホストコレクションへの追加」](#) に説明されているように、インストール可能なエラータはホストコレクションに適用することができます。

手順4.4 インストール可能なエラータをコンテンツホストに適用するには、以下を実行します。

1. **ホスト** → **コンテンツホスト** に移動します。
2. 管理するコンテンツホストの名前をクリックします。
3. **エラータ** タブで、インストールするエラータを選択します。
4. **選択を適用** をクリックして選択した更新をインストールします。

4.3. エラータ通知のサブスクリプション

『サーバー管理ガイド』の [電子メール通知の設定](#) で説明されているように、Satellite ユーザーのメール通知を設定することができます。ユーザーは、コンテンツビューのプロモート時やリポジトリの同期後に、適用可能およびインストール可能なエラータの概要を受信できます。

第5章 セキュリティーコンプライアンス管理

セキュリティーコンプライアンス管理は、セキュリティーポリシーの定義、それらのポリシーへのコンプライアンスの監査、および非コンプライアンスのインスタンスの解決などを行う継続的なプロセスです。セキュリティーポリシーが定義されると、ポリシーへのコンプライアンスを検証するために監査が行われます。非コンプライアンス状況は、組織の設定管理ポリシーに基づいて管理されます。セキュリティーポリシーは、ホスト固有のものから業界共通のものまで、その範囲は多岐に及びます。そのため、ポリシーは柔軟に定義することが求められます。

Security Content Automation Protocol (SCAP) は、セキュリティー設定ポリシーの定義を有効にします。たとえば、セキュリティーポリシーは、Red Hat Enterprise Linux を実行するホストの場合に SSH 経由のログインを **root** アカウントに許可しないように指定することができるかもしれません。Satellite 6 では、OpenSCAP プロジェクトで使用されるツールは、セキュリティーコンプライアンスの監査を実施するために使用されます。OpenSCAP についての詳細は、[Red Hat Enterprise Linux 7 セキュリティーガイド](#) を参照してください。Satellite Web UI を使用して、Red Hat Satellite の管理下にあるすべてのホストに対してスケジュールされるコンプライアンス監査およびレポート作成を実行できます。

以下の仕様が OpenSCAP でサポートされています。

- XCCDF: Extensible Configuration Checklist Description Format (バージョン 1.2)
- OVAL: Open Vulnerability and Assessment Language (バージョン 5.11)
- Asset Identification (バージョン 1.1)
- ARF: Asset Reporting Format (バージョン 1.1)
- CCE: Common Configuration Enumeration (バージョン 5.0)
- CPE: Common Platform Enumeration (バージョン 2.3)
- CVE: Common Vulnerabilities and Exposures
- CVSS: Common Vulnerability Scoring System (バージョン 2.0)

5.1. SCAP について

5.1.1. SCAP コンテンツ

SCAP コンテンツは、ホストのチェックに使用される設定およびセキュリティーベースラインが含まれるデータストリーム形式のコンテンツです。チェックリストは *extensible checklist configuration description format* (XCCDF) および *open vulnerability and assessment language* (OVAL) の脆弱性に記述されます。ルールとしても知られるチェックリスト項目は、システム項目の必要な設定を表します。たとえば、どのユーザーも **root** ユーザーアカウントを使用して SSH 経由でホストにログインできないように指定することができます。ルールは 1 つ以上の *プロファイル* に分類でき、複数のプロファイルで 1 つのルールを共有できるようにすることができます。SCAP コンテンツはルールとプロファイルの両方で構成されています。

SCAP コンテンツは、作成することも、ベンダーから取得することも可能です。Red Hat Enterprise Linux 用にサポートされるプロファイルは *scap-security-guide* パッケージで提供されます。SCAP コンテンツの作成については本ガイドで扱いませんが、独自のコンテンツをダウンロードし、デプロイし、変更し、作成する方法についての詳細は、[Red Hat Enterprise Linux 7 セキュリティーガイド](#) または [Red Hat Enterprise Linux 6 セキュリティーガイド](#) を参照してください。Red Hat Enterprise Linux と共に提供される SCAP コンテンツは SCAP 仕様 1.2 に準拠しています。

Satellite 6 の OpenSCAP コンポーネントと共に提供されるデフォルトの SCAP コンテンツは、Red Hat Enterprise Linux のバージョンによって異なります。

- ※ Red Hat Enterprise Linux 6 には、Red Hat Enterprise Linux 6 のコンテンツがインストールされています。
- ※ Red Hat Enterprise Linux 7 には、Red Hat Enterprise Linux 6 および Red Hat Enterprise Linux 7 の両方のコンテンツがインストールされています。

5.1.2. XCCDF プロファイル

XCCDF プロファイルは、ホストまたはホストグループの評価に使用されるチェックリストです。プロファイルは通常、業界標準またはカスタム標準にかかわらず、標準への準拠を確認するために作成されます。

利用可能なプロファイルを一覧表示するには、Satellite Web UI を開いた後に、ホスト → **Policies (ポリシー)** に移動し、関連するポリシーの横にあるドロップダウンリストから **編集** を選択し、**SCAP コンテンツ** タブを選択します。次に、関連のある **SCAP コンテンツ** を選択してから **XCCDF プロファイル** ドロップダウンリストで利用可能なプロファイルを閲覧できます。

Satellite 6 で提供されるプロファイルは、<https://fedorahosted.org/scap-security-guide> でホストされる SCAP セキュリティーガイドプロジェクトから取得できます。

5.1.3. コンプライアンスポリシー

コンプライアンスポリシーとしても知られるスケジュールされる監査は、XCCDF プロファイルに対して指定したホストのコンプライアンスをチェックするスケジュールタスクです。スキャンが実行されるスケジュールは Satellite Server で指定されますが、スキャン自体はホストで実行されます。スキャンが完了すると、*Asset Reporting File (ARF)* が XML 形式で生成され、Satellite Server にアップロードされます。スキャンの結果はコンプライアンスポリシーダッシュボードで確認できます。コンプライアンスポリシーはスキャンされるホストを変更しません。OpenSCAP コンテンツには、いくつかのプロファイルと関連付けられたルールが含まれますが、デフォルトでポリシーは含まれません。

5.2. インストール

5.2.1. OpenSCAP パッケージのインストール

手順5.1 OpenSCAP パッケージのインストール

OpenSCAP プラグインおよびコンテンツを Satellite Server およびすべての外部 Capsule Server にインストールします。

1. Satellite Server では、OpenSCAP プラグインおよびコンテンツをインストールします。

- a.

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

インストールが成功したかどうかは進行状況のインディケーターによって示され、**Success!** が表示されます。OpenSCAP プラグインは、Satellite web UI のホストメニューにある **Compliance (コンプライアンス)** セクションに追加され、以下のページが含まれます。

- ※ **Policies (ポリシー)**
- ※ **SCAP Contents (SCAP コンテンツ)**
- ※ **レポート**

- b.

```
# yum install puppet-foreman_scap_client
```

- すべての外部 Capsule Server で、OpenSCAP プラグインおよびコンテンツをインストールします。



注記

OpenSCAP 機能が Capsule Server で有効にされる場合、Puppet はそのサーバーですでに有効にされている必要があります。

```
# satellite-installer --enable-foreman-proxy-plugin-openscap
```

インストールが成功したかどうかは進行状況のインディケーターによって示され、**Success!** が表示されます。これにより、OpenSCAP スキャンを実行するためのホストのセットアップに必要な Puppet クラスが提供され、自動化されたコンプライアンスのスキャン用に Cron ジョブが作成されます。

- Puppet マスターロールが設定された外部 Capsule Server では、OpenSCAP クライアントをインストールします。

```
# yum install puppet-foreman_scap_client
```

関連のある外部 Capsule Server を特定するには、Satellite web UI を開いてから、**インフラストラクチャー → Capsules** に移動してから **機能** 列に一覧表示されている **Puppet** が設定された外部の Capsule Server を特定します。

5.2.2. デフォルト OpenSCAP コンテンツのロード

手順5.2 デフォルト OpenSCAP コンテンツのロード

- ※ OpenSCAP コンテンツを Satellite Server にロードします。

```
# foreman-rake foreman_openscap:bulk_upload:default
```

5.2.3. OpenSCAP Puppet モジュールのインポート

手順5.3 OpenSCAP Puppet モジュールのインポート

- OpenSCAP には Puppet 環境が必要ですが、デフォルトではそれらは Puppet モジュールを含むコンテンツビュー用にのみ作成されます。利用可能な Puppet 環境を一覧表するには、Satellite web UI を開いた後に、**設定 → 環境** に移動します。

Puppet 環境がない場合は、Satellite Server で CLI セッションを開き、**本番用 Puppet 環境**のディレクトリーを作成します。

```
# mkdir -p /etc/puppet/environments/production/modules
```

- OpenSCAP コンテンツを選択した Puppet 環境にインポートします。OpenSCAP で監査されるそれぞれのホストは Puppet 環境に関連付けられる必要があります。
 - Satellite web UI では、コンテキストメニューから **組織** および **ロケーション** を選択します。
 - 設定 → 環境** に移動します。

- c. インポート をクリックしてから **satellite.example.com** からインポート をクリックします。
- d. OpenSCAP を使用して監査が実行されるホストに関連付けられたそれぞれの Puppet 環境について、チェックボックスを選択してから **更新** をクリックします。他の Puppet 環境が存在しない場合は **本番** 環境を選択します。

foreman_scap_client Puppet モジュールが選択した環境に追加されます。

- e. **foreman_scap_client** Puppet モジュールが追加されていることを確認します。

設定 → 環境 に移動してから、Puppet 環境の列のある **クラス** をクリックします。この手順は、**foreman_scap_client** Puppet クラスが一覧表示されている場合に成功します。

5.2.4. 追加の SCAP コンテンツのアップロード

追加の SCAP コンテンツは、各自で作成したものか他から取得したものを問わず、Satellite Server にアップロードできます。SCAP コンテンツは、ポリシーに適用される前に Satellite Server にインポートされる必要があります。たとえば、Red Hat Enterprise Linux 7.2 リポジトリで利用可能な *scap-security-guide* RPM パッケージには、*Payment Card Industry Data Security Standard*(PCI-DSS) バージョン 3 のプロファイルが含まれます。このコンテンツは、オペレーティングシステムのバージョンに固有ではないため、Red Hat Enterprise Linux 7.2 を実行していない場合でも Satellite Server にアップロードできます。

手順5.4 追加の SCAP コンテンツのアップロード

1. Satellite Server Web にログインします。
2. ホスト → **SCAP contents (SCAP コンテンツ)** に移動して、**新規 SCAP コンテンツをアップロード** をクリックします。
3. **タイトル** テキストボックスにタイトルを入力します。例: **RHEL 7.2 SCAP Content**
4. **Choose file (ファイルの選択)** をクリックしてから、SCAP コンテンツが含まれるロケーションに移動し、**Open (開く)** を選択します。
5. **送信** をクリックします。

SCAP コンテンツファイルが正常にロードされると、**Successfully created RHEL 7.2 SCAP Content (RHEL 7.2 SCAP コンテンツが正常に作成されました)** と同様のメッセージが表示され、**SCAP Contents (SCAP コンテンツ)** の一覧に新規のタイトルが含まれます。

5.3. コンプライアンスポリシーの管理

5.3.1. ポリシーの作成

以下の手順に従ってコンプライアンスポリシーを作成します。ここでは、指定された時間にロケーションおよびホストまたはホストグループのいずれかに適用される SCAP コンテンツとプロファイルを指定します。

手順5.5 ポリシーを作成するには、以下を実行します。

1. Satellite web UI で **ホスト → Policies (ポリシー)** に移動し、**New Compliance Policy (新規コンプライアンスポリシー)** をクリックしてウィザードの手順に従います。
2. このポリシーの名前、説明 (オプション) を入力してから **次へ** をクリックします。
3. 適用する SCAP コンテンツおよび XCCDF プロファイルを選択してから **次へ** をクリックします。

4. ポリシーの適用がスケジュールされた時間を指定してから **次へ** をクリックします。

Period (期間) ドロップダウンリストから **毎週**、**毎月**、または **Custom (カスタム)** を選択します。

- ※ **毎週** を選択する場合、**Weekday (平日)** ドロップダウンリストから必要な曜日も選択します。
- ※ **毎月** を選択する場合、**Day of month (月の日付)** フィールドで月の必要な日付も指定します。
- ※ **Custom (カスタム)** を選択する場合、**Cron 行** フィールドに有効な Cron 式を入力します。

Custom (カスタム) オプションでは、**毎週** または **毎月** のいずれかのオプションよりもポリシーのスケジュールに柔軟性を持たせることができます。

5. ポリシーが適用されるロケーションを選択してから **次へ** をクリックします。
6. ポリシーが適用される組織を選択してから **次へ** をクリックします。
7. ポリシーが適用されるホストグループを選択してから **次へ** をクリックします。
8. **送信** をクリックします。

Puppet エージェントが選択したホストグループに属するホスト、またはポリシーが適用されているホストで実行される場合、OpenSCAP クライアントがインストールされ、Cron ジョブがポリシーの指定されたスケジュールと共に追加されます。**SCAP Content (SCAP コンテンツ)** タブでは、すべてのターゲットホストのディレクトリー `/var/lib/openscap/content/` に配信される SCAP コンテンツの名前を指定します。

5.3.2. ポリシーの表示

以下の手順に従って、特定の OpenSCAP コンテンツおよびプロファイルの組み合わせによって適用されるルールのプレビューを行います。これはポリシーを計画する際に便利です。

手順5.6 ポリシーを表示するには、以下を実行します。

1. Satellite web UI で、**ホスト** → **Policies (ポリシー)** に移動します。
2. **ガイドの表示** をクリックします。

5.3.3. ポリシーの編集

以下の手順に従ってポリシーを編集します。編集されたポリシーは、次に Puppet エージェントが Satellite Server で更新をチェックする際にホストに適用されます。これはデフォルトで 30 分ごとに実行されます。

手順5.7 ポリシーを編集するには、以下を実行します。

1. Satellite web UI で、**ホスト** → **Policies (ポリシー)** に移動します。
2. ポリシーの名前の右側にあるドロップダウンリストから、**編集** を選択します。
3. 必要な属性を編集します。
4. **送信** をクリックします。

編集されたポリシーは、次に Puppet エージェントが Satellite Server で更新をチェックする際にホストに適用されます。これはデフォルトで 30 分ごとに実行されます。

5.3.4. ポリシーの定義

以下の手順を実行して既存のポリシーを削除します。

1. Satellite web UI で、**ホスト** → **Policies (ポリシー)** に移動します。
2. ポリシーの名前の右側にあるドロップダウンリストから、**削除** を選択します。
3. 確認メッセージで **OK** をクリックします。

5.4. コンプライアンスのモニター

コンプライアンスのモニターは、監査が実行されていることを確認し、非コンプライアンスを確実に特定できるようにするための継続的なタスクです。Red Hat Satellite 6 は一元化したコンプライアンスのモニターと管理を可能にします。Satellite の管理下にあるホストについては、カスタムスケジュールに基づいてコンプライアンス状況のチェックが行われ、詳細が Satellite Server によって照合されます。コンプライアンスダッシュボードでは、ホストのコンプライアンスの概要を示し、ポリシーの範囲内の各ホストの詳細を表示できます。コンプライアンスレポートでは、各ホストの適用可能なポリシーへのコンプライアンスの詳細な分析を行います。この情報を利用して、各ホストが提示するリスクを評価し、ホストのコンプライアンスを確保するために必要なリソースをより効果的に管理できます。

SCAP を使用してコンプライアンスをモニターする際の共通の目的には以下が含まれます。

- ※ ポリシーコンプライアンスの表示
- ※ コンプライアンスの変更の検知

Satellite web UI は、これらの目的を達成するために必要なすべての情報を提供します。コンプライアンスポリシーダッシュボードでポリシーのコンプライアンス状況を検証します。コンプライアンスレポートの履歴を確認するか、または変更のメール通知をサブスクライブするかのいずれかによって、ポリシーコンプライアンスにおける変更を検出します。

5.4.1. コンプライアンスポリシーダッシュボード

コンプライアンスポリシーダッシュボードは、ホストのポリシーへのコンプライアンス状況の概要を示します。コンプライアンスポリシーのダッシュボードを表示するには、Satellite web UI を開いてから **ホスト** → **Policies (ポリシー)** に移動し、ポリシーの名前をクリックします。ダッシュボードでは以下の情報が提供されます。

- ※ ホストのポリシーへのコンプライアンス状況のハイレベルビューを表示するリングチャート
- ※ ホストのポリシーコンプライアンス状況についての統計の内訳 (表形式)
- ※ 各ホストのポリシーの最新レポートへのリンク

ダッシュボードビューは、ホストのコンプライアンス状況の統計的な概要を表示するため、コンプライアンスの管理をここから開始することができます。非コンプライアンスとして評価されたすべてのホストについては、**Failed (失敗)**の統計でコンプライアンスタスクの優先付けに使用できるメトリックを提供します。**未監査**として検出されたホストも、それらの状況が不明なために優先する必要があります。

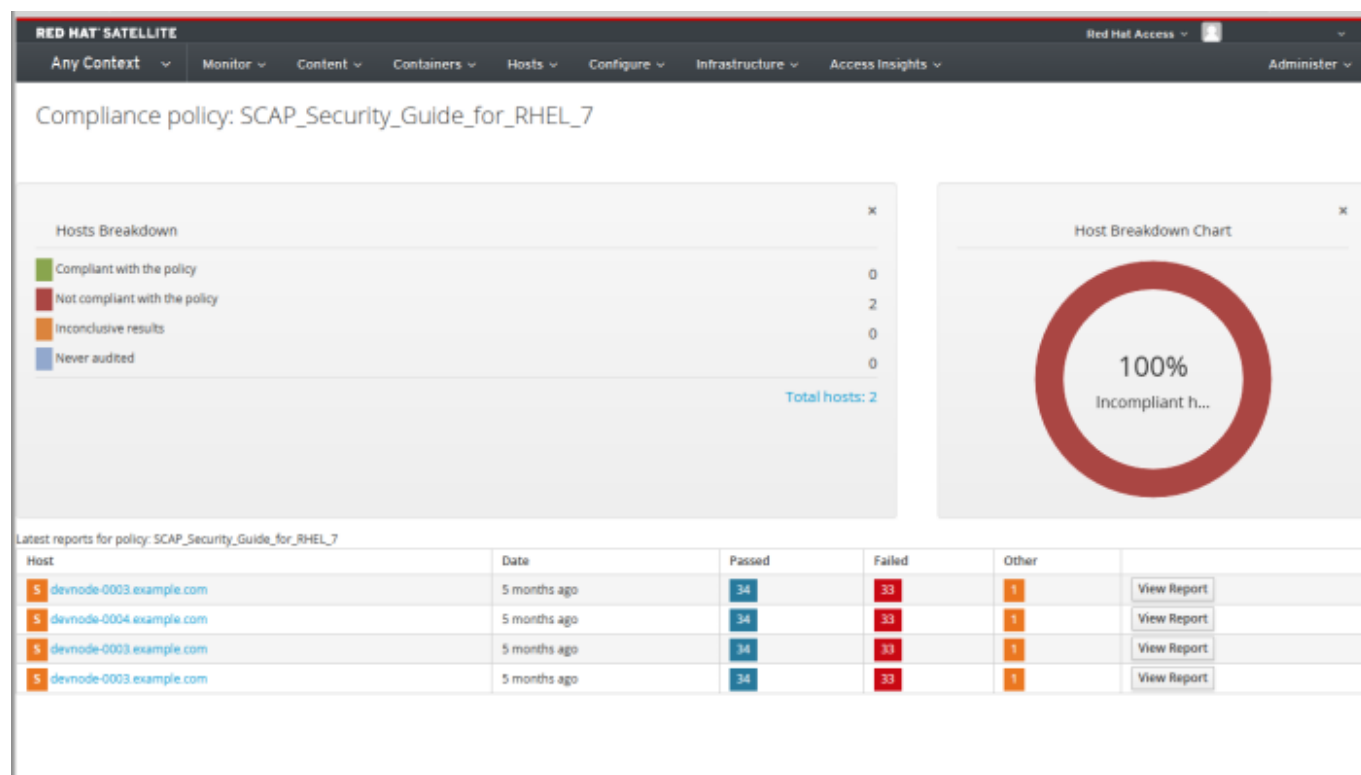


図5.1 コンプライアンスポリシーダッシュボード

5.4.2. コンプライアンスレポートの概要

コンプライアンスレポートは、ホストに対して実行されるポリシーの出力です。利用可能なすべてのコンプライアンスレポートを一覧表示するには、Satellite web UI を開いてから **ホスト** → **レポート** に移動します。各レポートについて、ポリシーごとに合格または不合格となったルールの合計数が一覧表示されます。デフォルトでは、すべてのレポートは日付の降順で一覧表示されます。並べ替え順序を変更するには、並べ替える列のラベルをクリックします。次に、その同じラベルをもう一度クリックして降順または昇順のいずれかに変更します。**Compliance Reports (コンプライアンスレポート)** ページで、**レポートを表示** をクリックして個別レポートを表示するか、または **検索** フィールドを使用してレポートの一覧をホストまたはホストのサブセットに制限します。コンプライアンスレポートを削除するには、**レポートを表示** の横にあるドロップダウンリストから **削除** を選択します。

ホストのポリシーコンプライアンスを管理する際に、一定期間におけるコンプライアンスの変更をモニターすることが役立ちます。Satellite 6 はコンプライアンスの変更を手動でモニターするのに必要な情報およびメールによる通知を提供します。[「コンプライアンスレポートの検索」](#) に詳述されているように、**検索** フィールドを使用してレポートの一覧を 1 つ以上のホストに制限し、変更を手動で評価します。『Server Administration Guide』の[Configuring Email Notifications](#) に詳述されているように、コンプライアンス変更についてのメールのメッセージをサブスクライブします。

Host	Date	Passed	Failed	Other	
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report
test-system.example.com	5 days ago	34	33	1	View Report

図5.2 コンプライアンスレポートの概要

5.4.3. コンプライアンスレポートの検索

コンプライアンスレポートの検索フィールドを使用して、レポートの一覧の範囲を制限することができます。ホストのサブセットに範囲を狭めることにより、最も必要とされるリソースを重点的に確認することができます。フィルターを適用するには、**検索** フィールドに検索条件を入力し、Enter を押すか、または **検索** をクリックします。実行される検索では大文字と小文字が区別されません。空の **検索** フィールドをクリックして利用可能な検索パラメーターの一覧を表示します。

利用可能なすべての検索演算子の詳細については、『サーバー管理ガイド』の [詳細な検索に対してサポートされる演算子](#) を参照してください。論理演算子の **and**、**not** および **has** を使用して複雑なクエリーを作成することができます。正規表現は有効な検索条件として使用できませんが、複数のフィールドを単一の検索式で 사용할 ことができます。

論理演算子

- ※ **not**: 式を否定します。
- ※ **has**: オブジェクトには指定したプロパティがなければなりません。
- ※ **and**: 検索条件を組み合わせます。

ユースケースの検索

以下の検索条件により、ルールが 6 回以上パスしなかったコンプライアンスレポートすべてを検索します。

```
failed > 5
```

以下の検索条件により、ホスト名に文字列 **prod-**。

```
host ~ prod- AND date > "Nov 5, 2015"
```

が含まれるホストについての、2015 年 11 月 5 日より後に作成されたすべてのコンプライアンスレポートが検索されます。

以下の検索条件により、1 時間前より `compliance_policy rhel7_audit` を使用して生成されたすべてのレポートが検索されます。

```
"1 hour ago" AND compliance_policy = date = "1 hour ago" AND
compliance_policy = rhel7_audit
```

すべての 利用可能なコンプライアンスレポートを再び一覧表示するには、**検索** 条件を削除してから Enter を押すか、または **検索** をクリックします。

検索のブックマーク

検索をブックマークして、同じ検索条件を再び適用することができます。

手順5.8 検索をブックマークするには、以下を実行します。

1. 検索条件を適用します。
2. **検索** 一覧から、**この検索をブックマーク** を選択します。
3. **名前** フィールドに記入します。

ブックマークをこの Satellite インスタンスの他のユーザーが利用できるようにするには、**公開** チェックボックスを選択します。

4. **送信** をクリックします。

ブックマークを使用するには、**ホスト → レポート** に移動し、**検索** ボタンの横にあるドロップダウン項目をクリックしてから、ブックマークをクリックします。

5.4.4. コンプライアンスレポートの表示

ホスト → レポート に移動してから、特定のホストの列にある **レポートを表示** をクリックします。

コンプライアンスレポートは以下のセクションで構成されています。

- Introduction (導入)
- Evaluation Characteristics (評価特性)
- Compliance and Scoring (コンプライアンスおよびスコアリング)
- Rule Overview (ルールの概要)

5.4.4.1. Evaluation Characteristics (評価特性)

このセクションでは、評価されたホスト、評価に使用されたプロファイル、および評価の開始と終了を含む、特定のプロファイルに対する評価についての詳細情報を提供します。参照用として ホストの IPv4、IPv6、および MAC アドレスも一覧表示されます。

Evaluation Characteristics (評価特性)

Target machine (ターゲットマシン)

評価されるホストの完全修飾ドメイン名 (FQDN) です。例: `test-system.example.com`

Benchmark URL (ベンチマーク URL)

ホストの評価に使用される SCAP コンテンツの URL です。例:
`/var/lib/openscap/content/1fbdc87d24db51ca184419a2b6f`

Benchmark ID (ベンチマーク ID)

ホストの評価に使用されるベンチマークの ID です。ベンチマークはプロファイルのセットです。
 例: `xccdf_org.ssgproject.content_benchmark_RHEL_7`

Profile ID (プロファイル ID)

ホストの評価に使用されるプロファイルの ID です。例:
`xccdf_org.ssgproject.content_profile_rht-ccp`

Started at (開始時刻)

ISO 8601 形式で示される評価が開始された日時です。例: `2015-09-12T14:40:02`

Finished at (終了日時)

ISO 8601 形式で示される評価の終了日時です。例: `2015-09-12T14:40:05`

Performed by (実行ユーザー)

評価をホストに実行したローカルのアカウント名です。例: `root`

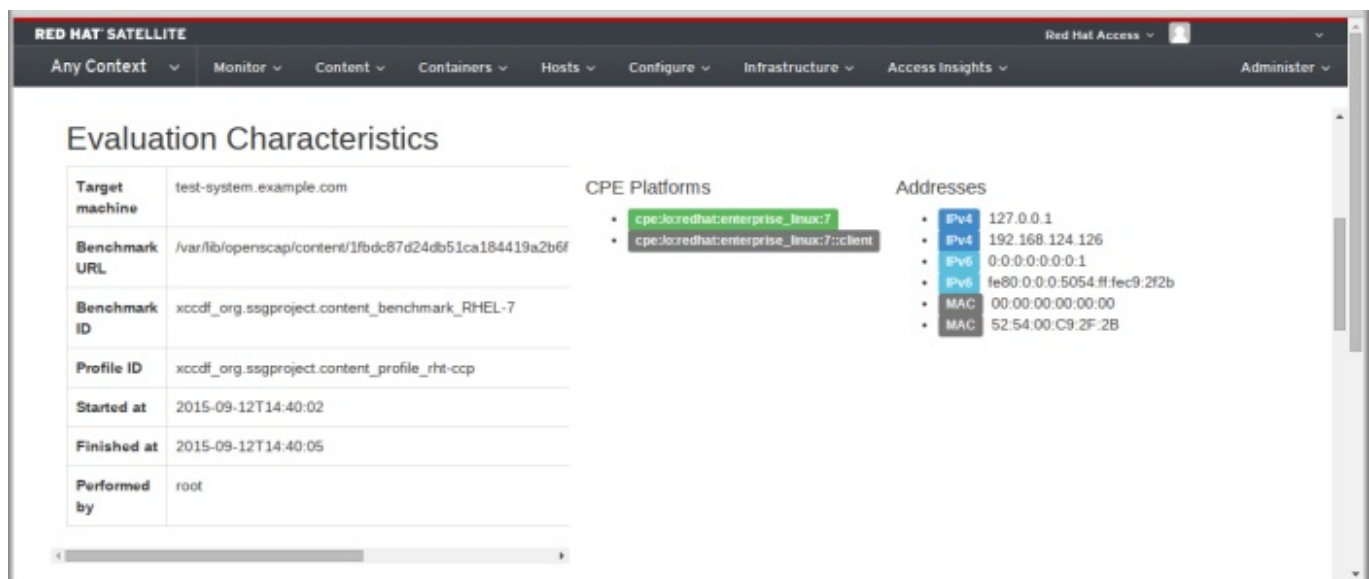


図5.3 Evaluation Characteristics (評価特性)

5.4.4.2. Compliance and Scoring (コンプライアンスおよびスコアリング)

このセクションでは、ホストがプロファイルのルールに準拠しているかどうかの概要、重大度別の非コンプライアンスの内訳、およびパーセンテージで示される全体のコンプライアンススコアを示します。ルールへのコンプライアンスがチェックされなかった場合には、**Rule results (ルール結果)**で **Other (その他)**と分類されます。

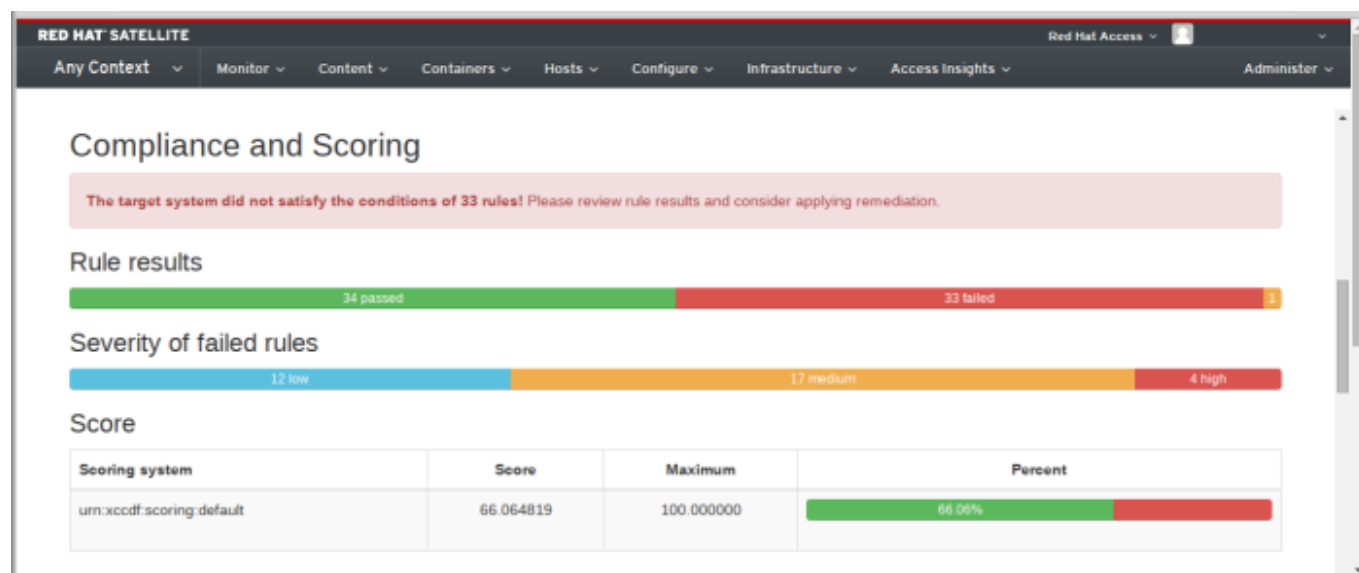


図5.4 Compliance and Scoring (コンプライアンスおよびスコアリング)

5.4.4.3. Rule Overview (ルールの概要)

このセクションでは、階層的に示されるルールと共にすべてのルールの詳細とコンプライアンスの結果を示します。

コンプライアンスレポートに組み込まれるルールの一覧を制限するためにチェックボックスを選択したり、クリアしたりします。たとえば、非コンプライアンスを重点的にレビューする場合には、**pass** および **informational** チェックボックスをクリアします。

すべてのルールを検索するには、**検索** フィールドに条件を入力します。検索は、入力時に動的に適用されます。**検索** フィールドは、単一のプレーンテキストの検索用語のみを受け入れ、それは大文字と小文字を区別しない検索に適用されます。検索の実行時には、説明が検索条件に一致するルールのみが一覧表示されます。検索フィルターを削除するには、検索条件を削除します。

それぞれの結果の説明については、**結果** 列に示される状況の上にカーソルを合わせます。

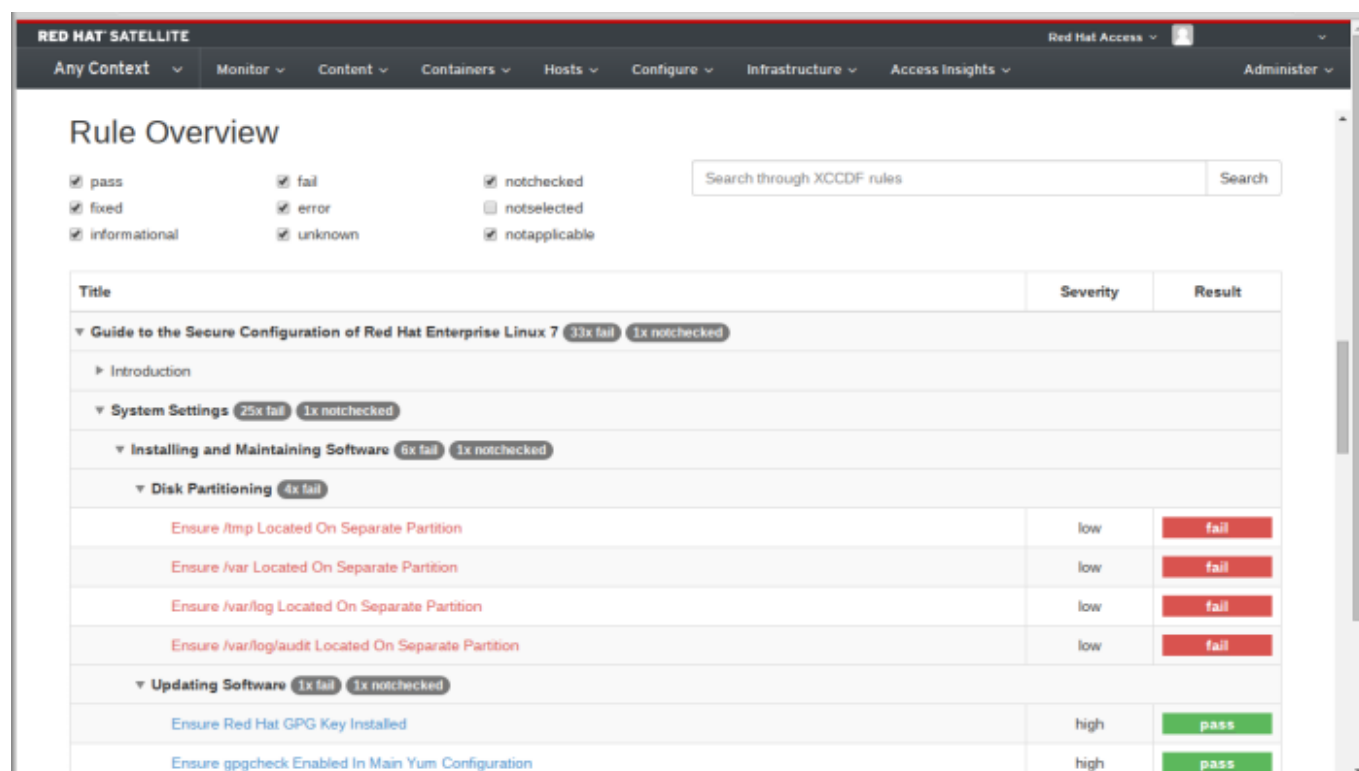


図5.5 Rule Overview (ルールの概要)

5.4.4.4. ルール結果の検査

ホストがルールのコンプライアンスに失敗した理由を判別するには、ルールのタイトルをクリックします。次に開かれるウィンドウでは、ルールの説明 (該当する場合は、ホストのコンプライアンスを確保する方法が含まれる)、ルールの根拠、および場合によっては修復スクリプトを含む追加情報が提供されます。

Set Password Minimum Age

Rule ID	xccdf_org.ssgproject.content_rule_accounts_minimum_age_login_defs
Result	fail
Time	2015-08-12T14:40:05
Severity	medium
Identifiers and References	<p>identifiers CCE-27002-5</p> <p>references (A-5(f), (A-5(1)(d)), 198, 75,</p>
Description	<p>To specify password minimum age for new accounts, edit the file <code>/etc/login.defs</code>, and add or correct the following line, replacing <code>DAYS</code> appropriately:</p> <pre>PASS_MIN_DAYS DAYS</pre> <p>A value of 1 day is considered for sufficient for many environments. The DoD requirement is 1.</p>
Rationale	Setting the minimum password age protects against users cycling back to a favorite password after satisfying the password reuse requirement.

Q&A details

Items found violating Tests the value of PASS_MIN_DAYS in /etc/login.defs:

Path	Content
/etc/login.defs	PASS_MIN_DAYS 0

Remediation script

```
var_accounts_minimum_age_login_defs="2"
grep -q ^PASS_MIN_DAYS /etc/login.defs && \
sed -i "s/PASS_MIN_DAYS.*/PASS_MIN_DAYS $var_accounts_minimum_age_login_defs/g" /etc/login.defs
if [ ! $? -eq 0 ]; then
```

図5.6 ルール評価の結果



警告

推奨される修復操作やスクリプトのいずれについても、まずそれらを本番以外の環境でテストしてから実装するようにしてください。

5.4.5. コンプライアンスのメール通知

Satellite Server は、**OpenSCAP policy summary (OpenSCAP ポリシー概要)** のメール通知をサブスクライブするすべてのユーザー宛に OpenSCAP 概要メールを送信します (『サーバー管理ガイド』の [電子メール通知の設定](#) を参照)。ポリシーが実行されるたびに、Satellite は直前の実行との比較で結果をチェックし、それらの間に変更がないかどうかを確認します。メールは各サブスクライバーがリクエストする頻度で送信され、各ポリシーの概要と最近の結果の情報を提供します。

OpenSCAP の概要 メールメッセージには、以下の情報が含まれます。

- 対象とする期間の詳細。
- すべてのホストの合計 (状況別): 変更済み、準拠、および非準拠。
- 各ホストの表形式の内訳と、合格、失敗、変更済み、または結果が不明な場合などのルールの合計を含む最新ポリシーの結果。

以下は、OpenSCAP 概要のメールメッセージの例になります。

RED HAT SATELLITE OPENSAP SUMMARY

Summary from about 3 hours ago to now

Summary report from Red Hat Satellite server at <http://satellite.example.com>

0 Changed	1 Compliant	1 Incompliant
--------------	----------------	------------------

Policies with hosts:

SCAP_Security_Guide_for_RHEL_7

Hostname	Passed	Failed	Other	Changed?
host1.example.com	No ARF reports for this policy			
host2.example.com	No ARF reports for this policy			
host3.example.com	1	25	2	No

Total of 3 hosts

第6章 コンテナの使用

Docker は、Linux コンテナ内でのアプリケーションの導入を自動化し、アプリケーションにライントタイム依存パッケージを持たせて 1 つのコンテナにパッケージ化する機能を提供するオープンソースプロジェクトです。Linux コンテナでは安全性を強化する一方、迅速なアプリケーション導入およびテスト、メンテナンス、トラブルシューティングなどの簡略化を可能にします。詳細については、Red Hat カスタマーポータル ^[1] の [Get Started with Docker Formatted Container Images on Red Hat Systems \(Docker フォーマットのコンテナイメージの使用法\)](#) についての記事を参照してください。

Docker フォーマットのコンテナは以下の部分で構成されています。

- ※ **コンテナ**: アプリケーションサンドボックスです。各コンテナは必要な設定データを保持するイメージをベースとしています。イメージからコンテナを起動すると、書き込み可能なレイヤーがイメージの上部に追加されます。コンテナをコミットする際はいつでも、変更を保存するために新規のイメージレイヤーが追加されます。
- ※ **イメージ**: 決して変更されることのない、コンテナの設定の静的なスナップショットです。コンテナへの変更は新規イメージレイヤーを作成することによってのみ保存できます。各イメージは 1 つまたは複数の親イメージに依存します。
- ※ **プラットフォームイメージ**: 親を持たないイメージです。プラットフォームイメージは、ランタイム環境、パッケージ、コンテナ化されたアプリケーションの実行に必要なユーティリティなどを定義します。プラットフォームイメージは書き込みができないため、変更はすべて上に重ねられるコピーイメージに反映されます。Red Hat Satellite から Red Hat Enterprise Linux プラットフォームイメージにアクセスする方法については、[例6.1 「Satellite での Red Hat Enterprise Linux コンテナの作成」](#) を参照してください。
- ※ **レジストリー**: ダウンロードできるイメージが含まれる公開または非公開のアーカイブです。一部のレジストリーを使うと、イメージをアップロードして他のユーザーに利用可能にすることができます。Red Hat Satellite では、ローカルおよび外部レジストリーからイメージをインポートすることができます。Satellite 自体はホストのイメージレジストリーとして機能できますが、ホストは変更をレジストリーに戻すことはできません。詳細は、[「コンテナの作成」](#) を参照してください。
- ※ **タグ**: リポジトリにあるイメージを、通常はイメージに保存されたアプリケーションのバージョンで区別するために使用されるマークです。リポジトリは、コンテナレジストリーの同様のイメージを分類するために使用されます。イメージには固有の英数字の ID のみを設定できるので、`repository:tag` などの形式により、人間の読み取り可能な方法でイメージを識別できます。詳細は、[「コンテナタグの使用」](#) および [「リポジトリの管理」](#) を参照してください。

Red Hat Satellite では、オンプレミスレジストリーを作成し、複数のソースからイメージをインポートし、コンテンツビューを使用してそれらをコンテナに配信することができます (イメージをコンテンツビューにロードする方法の詳細については、[「リポジトリのコンテンツビューへの追加」](#) を参照してください)。Satellite は、コンテナを実行するためのサーバーとして機能する 1 つ以上の Docker コンピュートリソースの作成をサポートします。これにより、イメージをインポートし、このイメージをベースとしたコンテナを開始し、コンテナのアクティビティをモニターし、その状態を新規のイメージレイヤーにコミットしてさらに伝播するようにできます。

6.1. コンテナの管理

以下のセクションでは、コンテナを作成し、表示し、開始し、停止し、コミットする方法について示します。

前提条件

Red Hat Satellite では、コンテナを Docker プロバイダータイプのコンピュートリソースのみにデプロイできます。そのため、コンテナの表示または作成の初回の試行時に、Satellite によって Docker コンピュートリソースの作成を求めるプロンプトが出されます。これを実行するには、まずコンテナホストを作成し、このホストをコンピュートリソースとして指定します。

手順6.1 コンテナホストを準備するには、以下を実行します。

1. Red Hat カスタマーポータルにある [Get Started with Docker Formatted Container Images on Red Hat Systems \(Docker フォーマットのコンテナイメージの使用方法\)](#) ガイドの

Getting Docker in RHEL 7 (RHEL 7 の Docker の取得) セクション [2] で説明されているように、イメージをホストする Red Hat Enterprise Linux 7 サーバーを準備し、このサーバーで **docker** サービスを有効にします。コンテナホストは、Satellite Server と同じマシンにデプロイすることも、独立してデプロイすることもできます。



注記

現在、Red Hat Enterprise Linux 7 はコンテナホスト用にサポートされる唯一のシステムです。`docker` パッケージは `rhel-7-server-extras-rpms` リポジトリで利用できます。Red Hat Enterprise Linux 6 システムは、現在のところコンテナをホストするシステムとしてサポートされていません。

2. コンテナホストで以下のコマンドを実行し、Satellite Server の CA 証明書をインストールします。

```
rpm -Uvh https://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

ここで、`satellite.example.com` は Satellite Server の完全修飾ドメイン名です。コンテナホストが Satellite ホストとしてすでに登録されている場合にはこの手順を省略します。

3. コンテナホストのロケーションに応じて以下のタスクを実行します。

A. コンテナホストが Satellite Server と同じマシンにある場合には、以下を実行します。

- a. `docker` ユーザーグループを作成し、`foreman` ユーザーをそれに追加します。

```
# groupadd docker
# usermod -aG docker foreman
```

- b. 以下のように `/etc/sysconfig/docker` ファイルの `OPTIONS` 変数を変更します。

```
OPTIONS='--selinux-enabled -G docker'
```

- c. 影響を受けるサービスを再起動して変更を適用します。

```
# systemctl restart docker.service
# katello-service restart
```

B. コンテナホストが Satellite Server 以外のマシンにある場合は、以下を実行します。

- a. コンテナホストのポートを開き、Satellite Server と通信します。これを実行するには、以下のように `/etc/sysconfig/docker` ファイルの `OPTIONS` 変数を変更します。

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

TLS が有効な場合はポート **2376** を使用できます。

- b. 以下のように docker サービスを再開し、設定を検証します。

```
# systemctl restart docker.service
# systemctl status docker.service
```

手順6.2 Docker コンピュートリソースを作成するには、以下を実行します。

1. ポート 5000 が Satellite Server で有効にされていることを確認します。コンテナホストはこのポートを使用して Satellite Server のコンテンツビューからイメージをプルします。
2. [「コンピュートリソース」](#) にあるようにコンピュートリソースを作成します。コンテナホストのロケーションに応じてリソース URL を指定します。
 - a. コンテナホストが Satellite Server と同じマシン上にある場合、**unix:///var/run/docker.sock** をリソース URL として設定します。
 - b. コンテナホストが Satellite Server 以外のマシンにある場合、以下の形式で URL を指定します。

```
http://container_host_fqdn:2375
```

ここで、*container_host_fqdn* はコンテナホストの完全修飾ドメイン名を表し、Satellite との通信用にコンテナホストで開かれるポート番号は **2375** か、TLS を使用している場合は **2376** にすることができます。

3. **テスト接続** をクリックしてコンテナホストが利用できるかどうかをテストします。
4. **送信** をクリックしてコンピュートリソースを作成します。

6.1.1. コンテナの作成

Satellite に 1 つ以上の Docker コンピュートリソースがある場合、コンテナを作成することができます。新規コンテナを作成するには、[手順6.3「コンテナを作成するには、以下を実行します。」](#) にある手順に従います。既存コンテナをモニターする方法については、[「コンテナのモニター」](#) を参照してください。

コンテナを作成するには、まずイメージをインポートする必要があります。これは、プラットフォームイメージまたは先に作成したレイヤー化したイメージである場合があります。Satellite は以下のイメージソースをサポートします。

- ※ **ローカルコンテンツ**: コンテナの作成時に **コンテンツビュー** オプションによって表されます。このオプションを使用すると、一部のコンテンツビューおよびライフサイクル環境で Capsule Server にすでに存在するリポジトリからイメージをインポートすることができます。ローカルレジストリーを作成し、これにデータを設定する方法についての詳細は、[「リポジトリの管理」](#) を参照してください。
- ※ **Docker ハブ**: Docker ハブレジストリーを検索して、そこからイメージをプルすることができます。検証済みのコンテンツを含む信頼されたイメージのみをプルするようにしてください。
- ※ **外部レジストリー**: 先に作成した外部レジストリーからイメージをインポートできます。レジストリーを Red Hat Satellite で作成する方法についての詳細は、[「外部レジストリーのインポート」](#) を参照してください。



注記

既存コンテナの設定を変更することはできません。設定を変更するには、[手順6.3「コンテナを作成するには、以下を実行します。」](#)で説明されているように、変更された設定を含む交換用のコンテナを作成する必要があります。そのため、実際のワークフローでコンテナが置き換え可能であることを確認してください。

手順6.3 コンテナを作成するには、以下を実行します。

1. コンテナ → 新規コンテナ に移動します。または、コンテナ → すべてのコンテナ に移動して 新規コンテナ をクリックします。

2. コンテナ作成の **Preliminary** ステージで、以下の設定を行います。

- ※ **コンピュートリソース** タブで、**Deployed on** ドロップダウンメニューからコンピュートリソースを選択します。コンピュートリソースの詳細については、[「コンピュートリソース」](#)を参照してください。

- ※ **ロケーション** タブで、新規コンテナを利用可能にするロケーションを選択します。

- ※ **組織** タブで、新規コンテナを利用可能にする組織を選択します。

次へ をクリックして進みます。

3. コンテナ作成の **イメージ** ステージでは、コンテナのベースとして機能するイメージをインポートします。これは、プラットフォームイメージの場合も、先に作成したレイヤー化されたイメージである場合もあります。以下のいずれかのオプションから選択してください。

- ※ **コンテンツビュー** タブを選択してイメージをライフサイクル環境からインポートします。ライフサイクル環境、コンテンツビュー、リポジトリ、タグおよび Capsule Server を指定します。

- ※ **Docker ハブ** タブを選択し、Docker ハブレジストリーからイメージをインポートします。イメージ名を **検索** フィールドに入力すると、Satellite はコンピュートリソースを自動的に検索します。眼鏡 (Looking Glass) のアイコンをクリックして Docker ハブを検索します。検索結果の一覧からイメージを選択し、ドロップダウンリストからタグを選択します。

- ※ **外部レジストリー** タブを選択し、既存のレジストリーからイメージをインポートします。ドロップダウンメニューからレジストリーを選択し、イメージ名でこれを検索します。Satellite は **タグ** フィールドに、選択したイメージ名に使用できるタグを設定します。詳細は、[「外部レジストリーのインポート」](#)を参照してください。

次へ をクリックして進みます。

4. コンテナ作成の **設定** ステージでは、以下の設定を行います。

- ※ コンテナ名を指定します。

- ※ コンテナ内で実行するコマンドを指定します。

- ※ コンテナが起動するとすぐに実行されるコマンドのエントリーポイントを指定します。デフォルトのエントリーポイントは **/bin/sh -c** です。

- ※ CPU をコンテナに割り当てます。たとえば、**0-2,16** は CPU 0, 1, 2、および 16 を表します。

- ※ コンテナに使用できる CPU 時間の相対的配分を定義します。

- ※ コンテナのメモリー制限を指定します。たとえば、**512m** により、コンテナのメモリー使用が 512 MB までに制限されます。

次へ をクリックして進みます。

5. **環境** というコンテナ作成の最終ステージでは、pseudo-tty を割り当てる必要があるかどうかを選択し、STDIN、STDOUT、および STDERR をコンテナに割り当てます。**環境変数の追加** をクリックしてコンテナのカスタム環境変数を作成します。**Run? (実行?)** チェックボックスを選択して作成後のコンテナを自動的に開始します。
6. **送信** をクリックしてコンテナを作成します。

コンテナの作成後に、Satellite はコンテナのメタデータの概要を表示します。デフォルトで、新規コンテナは無効にされています (コンテナの作成時に **Run? (実行?)** チェックボックスを選択していない場合)。コンテナの開始方法については、[手順6.5「コンテナを開始または停止するには、以下を実行します。」](#) を参照してください。

例6.1 Satellite での Red Hat Enterprise Linux コンテナの作成

Red Hat Enterprise Linux コンテナを Red Hat Satellite で有効にするには、以下の操作を実行します。

1. [「外部レジストリーのインポート」](#) で説明されているようにカスタムレジストリーを作成します。**registry.access.redhat.com** をレジストリー URL として指定します。
2. [「コンテナの作成」](#) で説明されているように新規コンテナを作成します。コンテナ作成の **イメージ** ステージで、**外部レジストリー** タブに移動し、直前の手順で作成されたレジストリーを選択します。検索フィールドを使用して Red Hat Enterprise Linux イメージの必要なバージョンを検索します。**設定** および **環境** ステージに進み、コンテナを完成します。

6.1.2. コンテナのモニター

Red Hat Satellite では、コンテナとコンテナの内部で実行されるプロセスの状態をモニターすることができます。一部のコンテナには **managed (管理)** というマークが付けられます。つまり、それらは Satellite 環境内で作成されており、プロビジョニングされていることを意味します。

以下の手順では、選択した組織を一覧表示し、コンテナのメタデータをモニターする方法について示します。

手順6.4 コンテナを調査するには、以下を実行します。

1. **コンテナ** → **すべてのコンテナ** に移動します。
2. **コンテナ** ページでは、すべての Docker コンピュートリソースに専用タブが付けられています。これらのタブのそれぞれには、各コンテナの選択されたパラメーターと共に利用可能なコンテナの表が含まれています。検査するコンピュートリソースのタブを選択します。
3. コンテナのメタデータを表示するには、検査するコンテナの名前をクリックします。Satellite ではコンテナのプロパティの表が表示されます。
4. **プロセス** タブでは、コンテナ内で現在実行されているプロセスを表示できます。プロセス名をクリックしてそのプロセスのメタデータを表示します。
5. コンテナが実行されている場合、その標準出力を **ログ** タブで確認できます。コンテナの作成時に **pseudo-tty** の割り当て チェックボックスを選択している場合、コンソールは対話的になります。そうでない場合は、コンテナの開始時に生成される初期の標準出力が表示されます。

6.1.3. コンテナの開始、コミットおよび削除

デフォルトで新規コンテナは無効にされます。コンテナを有効にして、コンピュータリソースでコンテナ化されたアプリケーションのプロセスを開始します。次にホストは、Web アプリケーションの場合と同様にコンテナと通信できます。以下の手順では、コンテナを開始し、停止する方法を示します。

手順6.5 コンテナを開始または停止するには、以下を実行します。

1. **コンテナ** → **すべてのコンテナ** に移動し、利用可能なコンテナの一覧を表示します。
2. 開始するコンテナの横にある **このマシンのパワーをオンにする** をクリックします。コンテナの開始後にボタンは **Power Off (パワーオフ)** に切り替わります。これにより、コンテナを停止することができます。これらの操作は **docker start** および **docker stop** コマンドと同等です。

以下の手順では、コンテナをコミットしてコンテナの状態を保存する新規のイメージレイヤーを作成します。

手順6.6 コンテナをコミットするには、以下を実行します。

1. **コンテナ** → **すべてのコンテナ** に移動し、利用可能なコンテナの一覧を表示します。
2. コミットするコンテナの名前をクリックします。
3. **コミット** をクリックします。Satellite は以下を実行するようにプロンプトを出します。
 - ※ リポジトリ名を指定します。これには単一の名前や、または **user/my-rhel-image** などのようにユーザー名と組み合わせを指定できます。
 - ※ タグをイメージに割り当てます。
 - ※ 問い合わせ先情報を指定します。
 - ※ イメージ情報のコメントを入力します。
4. **送信** をクリックします。



注記

コンテナは元のイメージのリポジトリにコミットされます。たとえば、コンテナが Docker ハブからプルされたイメージをベースにしている場合、コミットされた変更は Docker ハブに戻されます。

手順6.7 コンテナを削除するには、以下を実行します。

1. **コンテナ** → **すべてのコンテナ** に移動し、利用可能なコンテナの一覧を表示します。
2. 削除するコンテナの名前をクリックします。
3. **削除** をクリックします。
4. 警告ボックスで、**OK** をクリックしてコンテナを削除します。

6.2. リポジトリの管理

このセクションでは、コンテナイメージの内部リポジトリを作成する方法を示します。[「コンテナの作成」](#)で説明されているように内部リポジトリを使用してコンテナを作成することができます。

6.2.1. リポジトリの作成

リポジトリでは、Red Hat コンテンツ配信ネットワークまたはその他のソースのいずれかのコンテナコンテンツを同期することができます。

手順6.8 Docker リポジトリを作成するには、以下を実行します。

1. **コンテンツ** → **製品** に移動します。**新製品** をクリックし、製品名を指定してから **保存** をクリックします。
2. 直前の手順で作成した製品を選択し、**リポジトリ** タブに移動します。**リポジトリの作成** をクリックします。
3. リポジトリ名を指定し、**タイプ** ドロップダウンメニューから **docker** を選択します。これにより、追加のフィールドが表示されます。ここで、このレジストリーで同期するコンテンツソースの **URL** を指定します。**アップストリームリポジトリ名** フィールドでコンテンツソースからプルするリポジトリを指定します。



注記

以前のバージョンの Red Hat Satellite では、ローカルに保存されたコンテナイメージをリポジトリにアップロードすることができましたが、Red Hat Satellite 6.2 では、このアップロードはできなくなりました。

4. **保存** をクリックしてリポジトリを作成します。まず URL およびリポジトリ名が検証され、次にリポジトリが作成されます。

6.3. 外部レジストリーのインポート

以下の手順では、外部レジストリーを Satellite Server にインポートする方法を示します。

手順6.9 外部レジストリーをインポートするには、以下を実行します。

1. **コンテナ** → **レジストリー** に移動します。**New Registry (新規レジストリー)** をクリックします。
2. **レジストリー** タブで、レジストリーの名前および URL を指定します。これらの設定は必須です。オプションとして、レジストリーの簡単な説明を入力します。必要な場合は、レジストリーにアクセスするためのユーザー名およびパスワードを指定します。
3. **ロケーション** タブで、新規レジストリーを利用可能にするロケーションを選択します。
4. **組織** タブで、新規レジストリーを利用可能にする組織を選択します。
5. **送信** をクリックしてレジストリーを作成します。

6.4. イメージのコンピュートリソースへのインポート

イメージのインポートは、コンテナ作成時に必要な手順です。以下の手順に示されるように、コンテナの作成前にイメージをコンピュートリソースにインポートすることもできます。

手順6.10 イメージをコンピュートリソースにインポートするには、以下を実行します。

1. インフラストラクチャー → コンピュートリソース に移動し、コンピュートリソースの一覧を表示します。
2. 編集する docker コンピュートリソースを選択します。
3. 新規イメージ をクリックします。
4. イメージ名、オペレーティングシステム、アーキテクチャー、ユーザーの認証情報、および親イメージを含むイメージの詳細を指定します。ユーザーデータ を選択してこのイメージのユーザー入力を有効にします。
5. 送信 をクリックします。

6.5. コンテナタグの使用

タグは、コンテナ化されたアプリケーションの複数のバージョンを使用している場合などにはとくにイメージを整理するのに役立ちます。以下の手順は、タグを使用してイメージを検索する方法を示しています。

手順6.11 レジストリーをタグ別に検索するには、以下を実行します。

1. コンテンツ → **Docker タグ** に移動します。
2. 検索フィールドを使用してタグをイメージ名、タグまたはリポジトリ名でフィルターします。入力時に自動提案機能が実行されます。たとえば、以下のクエリーで、test_repo という名前のリポジトリのイメージに適用されるタグが検索されます。

```
repository = test_repo
```

3. 表示するタグの名前をクリックします。Satellite ではこのタグを使用するイメージの一覧を表示します。
4. イメージをクリックしてその環境およびコンテンツビューのバージョンを表示します。公開：フィールドには、コマンドラインでイメージをプルするのに使用できる URL が表示されます。

代替の比較演算子の一覧については、『サーバー管理ガイド』の[詳細な検索に対してサポートされる演算子](#)を参照してください。デフォルトでは、検索フィールドで入力文字列がタグ名として認識されます。たとえば、**centos** と入力してすべての centos タグを検索します。

[1] <https://access.redhat.com/articles/881893>

[2] <https://access.redhat.com/articles/881893#get>

第7章 アクティベーションキーの設定

アクティベーションキーはコンテンツホストの選択されたプロパティを定義します。コンテンツホストの登録時にアクティベーションキーを使用すると、プロセスをスピードアップし、単純化し、プロセスの一貫性を強化することができます。アクティベーションキーで以下を指定できます。

- 関連付けられるサブスクリプションおよびサブスクリプションの割り当て動作。
- 利用可能な製品およびリポジトリ。
- ライフサイクル環境およびコンテンツビュー。
- ホストコレクションのメンバーシップ。

十分なサブスクリプションがある限り、同じアクティベーションキーを複数のコンテンツホストに適用することができます。ただし、アクティベーションキーはコンテンツホストの初期設定のみを行います。コンテンツホストを組織に登録した後は、その組織が処理する他のコンテンツをコンテンツホストに手動で割り当てることができます。

コンテンツホストは、ホストの設定を定義するために組み合わせられる複数のアクティベーションキーに関連付けることができます。設定が競合する場合は、最後に指定されたアクティベーションキーが優先されます ([「アクティベーションキーを使用したホストの自動登録」](#) を参照してください)。

アクティベーションキーは、ホストが登録されている場合にのみ使用できます。アクティベーションキーに変更が加えられた場合、変更は、改訂されたアクティベーションキーでその後に登録されるホストにのみ適用されます。その変更は既存のホストには加えられません。

7.1. アクティベーションキーの作成

このセクションでは、アクティベーションキーを作成する方法について説明します。

手順7.1 アクティベーションキーを作成するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. **新規アクティベーションキー** をクリックします。以下の操作を実行します。
 - a. アクティベーションキーの名前を指定します。この設定は必須です。
 - b. オプションとして、アクティベーションキーに関連付けることのできるホストの数を制限する必要がある場合は **Unlimited Hosts** チェックボックスをクリアします。**制限** フィールドで数を指定します。
 - c. オプションとして、**説明** フィールドに適切な説明を入力します。また、このキーが適用される **環境** および **コンテンツビュー** を選択することもできます。
3. **保存** ボタンをクリックして、アクティベーションキーを作成します。

7.2. アクティベーションキーのサブスクリプションプロパティの定義

アクティベーションキーを使って、登録時にコンテンツホストをサブスクライブする方法を定義できます ([「アクティベーションキーを使用したホストの自動登録」](#) を参照してください)。アクティベーションキーで定義されるサブスクリプション動作は以下の 2 つの要因によって変わります。

1. サブスクリプションがアクティベーションキーに関連付けられているか？
2. 自動割り当てオプションは有効にされているか？

上記の要因に応じて、アクティベーションキーを使ったサブスクリプションとして 3 つのシナリオを想定することができます。

- ※ **サブスクリプションが指定されていないアクティベーションキー**: サブスクリプションが指定されておらず、自動割り当てが有効な場合に、アクティベーションキーを使用するホストは Satellite Server が提供する一覧から最適なサブスクリプションを検索します。これは、**subscription-manager --auto-attach** コマンドを実行する場合と同様です。
- ※ **自動割り当て用にカスタムサブスクリプションプールを指定するアクティベーションキー**: サブスクリプションが指定されていて自動割り当てが有効にされている場合、アクティベーションキーを使用するホストはアクティベーションキーで指定された一覧から最適なサブスクリプションを選択します。
- ※ **サブスクリプションセットが指定されたアクティベーションキー**: サブスクリプションが指定されており、自動割り当てが無効にされている場合、アクティベーションキーを使用するホストはアクティベーションキーに指定されたすべてのサブスクリプションに関連付けられます。

サブスクリプションを追加する方法については [手順7.2「サブスクリプションをアクティベーションキーに追加するには、以下を実行します。」](#) を参照し、自動割り当てを有効にする方法については、[手順7.4「アクティベーションキーで自動割り当てを有効にするには、以下を実行します。」](#) を参照してください。

注記

カスタム製品 (通常は Red Hat が提供しないコンテンツを含む製品) がアクティベーションキーに割り当てられている場合、この製品は、自動割り当ての設定の有無にかかわらず、登録されたコンテンツホストに対して常に有効にされます。

7.2.1. サブスクリプションの追加および削除

このセクションでは、サブスクリプションをアクティベーションキーに追加し、それらを削除する方法について説明します。

手順7.2 サブスクリプションをアクティベーションキーに追加するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 編集するアクティベーションキーの名前をクリックします。
3. **サブスクリプション** タブで、**追加** サブタブを選択します。
4. 利用可能なサブスクリプションの一覧から、追加するサブスクリプションを選択します。
5. **選択を追加** をクリックします。

手順7.3 アクティベーションキーからサブスクリプションを削除するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. アクティベーションキーの一覧が表示されます。サブスクリプションを削除するアクティベーションキーをクリックします。
3. **サブスクリプション** タブをクリックします。
4. **一覧/削除** サブに、割り当てられたサブスクリプションの一覧が表示されます。削除するサブスクリプションを選択します。
5. **選択を削除** をクリックします。

7.2.2. 自動割り当ての有効化

アクティベーションキーの自動割り当て設定は、登録時に自動的に割り当てられるサブスクリプションを決定します。

手順7.4 アクティベーションキーで自動割り当てを有効にするには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 編集するアクティベーションキーの名前をクリックします。
3. サブスクリプション タブをクリックします。
4. 自動割り当て の横にある編集アイコンをクリックします。
5. このチェックボックスを選択して自動割り当てを有効にします。
6. 保存 をクリックします。



注記

自動割り当てアクティベーションキーを使って仮想コンテンツホストを Satellite Server に登録するには、まず **virt-who** ユーティリティを使用してそれらのホストを、Virtual Datacenter (VDC) サブスクリプションのエントリーメントのあるハイパーバイザーにマップします。この前提条件が適用されない場合、仮想ホストは 24 時間の間に一時的な仮想サブスクリプションでのみ登録されます。詳細は、[Red Hat Satellite Virtual Instances Guide](#)を参照してください。

7.2.3. サービスレベルの設定

アクティベーションキーを設定して、アクティベーションキーを使って作成される新規ホストのデフォルトのサービスレベルを定義することができます。デフォルトのサービスレベルを設定すると、ホストに割り当てられる一致するサブスクリプションのみが選択されます。たとえば、アクティベーションキーのデフォルトのサービスレベルが Premium に設定されている場合、Premium サービスレベルのサブスクリプションのみが、登録時にホストに割り当てられます。

手順7.5 アクティベーションキーにサービスレベルを設定するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 編集するアクティベーションキーの名前をクリックします。
3. 詳細 タブをクリックします。
4. サービスレベル の横にある編集アイコンをクリックします。
5. ドロップダウンリストから必要なサービスレベルを選択します。ドロップダウンリストには、アクティベーションキーで利用できるサービスレベルのみが含まれます。
6. 保存 をクリックします。

7.3. アクティベーションキーの編集

このセクションでは、ホストコレクション、製品コンテンツ、および製品ライフサイクル環境など、アクティベーションキーのプロパティを編集する方法について説明します。

7.3.1. ホストコレクションの追加および削除

以下の手順は、ホストコレクションをアクティベーションキーに追加する方法を示しています。ホストコレクションはアクティベーションキーに関連付けて、アクティベーションキーを使用するホストが Satellite Server への登録時に関連付けられるホストコレクションに自動的に追加できるようにします。

手順7.6 ホストコレクションをアクティベーションキーに追加するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. ホストコレクションの追加先となるアクティベーションキーをクリックします。
3. **ホストコレクション** タブで **追加** サブタブをクリックし、利用可能なホストコレクションの一覧を表示します。
4. 追加するホストコレクションを選択してから、**選択を追加** をクリックします。

手順7.7 アクティベーションキーからホストコレクションを削除するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. アクティベーションキーの一覧が表示されます。ホストコレクションを削除するアクティベーションキーをクリックします。
3. **ホストコレクション** タブをクリックします。
4. **一覧/削除** サブタブで、アクティベーションキーに割り当てられたホストコレクションの一覧が表示されます。削除するホストコレクションのチェックボックスを選択します。
5. **選択を削除** をクリックして、アクティベーションキーからホストコレクションを削除します。

7.3.2. 製品コンテンツの編集

アクティベーションキーに利用できる製品の数、関連付けられるサブスクリプションによって決定されます。製品内での有効にするリポジトリの変更は、**製品コンテンツ** タグを使用して実行できます。

手順7.8 アクティベーションキーで製品コンテンツを編集するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 編集するアクティベーションキーの名前をクリックします。
3. **製品コンテンツ** タブをクリックして、サブスクリプションでアクティベーションキーに関連付けられた製品およびリポジトリを表示します。
4. 編集するリポジトリの横にある編集アイコンをクリックします。
5. ドロップダウンメニューから、リポジトリを有効にするか、または無効にするかどうかを選択します。**保存** をクリックして変更を適用します。

7.3.3. ライフサイクル環境およびコンテンツビューの設定

アクティベーションキーの作成時に、そのライフサイクル環境およびコンテンツビューを設定できます。以下の手順により、後にそれらの設定を変更することもできます。

手順7.9 アクティベーションキーのライフサイクル環境およびコンテンツビューを設定するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 編集するアクティベーションキーの名前をクリックします。
3. アクティベーションキーに関連付ける環境の横にあるチェックボックスをクリックします。ドロップダウンメニューからコンテンツビューを選択します。
4. **保存** をクリックします。

7.4. アクティベーションキーの削除

このセクションでは、アクティベーションキーを削除する方法について説明します。

手順7.10 アクティベーションキーを削除するには、以下を実行します。

1. コンテンツ → **Activation keys (アクティベーションキー)** をクリックします。
2. 削除するアクティベーションキーの名前をクリックします。
3. アクティベーションキーの詳細パネルの右上にある **削除** をクリックします。
4. 警告ボックスで **削除** をクリックしてキーを削除することを確認します。

7.5. アクティベーションキーを使用したホストの自動登録

以下の手順は、アクティベーションキーを使用してホストを自動登録する方法を示しています。アクティベーションキーが作成されたら、Satellite Server 上でのホストの登録時に **subscription-manager** ユーティリティを使用してこれを適用できます。インストールされる **subscription-manager** ユーティリティのバージョンは 1.10 以上である必要があることに注意してください。[「ホスト登録の設定」](#) に説明されているようにホストを準備してから、[手順7.11「アクティベーションキーを使ってホストを自動登録するには、以下を実行します。」](#) で説明されている手順に従います。

手順7.11 アクティベーションキーを使ってホストを自動登録するには、以下を実行します。

1. 古くなった登録データをシステムからクリアします。

```
# subscription-manager clean
```

2. Satellite Server からホストの CA 証明書のコピーをダウンロードし、これをインストールします。

```
# rpm -Uvh http://satellite.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```

3. システムを Satellite Server 上の必要な組織に登録します。--**activationkey** フラグを使用し、アクティベーションキーを使用してシステムを登録します。プロンプトが出されたらユーザー認証の詳細を入力します。

```
# subscription-manager register --org "Default_Organization" --activationkey "Test_Key"
```

4. システムが登録されると、リポジトリコンテンツにアクセスできるようになりますが、管理者は、Katello エージェントがクライアントシステムに登録されるまでパッケージおよびエラータの管理を行うことができません。これを実行するには、以下を実行します。


```
# yum install katello-agent
```

ホストを登録するための複数アクティベーションキーの組み合わせ

コンテンツホストの登録時に複数のアクティベーションキーを使用することができます。これにより、特定のサブスクリプションセット用にアクティベーションキーを作成してから、コンテンツホストの要件に従ってそれらを組み合わせることができます。たとえば、以下のコマンドは VDC および OpenShift サブスクリプションでコンテンツホストを ACME 組織に登録します。

```
# subscription-manager register --org "ACME" --activationkey "ak-VDC,ak-OpenShift"
```

アクティベーションキーの設定に競合が生じる場合、右端のキーが優先されます。たとえば、上記の **ak-OpenShift** アクティベーションキーの自動割り当て設定は **有効** にされており、**ak-VDC** キーの自動割り当て設定は **無効** にされている場合、コンテンツホストは自動割り当ての設定が **有効** の状態で登録されます。

第8章 GPG キーの設定

GPG キーにより、既存の GPG キーを、Satellite に追加するリポジトリと対になる Red Hat Satellite の製品とリポジトリに追加することができます。

8.1. GPG キーの作成

このセクションでは、GPG キーを Red Hat Satellite に追加する方法について説明します。

手順8.1 GPG キーを **Satellite** に追加するには、以下を実行します。

1. コンテンツ → **GPG キー** をクリックしてから **新規 GPG キー** をクリックします。
2. 名前 フィールドに、GPG キーの名前を入力します。
3. GPG キーファイルをアップロードするか、または GPG キーのコンテンツをテキストボックスに貼り付けます。
4. **保存** をクリックして GPG キーを Satellite に追加します。

8.2. GPG キーの削除

このセクションでは、Red Hat Satellite から GPG キーを削除する方法について説明します。

手順8.2 GPG キーを削除するには、以下を実行します。

1. コンテンツ → **GPG キー** をクリックします。
2. 削除する GPG キーをクリックしてから **GPG キーを削除** をクリックします。
3. 確認ボックスで **削除** をクリックして選択したキーを削除することを確認します。

第9章 プロビジョニング環境の設定

プロビジョニングは、ベアメタルシステムまたは新規の仮想コンピュートリソースから、完全に設定された使用できる状態にあるシステムまでを対象とするプロセスを指します。Red Hat Satellite は、数多くのホストの詳細に設定されるプロビジョニングを定義し、自動化する機能を提供します。このセクションでは、プロビジョニングに必要なコンポーネントを設定する方法を説明します。プロビジョニングプロセスについての詳細は、[Red Hat Satellite Provisioning Guide](#)を参照してください。

9.1. ホストグループの作成

ホストグループは、ホストがグループに配置される際に継承する一連のデフォルト値を定義します。複数のホストが 1 つのホストグループに属することが可能なだけでなく、複数のホストグループを階層にネストすることもできます。組織内のすべてのホストを表示する「ベース」または「親」ホストグループを作成してから、特定の設定を行うためにその親の下にネストされるホストグループまたは「子」ホストグループを作成することができます。このセクションでは、ホストグループを作成する方法を説明します。

手順9.1 ホストグループを Satellite に追加するには、以下を実行します。

1. **設定** → **ホストグループ** をクリックしてから **新規ホストグループ** をクリックします。
2. ホストグループの必要な詳細情報を入力してから、**送信** をクリックします。

ホストグループの属性

以下の表は、Satellite ホストグループに適用される属性について説明しています。

表9.1 ホストグループ属性の表

サブメニュー	オプション	説明
ホストグループ	親	新規ホストグループの親ホストグループです。
	名前	ホストグループの名前です。
	ライフサイクル環境	このホストグループを含む環境です。
	Puppet CA	Puppet CA サーバーに使用する Red Hat Satellite Capsule Server です。
	Puppet マスター	Puppet マスターとして使用する Red Hat Satellite Capsule Server です。
Puppet クラス	組み込み済みのクラス	ホストグループに組み込まれている Puppet クラスです。
ネットワーク	利用可能なクラス	ホストグループと共に使用できる Puppet クラスです。
	ドメイン	ホストグループ内のホストのドメインです。
オペレーティングシステム	サブネット	ホストグループ内のホストのサブネットです。
	アーキテクチャー	ホストグループ内のシステムのデフォルトアーキテクチャーです。
	オペレーティングシステム	ホストグループ内のシステムのデフォルトオペレーティングシステムです。
	メディア	オペレーティングシステムのインストールメディアのある場所です。
	パーティションテーブル	オペレーティングシステムをインストールするためのファイルシステムのパーティションレイアウトです。
	root パスワード	オペレーティングシステムの root パスワードです。

サブメニュー	オプション	説明
パラメーター	パラメーターの追加	ホストグループのパラメーターを設定するために名前と値のペアを提供します。
組織	組織	このホストグループを所有する組織です。
アクティベーションキー	コンテンツ環境	<code>@host.params['kt_activation_keys']</code> としてテンプレートで利用可能になるアクティベーションキーを定義します。

9.2. パラメーター

Red Hat Satellite のパラメーターは、ホストのプロビジョニング時に使用するキーと値のペアを定義します。これらのパラメーターは、Puppet の概念であるデフォルトスコープパラメーターに似ています。パラメーターは、Puppet でホストをセットアップする際に定義することができます。

パラメーターのタイプ

Red Hat Satellite には 2 種類のパラメーターがあります。

単純パラメーター

キーと値のペアの関係を定義する文字列パラメーターです。それらはユーザー設定で上書きされることはありませんが、Satellite のパラメーター階層に基づいて上書きされます。Global (グローバル)、organization-level (組織レベル)、location-level (ロケーションレベル)、domain-level (ドメインレベル)、operating system level (オペレーティングシステムレベル)、host group (ホストグループ)、および host (ホスト) パラメーターは単純パラメーターに含まれます。

スマートパラメーター

キーの値を定義するだけでなく、特定のオブジェクトタイプの条件付き引数、検証および上書きを可能にする複雑なパラメーターです。スマートパラメーターにより Puppet クラスは外部データを取得することができます。スマートパラメーターは、Puppet 用語で *パラメーター化クラス* と呼ばれる Puppet クラスで使用されます。それらのパラメーターの階層は Web UI で設定できます。

以下のパラメーター階層は単純なパラメーターに対して適用されます。

グローバルパラメーター

Satellite のすべてのホストに適用されるデフォルトパラメーターです。**設定 → グローバルパラメーター** で設定されます。

組織レベルのパラメーター

特定の組織のすべてのホストに影響を与えるパラメーターです。組織レベルのパラメーターはグローバルパラメーターを上書きします。**管理 → 組織 → 編集 → パラメーター** で設定されます。

ロケーションレベルのパラメーター

特定ロケーションのすべてのホストに影響を与えるパラメーターです。ロケーションレベルのパラメーターは組織レベルのパラメーターおよびグローバルパラメーターを上書きします。**管理 → ロケーション → 編集 → パラメーター** で設定されます。

ドメインパラメーター

特定ドメインのすべてのホストに影響を与えるパラメーターです。ドメインパラメーターはロケーションレベル以上のパラメーターを上書きします。インフラストラクチャー → ドメイン → **[choose_a_domain]** → パラメーター で設定されます。

オペレーティングシステムレベルのパラメーター

特定のオペレーティングシステムを持つすべてのホストに影響を与えるパラメーターです。オペレーティングシステムレベルのパラメーターは、ドメイン以上のパラメーターに影響を与えます。ホスト → オペレーティングシステム → **[choose_an_operating_system]** → パラメーター で設定されます。

ホストグループパラメーター

特定のホストグループのすべてのホストに影響を与えるパラメーターです。ホストグループパラメーターはオペレーティングシステムレベル以上のパラメーターを上書きします。設定 → ホストグループ → **[choose_a_host_group]** → パラメーター で設定されます。

ホストパラメーター

特定のホストに影響を与えるパラメーターです。従前より継承されるすべてのパラメーターは「パラメーター」サブタブに表示され、上書きすることができます。ホスト → すべてのホスト → 編集 → パラメーター で設定されます。

パラメーターと Puppet クラスの使用

Red Hat Satellite では、2 つの方法によって Puppet クラスで使用するためにホストの Puppet マスターに値を指定することができます。

スマート変数

スマートパラメーターを持たないクラス用に、キー/値の形式でグローバルパラメーターを Puppet マスターに指定するメソッドです。スマート変数は Puppet マニフェストの値の上書きを可能にします。それらは、クラスにスマートパラメーターがない場合や、グローバルパラメーターが必要となる特殊な場合に使用することが意図されています。また、複数の値を持たせることができますが、それらの値は階層のコンテキストやユーザーが適用する様々な条件によって異なります。これらの変数は Puppet がクラスをパラメーター化する前に存在していたものであり、下位互換性または検証が必要なグローバルパラメーターを使用し、グローバルパラメーターを特定の Puppet クラスにのみ適用できます。また文字列以外のタイプで使用できます (文字列タイプも使用できますが、その場合は単純パラメーターを使用すればよいことになります)。

パラメーター化クラス

スマートパラメーターに含まれる Puppet クラスです。これらのクラスは Puppet マスターからインポートされ、**\$::name** (推奨) または **\$name** などのパラメーターの名前がクラスを作成した人によって定義され、これを変更することはできません。これらのクラスにより、グローバルではなく特定クラスの変数の値を決定することができます。

設定されたパラメーターは各ホストの対応する YAML ファイルに組み込まれ、Puppet マスターに送信されます。YAML ファイルは、特定ホストのページの Web UI で表示できます。

す。**/etc/foreman/settings.yaml** 設定ファイルは **satellite-installer** コマンドの次回の実行時に上書きされるため、手動で変更することはできません。



重要

パラメーター化クラストのサポートは Satellite 6 でデフォルトで有効にされます。有効にされていることを確認する必要がある場合は、**管理 → 設定** に移動して **Puppet** タブをクリックし、**Parametrized_Classes_in_ENC** (ENC のパラメーター化クラス) が **True** に設定されているかどうかを確認できます。

9.2.1. グローバルな単純パラメーターの作成

この手順は、新規グローバルパラメーターを Satellite に追加する方法を示します。

手順9.2 グローバルな単純パラメーターを作成するには、以下を実行します。

1. **設定 → グローバルパラメーター** をクリックします。
2. **新規パラメーター** をクリックします。
3. パラメーターのキーの **名前** を入力します。
4. パラメーターの **値** を入力します。
5. オプションで、値を Web UI で非表示にするかどうかを選択します。
6. **送信** をクリックします。

9.2.2. スマート変数の設定

以下の手順では、Puppet クラスで値を上書きできるようにスマート編集を設定します。

手順9.3 スマート変数を設定するには、以下を実行します。

1. **設定 → Puppet クラス** をクリックします。
2. 一覧からクラスを選択します。
3. **スマート変数** タブをクリックします。これにより、新規の画面が表示されます。左側のセクションには、クラスがサポートする使用可能なパラメーターの一覧が含まれます。右側のセクションには、選択したパラメーターの設定オプションが含まれます。**変数の追加** をクリックして新規パラメーターを追加します。または、左側の一覧からパラメーターを選択します。
4. キー フィールドに名前を入力します。
5. **詳細** テキストボックスを編集して、プレーンテキストのメモを追加します。
6. 渡すデータの **キータイプ** を選択します。最も一般的なタイプは文字列ですが、他のデータタイプもサポートされています。
7. ホストが一致しない場合に Puppet Master に送信されるパラメーターの **Default Value** を入力します。
8. フィールドに作業中に表示したくないデータが含まれる場合、オプションで **非表示の値** を選択します。
9. **Optional Input Validator** セクションを使用してパラメーターに許可される値を制限します。**バリデータータイプ** (コンマ区切り値の **list** か、または正規表現の **regex** のいずれか) を選択し、許可される値または正規表現コードを **バリデータールール** フィールドに入力します。

10. **Prioritize attribute order** セクションは、条件付き引数に基づいて特定ホストの値を上書きするオプションを提供します。属性タイプおよびその値は *matcher* として知られています。

- a. エントリーを一覧に配置して、ホスト属性またはファクトが *Matcher* に対して評価される優先順位の **順序** を設定します。デフォルト一覧に追加することができます。*Matcher* 間で論理 AND 条件を作成するには、それらをコンマ区切りの一覧として一行に配置します。
- b. **Matcher の追加** をクリックして条件付き引数を追加します。一致する属性は **順序** 一覧のエントリーに対応するはずで、*Matcher* が設定されていない場合、デフォルト値のみが上書き機能として使用されます。

たとえば、Puppet マスターに指定する必要があるパラメーターの値が **server1.example.com** の完全修飾ドメイン名を持つホストの **test** の場合、*Matcher* を **fqdn=server1.example.com** として、**値** を **test** に設定します。

一致の手順は以下のとおりです。

- a. マッチャーがホスト属性の場合はそれを使用します。
- b. 該当する名前を持つ属性がない場合は、一致するホストパラメーター (パラメーターの階層に基づいて継承される) を探します。
- c. それでも一致しなければホストのファクトをチェックします。

Facter にある属性で、ホストの属性と混同しない属性を使用することが推奨されています。ホスト属性はホストパラメーターか、またはホストグループ、ドメイン、および組織などのホストの関連付けを使用することができます。マッチャーはいずれかのホストで固有である必要があります。たとえば、ホストが複数の config グループに属する可能性があるものの、ホストには 1 つのロケーションのみが設定され、ロケーションのファクトは有効なマッチャーとなるため、config グループのファクトを使用することはできません。

動的なデータは、*Embedded Ruby* (ERB) テンプレート構文の **値** フィールドでパラメーターおよび Puppet ファクトを使用すると利用できます。たとえば、Puppet ファクトを値の一部として使用するには、以下を実行します。

```
<%= @host.facts['network_eth0'] %>
```

利用可能な Puppet ファクトを一覧表示するには、**モニター → ファクト** に移動します。

11. **送信** をクリックして変更を保存します。

Puppet モジュールの使用についての詳細は、[Adding Puppet Modules to Red Hat Satellite 6](#)を参照してください。ERB 構文についての詳細は、[付録A テンプレート作成の参照](#)を参照してください。

9.2.3. Puppet マスターからのパラメーター化されたクラスのインポート

以下の手順では、Puppet マスターからパラメーター化されたクラスをインポートします。

注記

パラメーター化されたクラスのインポートは、Puppet モジュールが製品およびコンテンツビューで管理される場合に自動的に実行されます。

手順9.4 パラメーター化されたクラスをインポートするには、以下を実行します。

1. Satellite web UI では、コンテキストメニューから **組織** および **ロケーション** を選択します。

2. 設定 → **Puppet** クラス をクリックします。
3. **ホスト名からインポート** をクリックして、Puppet マスターからパラメーター化されたクラスをインポートします。
4. **Puppet** クラス ページが新規クラスの一覧と共に表示されます。

9.2.4. スマートクラスパラメーターの設定

以下の手順は、クラス内にパラメーターを設定します。パラメーターが含まれるクラスは **パラメーター化されたクラス** と呼ばれています。

手順9.5 スマートクラスパラメーターを設定するには、以下を実行します。

1. 設定 → **Puppet** クラス をクリックします。
2. **パラメーター** 列に示されるパラメーターを含むクラスを一覧から選択します。
3. **スマートクラスパラメーター** タブをクリックします。これにより新しい画面が表示されます。左側のセクションには、クラスがサポートする使用可能なパラメーターの一覧が含まれます。右側のセクションには、選択したパラメーターの設定オプションが含まれます。
4. 左側の一覧からパラメーターを選択します。
5. **詳細** テキストボックスを編集して、プレーンテキストのメモを追加します。
6. **上書き** を選択してこの変数に対する Satellite のコントロールを許可します。チェックボックスが選択されていない場合、Satellite は新規の変数を Puppet に渡しません。
7. 渡すデータの **キータイプ** を選択します。最も一般的なタイプは文字列ですが、他のデータタイプもサポートされています。
8. ホストが一致しない場合に Puppet Master に送信されるパラメーターの **Default Value** を入力します。
9. 上書きの一致がない場合、値を Puppet Master に送信 **しない** ようにするには、オプションで **Puppet デフォルトの使用** を選択します。
10. フィールドに作業中に表示したくないデータが含まれる場合、オプションで **非表示の値** を選択します。
11. **Optional Input Validator** セクションを使用してパラメーターに許可される値を制限します。**バリデータータイプ** (コンマ区切り値の **list** か、または正規表現の **regexp** のいずれか) を選択し、許可される値または正規表現コードを **バリデータールール** フィールドに入力します。
12. **上書き** オプションが選択されていると **属性の優先度付け** セクションが表示されます。これは条件付き引数に基づいて特定ホストの値を無効にするオプションを提供します。この属性タイプおよびその値は **Matcher** として知られています。
 - a. エントリーを一覧に配置して、ホスト属性またはファクトが **Matcher** に対して評価される優先順位の **順序** を設定します。デフォルト一覧に追加することができます。**Matcher** 間で論理 AND 条件を作成するには、それらをコンマ区切りの一覧として一行に配置します。
 - b. **Matcher の追加** をクリックして条件付き引数を追加します。一致する属性は **順序** 一覧のエントリーに対応するはずで、**Matcher** が設定されていない場合、デフォルト値のみが上書き機能として使用されます。

たとえば、Puppet マスターに指定する必要のあるパラメーターの値が **server1.example.com** の完全修飾ドメイン名を持つホストの **test** の場合、**Matcher** を

`fqdn=server1.example.com` として、**値** を **test** に設定します。

マッチングの優先順位は以下のとおりです。

- a. マッチャーがホスト属性の場合はそれを使用します。
- b. 該当する名前を持つ属性がない場合は、一致するホストパラメーター (パラメーターの階層に基づいて継承される) を探します。
- c. それでも一致しなければホストのファクトをチェックします。

Facter にある属性で、ホストの属性と混同しない属性を使用することが推奨されています。

動的なデータは、*Embedded Ruby* (ERB) テンプレート構文の **値** フィールドでパラメーターおよび Puppet ファクトを使用すると利用できます。たとえば、Puppet ファクトを値の一部として使用するには、以下を実行します。

```
<%= @host.facts['network_eth0'] %>
```

利用可能な Puppet ファクトを一覧表示するには、**モニター → ファクト** に移動します。

13. **送信** をクリックして変更を保存します。

Puppet モジュールの使用についての詳細は、[Adding Puppet Modules to Red Hat Satellite 6](#)を参照してください。ERB 構文についての詳細は、[付録A テンプレート作成の参照](#)を参照してください。

9.3. プロビジョニング設定の構成

このセクションでは、プロビジョニング環境の要素を作成し、それらを設定する方法を示します。

9.3.1. ドメイン

Satellite には、Red Hat Satellite Capsule Server の DNS を使ってドメイン名を割り当てる機能があります。この機能により、ユーザーは特定ドメイン内のホストをグループ化し、これらに名前を付けることができます。

手順9.6 ドメインを作成するには、以下を実行します。

1. **インフラストラクチャー → ドメイン** をクリックします。
2. **新規ドメイン** をクリックします。ドメイン タブで以下の設定を指定します。
 - a. ドメインの **名前** を指定します。これは必須の DNS ドメイン名です。
 - b. ドメインの **詳細** を入力します。
 - c. DNS で有効にされた Capsule Server を選択します。
3. **パラメーター** タブで、ドメインパラメーターを指定します。
4. **Locations (ロケーション)** タブで、ドメインのロケーションを選択します。
5. **組織** タブで、ドメインの組織を選択します。

**重要**

ロケーションと組織はデバッグ作業に役立つため、これらが設定されていることを確認します。

6. **送信** をクリックします。

9.3.2. サブネット

Satellite には、システムの複数のグループ用にネットワークを作成する機能があります。サブネットは標準 IP アドレス設定を使用してネットワークを定義し、Red Hat Satellite Capsule Server の DHCP 機能を使用して IP アドレスをサブネット内のシステムに割り当てます。

9.3.2.1. サブネットの作成

以下の手順は、サブネットを作成する方法を示しています。

手順9.7 サブネットを作成するには、以下を実行します。

1. **インフラストラクチャー** → **サブネット** をクリックします。
2. **新規サブネット** をクリックします。**サブネット** タブで、以下の設定を指定します。
 - a. サブネットの **名前**、**ネットワーク (IP アドレス)**、および **マスク** を指定します。これらの設定は必須です。
 - b. オプションで、**ゲートウェイ**、**プライマリー DNS**、**セカンダリー DNS**、および **VLAN ID** を指定します。ゲートウェイアドレスは、IPAM およびブートモードが DHCP (デフォルト) に設定されている場合のみのオプションになることに注意してください。これらのデフォルトモードを変更することにする場合は、ゲートウェイおよび DNS を指定することも必要です。

また、**IPAM モード** (「DHCP」、「内部 DB」、または「なし」) を選択して、**開始アドレス** および **終了アドレス** フィールドで IP 割り当ての範囲を定義することもできます。
 - c. サブネットのデフォルト **ブートモード** (「DHCP」または「静的」) を選択します。
3. **ドメイン** タブで、サブネットの適用可能なドメインを選択します。
4. **Capsules (プロキシ)** タブで、DHCP、TFTP、DNS、および Discovery サービスをホストするために使用される Capsule Server を選択します。
5. **ロケーション** タブで、サブネットのロケーションを選択します。
6. **組織** タブで、サブネットの組織を選択します。

**重要**

ロケーションと組織はデバッグ作業に役立つため、これらが設定されていることを確認します。

7. **送信** をクリックします。

9.3.3. アーキテクチャー

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。ただし、アーキテクチャーはデフォルトのインストールには存在しないため、作成する必要があります。

手順9.8 アーキテクチャーを作成するには、以下を実行します。

1. **ホスト → アーキテクチャー** をクリックしてから **新規アーキテクチャー** をクリックします。
2. アーキテクチャーの **名前** を指定します。
3. このアーキテクチャーが含まれるすべての **オペレーティングシステム** を選択します。該当するオペレーティングシステムがない場合は、**ホスト → オペレーティングシステム** 下でこれらを作成し、割り当てることができます。
4. **送信** をクリックします。

9.3.4. コンピュートリソース

コンピュートリソースは、仮想化およびクラウドのプロバイダーが提供するハードウェアを抽象化したものです。Satellite はコンピュートリソースを使用して、仮想マシンおよびコンテナのプロビジョニングを行います。サポートされるプライベートプロバイダーには、Red Hat Enterprise Virtualization、oVirt、OpenStack、VMware および Docker が含まれます。サポートされるパブリッククラウドプロバイダーには Amazon EC2、Google Compute Engine、および Rackspace が含まれます。

手順9.9 コンピュートリソースを追加するには、以下を実行します。

1. **インフラストラクチャー → コンピュートリソース** に移動します。
2. **新規のコンピュートリソース** をクリックします。コンピュートリソース タブで、以下の設定を指定します。
 - a. コンピュートリソースの **名前** および **プロバイダー** タイプを指定します。オプションとして、**説明** を挿入します。
 - b. 選択したプロバイダーのタイプによって、次のいくつかのフィールドで認証とデータセンターの詳細情報が求められます。それぞれのプロバイダータイプについてさらに詳しくは、以下の表を参照してください。

表9.2 プロバイダーの設定

タイプ	説明
RHEV	Red Hat Enterprise Virtualization 環境に適しています。コンピュートリソースを抽象化するためにシステムで Manager API の URL 、有効な Username および Password 、および Datacenter が必要になります。 Load Datacenters をクリックしてドロップダウンメニューを設定します。オプションで Quota ID を指定し、 X509 Certification Authorities フィールドに 1 つ以上の証明書機関を指定します。
Libvirt	Libvirt ベースの環境に適しています。仮想マシンの URL が必要になります。 Display type を選択します。 Test Connection をクリックして仮想マシンが利用可能かどうかをテストします。 Console passwords を選択して、表示接続にランダムに生成されたパスワードを設定します。

タイプ	説明
VMware	VMware ベースの環境に適しています。コンピュートリソースを抽象化するために VCenter/Server のホスト名、有効な VMware Username および Password 、および Datacenter が必要になります。 Load Datacenters をクリックしてドロップダウンメニューを設定します。証明書 Fingerprint を指定し、 Console passwords を選択して表示接続でランダムに生成されたパスワードを設定します。
RHEL OpenStack Platform	OpenStack ベースの環境に適しています。コンピュートリソースを抽象化するために OpenStack サーバーの URL 、有効な OpenStack Username および Password 、および Tenant が必要になります。 Load Tenants をクリックしてドロップダウンメニューを設定します。
Rackspace	Rackspace のパブリッククラウドアカウントに適しています。コンピュートリソースを抽象化するために Rackspace API の URL 、有効な Rackspace Username および API Key 、および Region が必要になります。 Test Connection をクリックして選択したリージョンへの接続が有効であることを確認します。
EC2	Amazon EC2 のパブリッククラウドアカウントに適しています。有効な Amazon EC2 アカウントから利用できる アクセスキー および 秘密鍵 が必要です。リソースを抽象化するためのデータセンターとして機能させるには リージョン が必要です。 リージョンのロード ボタンをクリックして選択ボックスにデータを設定します。
Google	Google Compute Engine のパブリッククラウドアカウントに適しています。 Google Project ID 、有効な Client Email および p12 ファイルへの Certificate path が必要になります。 Zone を指定してコンピュートリソースを抽象化することもできます。 Load zones をクリックしてドロップダウンメニューを設定します。
Docker	コンテナレジストリーに適しています。内部または外部のコンピュートリソースの URL が必要になります。オプションで Username 、 Password 、およびコンタクト先の Email を指定します。 Test Connection をクリックして接続が利用可能であるかどうかをテストします。

3. **ロケーション** タブで、必要なロケーションを **選択された項目** の一覧に追加するためにそれらのロケーションを選択します。
4. **組織** タブで、必要な組織を **選択された項目** の一覧に追加するためにそれらの組織を選択します。



重要

ロケーションと組織はデバッグ作業に役立つため、これらが設定されていることを確認します。

5. **送信** をクリックします。

9.3.5. コンピュートリソースとしての Libvirt の設定

Libvirt ハイパーバイザーが使用されるシステムで、以下のパッケージがインストールされていることを確認します。

```
# yum install qemu-kvm libvirt virt-manager
```

ハイパーバイザーマシンおよび Satellite Server が実行されているベースシステムで、ハイパーバイザーホストの FQDN が正常に解決されていることを確認します。

- ※ Web UI ブラウザーが **virt-manager** とは別のシステムで実行されている場合は、[手順9.11「Libvirt への SSH アクセスを設定するには、以下を実行します。」](#)に進んでください。
- ※ Web UI ブラウザーが **virt-manager** と同じシステムで実行されている場合、以下の手順を実行して Libvirt コンピュートリソースを追加します。

手順9.10 Libvirt コンピュートリソースを追加するには、以下を実行します。

1. インフラストラクチャー → コンピュートリソース に移動します。
2. **新規のコンピュートリソース** をクリックします。コンピュートリソース タブで、以下の設定を指定します。
 - a. **名前** を指定し、**プロバイダー** ドロップダウンメニューから、コンピュートリソースのタイプとして Libvirt を選択します。オプションで **説明** を挿入します。
 - b. **URL** フィールドに、

qemu:///system

 のように文字列を入力します。
 - c. **タイプの表示** ドロップダウンメニューから、**VNC** を選択します。
 - d. このコンピュートリソースが新規の Libvirt ゲストにのみ使用される場合は、オプションで **コンソールのパスワード** チェックボックスを選択します。このオプションは、先に設定された Libvirt ゲストと共に使用することはできません。
 - e. **テスト接続** をクリックします。接続が正常な場合、ボタンが緑色に変わります。
 - f. **送信** をクリックして設定を保存します。
 - g. **仮想マシン** タブを選択します。先に設定した Libvirt ゲストが表示されます。

手順9.11 Libvirt への SSH アクセスを設定するには、以下を実行します。

他に指定がない限り、Red Hat Satellite が実行されているシステムで以下の手順を実行します。

1. **foreman** ユーザーの SSH ファイルパーミッションおよび SELinux コンテキストが正しいことを確認します。

```
# ls -Zd /usr/share/foreman/.ssh
drwx-----. foreman foreman system_u:object_r:ssh_home_t:s0
/usr/share/foreman/.ssh
```

2. **foreman** ユーザーの SSH キーを作成します。

```
# su - foreman -s /bin/bash
-bash-4.2$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/usr/share/foreman/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /usr/share/foreman/.ssh/id_rsa.
Your public key has been saved in /usr/share/foreman/.ssh/id_rsa.pub.
```



```

The key fingerprint is:
07:47:a9:23:d2:fe:2f:07:fb:55:75:46:3e:8e:6e:69
foreman@satellite.example.com
The key's randomart image is:
+--[ RSA 2048 ]-----+
|           ..      .|
|           ..      0|
|      .   ...   .|=|
|     . o oo     ooo|
|     o .S..    ...|
|       . . .   ...|
|       . o   .E   |
|         + . .o   |
|         =0       |
+-----+

```

3. **SSH** 公開キーをリモートハイパーバイザーシステムにコピーします。たとえば、Libvirt ホストが `kvm.example.com` の場合は以下ようになります。

```

-bash-4.2$ ssh-copy-id root@kvm.example.com
The authenticity of host 'kvm.example.com (192.168.1.2)' can't be
established.
ECDSA key fingerprint is
78:79:41:d0:b8:40:d5:4a:6d:7f:22:03:bd:cd:a0:dd.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
are prompted now it is to install the new keys
foreman@192.168.1.2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@kvm.example.com'"
and check to make sure that only the key(s) you wanted were added.

```

4. **SSH** をリモートシステムに接続して、パスワードのプロンプトが表示されないことを確認します。

```
-bash-4.2$ ssh root@kvm.example.com
```

パスワードのプロンプトが **出されることがありません**。公開キーはリモートシステムの `.ssh/authorized_keys` ファイルにあります。ログインしてキーを確認後に終了します。

```
-bash-4.2$ exit
```

5. Web UI で、インフラストラクチャー → コンピュートリソース に移動してから **新規のコンピュートリソース** をクリックします。
6. **名前** フィールドで適切な名前を見つけます。
7. **プロバイダー** ドロップダウンメニューから **Libvirt** を選択します。
8. **URL** フィールドで、以下のような形式の文字列を入力します:

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


ここで、`kvm.example.com` は Libvirt ホストの FQDN です。

9. **タイプの表示** ドロップダウンメニューで **VNC** を選択します。
10. オプションとして、コンピュータリソースが新規の Libvirt ゲストのみに使用される場合に **コンソールのパスワード** チェックボックスを選択します。このオプションは、先に設定した Libvirt ゲストと共に使用することはできません。
11. **テスト接続** をクリックします。接続が正常な場合、ボタンが緑色に変わります。
12. **送信** をクリックして設定を保存します。
13. **仮想マシン** タブを選択します。先に設定した Libvirt ゲストが表示されます。

手順9.12 noVNC コンソールを設定するには、以下を実行します。

前提条件:

- ※ SSH キーは Satellite Server のベースシステムの **foreman** ユーザーに対して設定する必要があります (前述)。
 - ※ 既存の Libvirt ゲストは、表示タイプとして VNC サーバーを使用できるように設定される必要があります。この際、ポートは **Auto** に設定され、VNC パスワードは選択されません。
1. ハイパーバイザーホストシステムで、**VNC** サービスをポート **5900** から **5930** までで許可するようにファイアウォールを許可できるように設定します。

A. Red Hat Enterprise Linux 6 の場合:

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

B. Red Hat Enterprise Linux 7 の場合:

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

2. Web UI に使用されるブラウザで、以下のように Satellite Server 証明書を信頼します。
 - a. Satellite Server の公的なダウンロードページ (例: <https://satellite.example.com/pub/>) に移動し、証明書ファイル **katello-server-ca.crt** をクリックします。
 - b. Web サイトを特定するために証明書を信頼する選択をします。
3. web UI に使用されるブラウザで、*HTTP strict transport security* (HSTS) を無効にします。HSTS については、[RFC 6797](#) で説明されています。Firefox の場合、ブラウザアドレスバーに **About:Config** を入力し、以下のブーリアンを **True**:

```
network.websocket.allowInsecureFromHTTPS
```

に設定します。

4. web UI のブラウザで FQDN を使用していることを確認します。NoVNC は URL のドメイン名が証明書の CN 値に一致しない場合には機能しません。これは FQDN と同じである必要があります。以下のようにコマンドを使用して CN 値をチェックします。

```
# openssl x509 -text -noout -in /etc/pki/katello/certs/katello-
```

```
apache.crt | grep CN
```

```
Issuer: C=US, ST=North Carolina, L=Raleigh, O=Katello, OU=SomeOrgUnit,  
CN=satellite.example.com
```

```
Subject: C=US, ST=North Carolina, O=Katello, OU=SomeOrgUnit,  
CN=satellite.example.com出力省略
```

5. インフラストラクチャー+**C**コンピュートリソース に移動します。Libvirt リソースの名前を選択します。**仮想マシン** タブで、Libvirt ゲストの名前を選択します。マシンの電源がオンになっていることを確認してから、**コンソール** を選択します。コンソールウィンドウは noVNC ハンドシェイクの完了後に表示されます。

9.3.6. ハードウェアモデル

ハードウェアモデルは、Scalable Processor Architecture (SPARC) ベースのシステムに無人インストールを実行する際に役立ちます。

手順9.13 ハードウェアモデルを作成するには、以下を実行します。

1. **ホスト** → **ハードウェアモデル** をクリックします。
2. **新規モデル** をクリックします。
3. ハードウェアモデルの **名前** を指定します。
4. SPARC ビルドの場合、CPU の **ハードウェアモデル** と **ベンダークラス** を挿入します。他のアーキテクチャーの場合、これらのフィールドに値は必要ありません。
5. **情報** テキストボックスにハードウェアモデルの詳細を入力します。
6. **送信** をクリックします。

9.3.7. インストールメディア

Red Hat Satellite は、キックスタートツリーや新規ホストのインストール用のコンテンツとしてインストールメディア (ISO イメージ) を使用します。

手順9.14 インストールメディアを追加するには、以下を実行します。

1. **ホスト** → **インストールメディア** をクリックします。
2. **新規メディア** をクリックします。**メディア** タブで、以下の設定を指定します。
 - a. インストールメディアの **名前** を入力します。この設定は必須です。
 - b. インストールメディアへの **パス** を入力します。オプションとして、URL または有効な NFS サーバーのいずれかを使用できます。この設定は必須です。
 - c. **オペレーティングシステムファミリー** を選択して、インストールメディアのタイプを定義します。
3. **ロケーション** タブで必要なロケーションを選択し、それらを **選択された項目** の一覧に追加します。
4. **組織** タブをクリックしてから、必要な組織を選択し、それらを **選択された項目** の一覧に追加します。



重要

ロケーションと組織はデバッグ作業に役立つため、これらが設定されていることを確認します。

5. **送信** をクリックします。

9.3.8. パーティションテーブル

パーティションテーブルは、システムのプロビジョニング時の新規インストール用のパーティションとファイルシステムのレイアウトを定義します。Satellite ユーザーは、ホストのディスクレイアウトをパーティションの明示的な順序として指定するか、または動的なディスクレイアウトのスクリプトを使用します。

手順9.15 パーティションテーブルを作成するには、以下を実行します。

1. **ホスト** → **パーティションテーブル** をクリックします。
2. **新規パーティションテーブル** をクリックします。
3. パーティションテーブルの **名前** を入力します。
4. パーティションテーブルの **レイアウト** を指定します。**レイアウト** フィールドは、動的なディスクパーティションスクリプトも入力できます。
5. **OS 種類** ドロップダウンリストからオペレーティングシステムを選択します。
6. **送信** をクリックします。

新規パーティションテーブルは、[「オペレーティングシステム」](#) で説明されているようにオペレーションシステムに関連付ける必要があります。

9.3.9. 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、無人インストールを実行するための体系的な手段です。プロビジョニングテンプレートは、bash スクリプト、キックスタートスクリプト、および PXE ベースのインストールなどの複数の方法で実行することができます。

手順9.16 プロビジョニングテンプレートの作成:

1. **ホスト** → **プロビジョニングテンプレート** をクリックします。
2. **新規テンプレート** をクリックします。**テンプレート** タブで、以下の設定を指定します。
 - a. テンプレートの **名前** を指定します。
 - b. **テンプレートエディター** フィールドにテンプレートを挿入します。または、**参照** をクリックしてテンプレートをアップロードします。これにより、**テンプレートエディター** フィールドの内容が選択したファイルの内容に置き換わります。
 - c. オプションとして、**監査コメント** フィールドにコメントを入力します。Satellite ではコメントをテンプレート履歴に追加して変更を追跡します。**履歴** タブの下でテンプレート履歴を表示します。
3. **タイプ** タブで、**スニペット** を選択し、特定のスクリプトやテンプレートタイプを定義せずにテンプレートコードを保存するか、または **タイプ** ドロップダウンメニューからタイプを選択します。

4. **関連付け** タブで、テンプレートに関連付けるホストグループ、環境およびオペレーティングシステムを選択します。**適用可能なオペレーティングシステム** 一覧からオペレーティングシステムを選択します。**組み合わせの追加** をクリックして **ホストグループ** および **環境** を選択し、テンプレートの使用を制限します。関連付けはスニペットタイプでは使用できないことに注意してください。
5. **関連付け** タブで、既存テンプレートの履歴を表示できます。新規テンプレートの作成時に履歴は利用できません。
6. **ロケーション** タブで、テンプレートのロケーションを選択します。
7. **組織** タブで、テンプレートの組織を選択します。



重要

ロケーションと組織はデバッグ作業に役立つため、これらが設定されていることを確認します。

8. **送信** をクリックします。

プロビジョニングテンプレートについての詳細は、『Red Hat Satellite Provisioning Guide』の [Creating Provisioning Templates](#) を参照してください。

9.3.10. プロビジョニング時間を削減するための gPXE の設定

PXE ブートファイルのダウンロード時にプロビジョニング時間を削減するために、gPXE は **HTTP** などの追加のプロトコルの使用可能にしてダウンロード時間を短縮します。gPXE を使用するには、以下を実行します。

- ※ **TFTP** サーバーにするように設定されたシステムで、**/usr/share/syslinux/gpxelinuxk.0** を **/var/lib/tftpboot** にコピーします。
- ※ **/etc/dhcp/dhcpd.conf** の **PXE Handoff** セクションで、**DHCP filename** オプションを **pxelinux.0** から **gpxelinuxk.0** に変更します。
- ※ プロビジョニングテンプレートを以下のように作成してから、それらをデフォルトテンプレートと共にオペレーティングシステムに割り当てます。

手順9.17 gPXE プロビジョニングテンプレートを設定するには、以下を実行します。

1. **ホスト** → **プロビジョニングテンプレート** をクリックします。
2. テンプレートの **Kickstart default PXELinux** を見つけ、**クローン** を選択します。
3. たとえば、**Kickstart default gPXELinux** などの名前を入力します。
4. テンプレートエディターで、**@initrd** を検索し、これを **@host.url_for_boot(:initrd)** に置き換えます。
5. テンプレートエディターで、**@kernel** を検索し、これを **@host.url_for_boot(:kernel)** に置き換えます。
6. **タイプ** タブを選択します。**タイプ** ドロップダウンメニューから **PXELinux** を選択します。
7. **関連付け** タブで、テンプレートに関連付けるホストグループ、環境およびオペレーティングシステムを選択します。**適用可能なオペレーティングシステム** 一覧からオペレーティングシステムを選択します。**組み合わせの追加** をクリックし、**ホストグループ** および **環境** を選択してからテンプレートの使用を制限します。

8. **送信** をクリックします。

9.3.11. オペレーティングシステム

オペレーティングシステムは、インストール方法とメディアの組み合わせを定義するものであり、ファミリーに分類されます。デフォルトでは、Red Hat Satellite は **RedHat** ファミリーを使用します。ファミリーの分類により、Satellite ではホストのプロビジョニング時の特定の動作を変更することができます。

手順9.18 オペレーティングシステムを追加するには、以下を実行します。

1. **ホスト → オペレーティングシステム** をクリックします。
2. **新規オペレーティングシステム** をクリックします。オペレーティングシステム タブで、以下の設定を指定します。
 - a. オペレーティングシステムの **名前** およびその **メジャーバージョン** を入力します。これらの設定は必須です。
 - b. オプションで **マイナーバージョン** を定義し、**OS Family (OS 種別)** を選択してからオペレーティングシステムの **説明** を追加します。
 - c. **root パスワードのハッシュ** (MD5、SHA256、SHA512) を選択します。
 - d. 利用可能なアーキテクチャーの一覧から **アーキテクチャー** を選択します。利用可能なアーキテクチャーがない場合は、[「アーキテクチャー」](#) で説明されているようにそれらを作成し、**ホスト → アーキテクチャー** の下で割り当てます。
3. **パーティションテーブル** タブで、一覧から適用可能なファイルシステムレイアウトを選択します。パーティションテーブルの作成についての詳細は、[「パーティションテーブル」](#) を参照してください。
4. **インストールメディア** タブで、一覧から適用可能なインストールメディアを選択します。インストールメディアの追加についての詳細は、[「インストールメディア」](#) を参照してください。
5. **テンプレート** タブで、既存のオペレーティングシステムの編集時にプロビジョニングテンプレートを割り当てることができます。このオプションは、新規のオペレーティングシステムの作成時には利用できません。プロビジョニングテンプレートの作成についての詳細は、[「プロビジョニングテンプレート」](#) を参照してください。
6. **パラメーター** タブで、オペレーティングシステムのパラメーターを追加できます。
7. **送信** をクリックします。

9.4. ホスト情報の保存および維持

Red Hat Satellite 6 はアプリケーションの組み合わせを使用して管理対象ホストについての情報を収集し、それらのホストがあるべき状態で維持されるようにします。これらのアプリケーションには以下が含まれます。

- ※ **Foreman**: 物理システムと仮想システムのプロビジョニングとライフサイクル管理を可能にします。Foreman は、キックスタートや Puppet モジュールなどの各種の方法を使ってこれらのシステムを自動的に設定します。
- ※ **Puppet**: ホストを設定するためのクライアント/サーバーアーキテクチャーです。Puppet マスター (サーバー) および Puppet エージェント (クライアント) で構成されます。

- ✧ **Facter:** Puppet のシステムインベントリツールです。Facter は、ハードウェアの詳細、ネットワーク設定、OS タイプおよびバージョン、IP アドレス、MAC アドレス、SSH キーなどの基本的な情報 (ファクト) を収集します。これらのファクトは Puppet マニフェストで変数として利用可能になります。

Puppet、Facter およびファクトの使用については、以下で詳述します。

9.4.1. Puppet アーキテクチャー

Puppet は通常、エージェント/マスター (クライアント/サーバーとしても知られる) アーキテクチャーで実行されます。ここで、Puppet サーバーは重要な設定情報を管理し、管理対象ホスト (クライアント) は独自の設定カタログのみを要求します。Puppet は以下の 2 つの手順でホストを設定します。

- ✧ カタログのコンパイル
- ✧ カタログの該当ホストへの適用

エージェント/マスターの設定では、Puppet クライアントは Facter が収集するファクトとその他の情報を Puppet マスターに送信します。Puppet マスターはこれらのファクトに基づいてカタログをコンパイルし、このカタログをクライアントに送信します。クライアントはすべての変更についてのレポート、または **--noop** パラメーターが使用されていた場合に加えられていた可能性のある変更についてのレポートを Puppet マスターに送信します。次に Puppet マスターはその結果を Foreman に送信します。このカタログは 1 つの特定のホストのあるべき状態について記述します。また、リソース間の依存関係を含むホストで管理するリソースを一覧表示します。エージェントはカタログをホストに適用します。

マスターとエージェント間のこの通信は、デフォルトで 30 分ごとに行われます。**runinterval** パラメーターを使用して、**/etc/puppet/puppet.conf** ファイルに異なる値を指定することができます。また、**puppet agent apply** を実行して通信を手動で開始することもできます。

9.4.2. Facter およびファクトの使用

Facter は Puppet のシステムインベントリツールで、これには多数のビルトインファクトが組み込まれています。Facter はローカルホストのコマンドラインで実行し、ファクト名と値を表示できます。Facter はカスタムファクトを使って拡張し、それらを使用してホストのサイト固有の詳細情報を Puppet マニフェストに公開できます。また、Facter が提供するファクトを使用して Puppet マニフェストの条件式を作成することができます。

Puppet はリソースに基づいてシステムの状態を判別します。たとえば、Puppet に対して **httpd** サービスは常に実行状態にする必要があると指示することができます。Puppet はそれを処理する方法を認識します。異なるオペレーティングシステムを管理している場合は、**osfamily** ファクトを使用して条件式を作成し、Puppet に対して確認するサービスまたはインストールするパッケージについて指示します。**operatingsystemmajrelease** および **versioncmp** パラメーターを使用して、同じオペレーティングシステムの異なるバージョンに基づいて条件式を作成できます。条件式の使用例については、[例9.1「条件式とファクトの使用」](#)を参照してください。

例9.1 条件式とファクトの使用

```
if $::osfamily == 'RedHat' {
  if $::operatingsystemmajrelease == '6' {
    $ntp_service_name = 'ntpd'
  }

  elsif versioncmp($::operatingsystemmajrelease, '7') >= 0 {
    $ntp_service_name = 'chrony'
  }
}
```


注記

この例では、式 `"versioncmp($::operatingsystemmajrelease, '7') >= 0"` を使用してバージョン 7 以降の Red Hat Enterprise Linux のテストを実行します。式 `"$::operatingsystemmajrelease >= '7'"` を使用してこのテストを実行しないでください。この詳細および他の Puppet 機能については、<https://docs.puppetlabs.com/references/latest/function.html#versioncmp> を参照してください。

Puppet は動作がファクトに非常に似ている他の特殊な変数も設定します。詳細は、[Special Variables Added by Puppet](#) [3] および [Core Facts](#) [4] を参照してください。

9.4.2.1. 特定ホストのファクトの表示

Puppet は、Facter の組み込まれたコアファクトと、Puppet モジュールにあるカスタムまたは外部のファクトにアクセスできます。利用可能なファクトは、コマンドライン (**facter -p**) で、また Web UI (モニター → ファクト) でも表示できます。ファクトの一覧を閲覧したり、**検索** ボックスを使用して特定ファクトを検索したりできます。たとえば、「**facts.**」と入力して利用可能なファクトの一覧を表示します。

注記

利用可能なファクトの一覧の長さは非常に長くなります。UI は一度に 20 ファクトのみを表示します。ファクトの一覧は、詳細を入力すると徐々にフィルターされます。たとえば、「facts.e」と入力すると、「e」文字で始まるすべてのファクトが表示されます。

手順9.19 特定ホストのファクトを表示するには、以下を実行します。

1. メインメニューで、**ホスト → すべてのホスト** をクリックしてから検査する必要のあるホストの名前をクリックします。
2. **詳細** ペインで、**ファクト** をクリックし、ホストについての既知のファクトすべてを表示します。

注記

- ※ このページに一覧表示されているファクトについて、**Chart (チャート)** をクリックし、このファクト名のすべての管理対象ホストにおけるディストリビューションチャートを表示します。
- ※ 検索をブックマークすると、今後の検索を容易にできます。検索を絞る場合は **検索** ボタンの横のドロップダウン矢印をクリックして **この検索をブックマーク** をクリックします。ブックマークされた検索が **検索** ドロップダウンリストに表示され、またメインメニューの **管理 → ブックマーク** にも表示されます。

9.4.2.2. ファクトに基づくホストの検索

Facter 情報を使用して特定のホストを検索できます。これは、**facts.architecture = x86_64** などの特定のファクト条件に一致するすべてのホストを検索できることを意味します。

手順9.20 ファクトに基づいてホストを検索するには、以下を実行します。

1. メインメニューで、モニター → ファクト をクリックし、ファクト値 ページを表示します。
2. 検索 フィールドで、フィルターをかけるファクトの名前を入力します。特定の名前、名前/値のペアなどで検索を実行できます。
3. 検索 をクリックして、一致するホストの一覧を検索します。

9.4.2.3. カスタムファクトのレポート

管理対象ホストからのカスタム情報の取得は Red Hat Satellite 6 で完全にサポートされています。このセクションでは、Puppet Forge から取得される Puppet モジュールの使用について説明しますが、この原則は Puppet モジュールの他のソースにも同様に適用されます。

標準の Facter インターフェースでレポートされるファクト数は拡張することができます。たとえば、ファクトをモジュールの変数として使用するために収集できます。インストールされたパッケージを記述するファクトが利用可能な場合、このデータを検索し、情報に基づく設定管理の決定を行うことができます。

ホストにインストールされたパッケージについてのレポートを取得するプロセスは以下のようになります。

- ※ マニフェスト **pkginventory** は Puppet Forge から取得され、ベースシステムに保存されます。
- ※ Puppet モジュールはコンテンツビューに追加され、これはシステムにプロモートされてからそのシステムにデプロイされます。
- ※ その後、システムのファクトのクエリーがパッケージ名を使用して実行されます。この例では、ホスト *hostname* および Satellite ユーザーが認証情報の *username* および *password* と共に使用されます。以下の API クエリーは検索文字列「bash」に一致したファクトを返します。

```
curl -u username:password -X GET
http://localhost/api/hosts/:hostname/facts?search=bash
{"hostname":{"pkg_bash":"4.2.45-5.el7_0.4"}}
```

この検索はパッケージのバージョンを返します。これは外部データベースにデータを設定するために使用される可能性があります。

pkginventory Puppet モジュールの追加

pkginventory Puppet モジュールを Red Hat Satellite Server アプリケーションに追加するには、モジュールを <https://forge.puppetlabs.com/ody/pkginventory> から Satellite Server アプリケーションがインストールされているベースシステムにダウンロードし、以下の手順に従います。

Puppet モジュールは、通常 Puppet モジュールというカスタムリポジトリに保存されます。以下の手順では、カスタムリポジトリがこの名前で作成されていることを想定します。Puppet モジュールのカスタムリポジトリをまだ作成していない場合は、『Red Hat Satellite クイックスタートガイド』の [Creating Custom Products and Enabling Repositories \(カスタム製品の作成およびリポジトリの有効化\)](#) を参照してください。

手順9.21 Puppet モジュールをリポジトリにアップロードするには、以下を実行します。

1. Puppet モジュールをベースシステムにダウンロードします。ダウンロードされているモジュールには、**.tar.gz** 拡張子が使用されます。
2. コンテンツ → 製品 をクリックしてから、Puppet モジュールリポジトリに関連付けられた名前フィールドの製品名をクリックします (例: **Custom Products (カスタム製品)**)。

3. **リポジトリ** タブで、変更する Puppet モジュールリポジトリを選択します (例:**Puppet Modules**)。
4. **Puppet モジュールのアップロード** セクションで、**参照** をクリックし、ダウンロードしたモジュールに移動します。
5. **アップロード** をクリックします。

Puppet モジュールをクライアントやコンテンツホストに配信するには、モジュールはコンテンツビューに適用し、公開される必要があります。以下の手順に従ってモジュールをコンテンツビューに追加します。

手順9.22 モジュールをコンテンツビューに追加するには、以下を実行します。

1. **コンテンツ** → **コンテンツビュー** をクリックして、**名前** メニューからコンテンツビューを選択します。
2. **Puppet モジュール** タブで、**新規モジュールの追加** をクリックします。インストールされたモジュールの一覧が表示されます。
3. **アクション** 列で、**バージョンの選択** をクリックして追加するモジュールを選択します。利用可能なバージョンの一覧が表示されます。
4. 追加するモジュールのバージョンの横にある **バージョンの選択** をクリックします。
5. **新規バージョンの公開** をクリックして新規コンテンツビューを作成します。
6. オプションで説明を追加し、**保存** をクリックします。

[3] https://docs.puppetlabs.com/puppet/3.7/reference/lang_facts_and_builtin_vars.html#special-variables-added-by-puppet

[4] https://docs.puppetlabs.com/facter/latest/core_facts.html

第10章 ホストの設定

Red Hat Satellite では、ホストは Red Hat Subscription Manager がインストールされたクライアントシステムです。Red Hat Subscription Manager は更新を Red Hat Satellite に送信し、Red Hat Satellite は更新をこれらのクライアントシステムに送信します。

ホストを管理するには、まず登録を行う必要があります。ホストが登録されると、そのホストを **ホスト** タブで表示させたり編集したりすることができるようになります。これにより、ユーザーはサブスクリプションの追加と管理、ソフトウェアパッケージの追加と削除、および更新の適用などを実行できるようになります。

10.1. ホストの参照




Satellite Server web UI では、Satellite Server が認識するすべてのホストを閲覧できます。画面上部の **ホスト** タブに移動し、以下の項目を含むドロップダウンメニューを開きます。

- ✦ **すべてのホスト:** Satellite Server が認識するすべてのホストの一覧です。
- ✦ **検出されたホスト:** Discovery プラグインによってプロビジョニングネットワークで検出されたベアメタルホストの一覧です。
- ✦ **コンテンツホスト:** コンテンツおよびサブスクリプションに関連付けられたタスクを管理するホストの一覧です。
- ✦ **ホストコレクション:** エラータのインストールなどの一括操作に使用されるユーザー定義のホストコレクションの一覧です。

10.2. ホストの状態タイプ

Satellite Server が認識するそれぞれのホストには、ホストに対する最新のアクションまたはそのホストに適用される今後の変更に基づいて状態のタイプが割り当てられます。**ホスト** → **すべてのホスト** に移動して、各ホストの状態を表示します。以下の表は、ホストに割り当てることのできる状態タイプを示しています。

表10.1 ホストの状態タイプ

アイコン	状態	説明
	エラー	エラーがホストで検出されています。マウスをエラーアイコン上に置くと、エラーの実際の理由を示すヒントが表示されます。ホストをクリックすると、問題のさらに詳細なレポートを表示できます。
	警告	ホストが設定されていますが、最終レポート期間のホストについてのレポートは収集されていません。
	OK	ホストについての保留中のアクションがなく、最終レポート期間には保留中の変更やエラーがありません。

10.3. ホストの概要

ホストの概要ページには、指定されたホストおよびホストとインストーラー間の接続についての情報が表示されます。ホストの概要ページを表示するには、**ホスト** → **すべてのホスト** を選択してから、ホストの名前をクリックします。

詳細

詳細バーには、ホストについての詳細情報へのショートカットを提供するボタンの列と、重要な詳細情報とイベントの概要を表示するタブが含まれます。

- ✦ **監査:** 現在のホストの監査エントリーが含まれるページです。
- ✦ **ファクト:** 現在のホストのファクトの一覧が含まれるページです。このボタンは、インストーラーがホストのファクトを収集した後にのみ選択できます。
- ✦ **レポート:** 現在のホストのレポートの一覧を含むページです。このボタンは、インストーラーがホストのレポートを収集した後にのみ選択できます。
- ✦ **YAML:** ホストの IP アドレス、MAC アドレス、名前、ホストに適用されているパラメーターの値などのホストについての YAML 形式の詳細が記載されたページです。
- ✦ **プロパティ:** ホストの IP アドレス、MAC アドレス、およびホストに適用されているオペレーティングシステムのエントリーなどのホストについての一般的な詳細情報の一覧です。
- ✦ **メトリック:** ホストについてレポートされているすべてのイベントの概要を示す表です。
- ✦ **テンプレート:** ホストが現在アクセスできるすべてのプロビジョニングテンプレートの一覧です。この一覧に含まれるプロビジョニングテンプレートはホストに適用されるオペレーティングシステムのエントリーに基づいて自動的に設定されます。
- ✦ **NIC:** ホストについて設定された NIC の詳細情報を示す表です。

ホストのアクション

以下のボタンのいずれかをクリックしてホストに対する共通のアクションを実行します。

- ✦ **ジョブを実行:** ホストでのジョブの実行を許可します。ジョブの実行についての詳細は、[12章Satelliteホストでのジョブの実行](#) を参照してください。
- ✦ **ブートディスク:** ホストのブートディスクを選択できるメニューです。ホストのブート ISO の作成についての詳細は、『Red Hat Satellite Provisioning Guide』の [Creating New Hosts with PXE-less Provisioning](#) を参照してください。
- ✦ **編集:** ホストの設定を行えるホストの詳細ページを開きます。インストーラーがすべての設定を自動的に行うため、通常は手動の設定が不要となることに注意してください。
- ✦ **ビルド:** 次回のホストの起動時にプロビジョニングされるホストにフラグを設定します。インストーラーがプロビジョニングプロセスのすべての側面を管理するため、通常はホストを手動でプロビジョニングする必要がないことに注意してください。
- ✦ **削除:** ホストをユーザーインターフェースから削除します。

ホストのグラフ

ホストの概要ページには、ホストに実行された最新の Puppet 実行の状態を表示する 2 つのグラフが含まれます。

- ✦ **ランタイム:** **設定の取得** と **ランタイム** という 2 つのデータポイントを追跡します。**設定の取得** データポイントは、所定の Puppet 実行時にホストの情報を収集するためにかかる時間を表し、**ランタイム** データポイントは、Puppet 実行にかかる時間を表します。これらのデータはどちらも秒単位で測定されます。
- ✦ **リソース:** Puppet 実行時にホストに実行されるアクションの数を追跡します。このグラフに表示されるカテゴリーは **レポート** ページに表示されるカテゴリーと同一であり、各カテゴリーのアクション数を使

用して測定されます。

10.4. ホストの作成

以下の手順では、Red Hat Satellite でホストを作成する方法を説明します。

手順**10.1** ホストを作成するには、以下を実行します。

1. ホスト → **新規ホスト** をクリックします。
2. ホスト タブで、必要な詳細を入力します。
3. **Puppet** クラス タブで、組み込む必要のある Puppet クラスを選択します。
4. インターフェース タブで、以下のアクションを実行します。
 - a. **ドメイン** および **レルム** の詳細を入力します。ホストのプロビジョニングを可能にするためにドメインを指定する必要があります。これにより、**サブネット** の一覧が適切なサブネットのセレクションで自動的に更新されます。
 - b. **主インターフェース** の詳細を入力します。選択したサブネットに DHCP で有効にされた Capsule Server がある場合、IP アドレスが自動的に提案されます。**新規の提案** をクリックして異なるアドレスを生成します。
 - c. オプションとして、**インターフェースの追加** をクリックし、追加のネットワークインターフェースを組み込みます。詳細は、[「追加のネットワークインターフェースの設定」](#) を参照してください。
5. **オペレーティングシステム** タブに必要な詳細を入力します。ドロップダウンリストからパーティションテーブルを選択するか、または **ディスク (Custom partition table)** フィールドにカスタムパーティションテーブルを入力することができます。これら両方を指定することはできません。
6. **パラメーター** タブで、**パラメーターの追加** をクリックし、必要なパラメーターを追加します。これには、すべての Puppet クラスパラメーターや、ホストに関連付けられているホストパラメーターが含まれます。
7. **追加情報** タブで、ホストについての追加情報を入力します。
8. **送信** をクリックしてプロビジョニングリクエストを完了します。

10.5. ホスト登録の設定

Red Hat Enterprise Linux ホストは、デフォルトでカスタマーポータル Subscription Management に登録されます。それぞれのホストの設定を更新して、適切な Satellite Server または Capsule Server から更新を受信できるようにする必要があります。

前提条件

- ✧ ホストは以下の Red Hat Enterprise Linux バージョンを使用している必要があります。
 - 5.8 以上 (s390x の場合は 5.7 以上)
 - 6.4 以上
 - 7.0 以上

- ✳ Red Hat Enterprise Linux のすべてのアーキテクチャがサポートされます (i386、x86_64、s390x、ppc_64)。
- ✳ Satellite Server、任意の Capsule Server、およびすべてのホストが同じ NTP サーバーで同期されていることを確認します。
- ✳ 時刻同期ツールが Satellite Server、任意の Capsule Server、およびホストで実行されていることを確認します。

- Red Hat Enterprise Linux 6 の場合は、以下を実行してください。

```
# chkconfig ntpd on; service ntpd start
```

- Red Hat Enterprise Linux 7 の場合は、以下を実行してください。

```
# systemctl start chronyd; systemctl enable chronyd
```

- ✳ デーモン **rhsmcertd** がホストで実行されていることを確認します。

- Red Hat Enterprise Linux 6 の場合は、以下を実行してください。

```
# service rhsmcertd start
```

- Red Hat Enterprise Linux 7 の場合は、以下を実行してください。

```
# systemctl start rhsmcertd
```

以下の手順では、Red Hat Satellite に登録されるようにホストを自動的に設定する方法を示します。

手順10.2 ホストを自動的に設定するには、以下を実行します。

1. Satellite Server または Capsule Server の完全修飾ドメイン名 (FQDN) をメモしておきます (例: **server.example.com**)。
2. ホストでターミナルを開き、root でログインします。
3. 事前作成のブートストラップ RPM を、ホストが登録される Satellite Server または Capsule Server からインストールします。ブートストラップ RPM はホストのコンテンツソースのロケーションを更新し、ホストが Red Hat Satellite に指定されたコンテンツソースからコンテンツをダウンロードできるようにします。

```
# rpm -Uvh http://server.example.com/pub/katello-ca-consumer-latest.noarch.rpm
```



重要

実行中の Docker デーモンが再起動します。



注記

katello-ca-consumer-hostname-1.0-1.noarch.rpm は、サーバーのホスト名を含む利用可能な追加の *katello-ca-consumer RPM* です。一方 *katello-ca-consumer-latest.noarch.rpm* rpm は常に最新バージョンを反映します。どちらも同じ目的を果たします。

10.6. 登録

10.6.1. ホストの登録

以下の手順は、ホストを Satellite Server または Capsule Server に登録する方法を示しています。Satellite Server がプロビジョニングするホストは、**ホスト → すべてのホスト** からアクセスできる **ホスト** ページに表示されます。キックスタート後のフェーズまたはターミナルで Red Hat Subscription Manager 経由で Satellite Server に登録されるホストは、**ホスト → コンテンツホスト** からアクセスできる **コンテンツホスト** に表示されます。

コンテンツホスト ページには、コンテンツホストの検索可能な一覧が含まれます。**検索** フィールドを使用してコンテンツホストを検索する場合、アスタリスク (*) を使用して部分的な文字列検索を実行します。

たとえば、**dev-node.example.com** という名前のコンテンツホストを検索する場合、検索文字列 **dev-node** を入力しても結果は返されないため、代わりに **dev-node*** を使用します。または、***node*** を使用しても、コンテンツビュー **dev-node.example.com** が検索されます。

前提条件

- ※ [「ホスト登録の設定」](#) のすべての手順を実行します。
- ※ ホスト用にアクティベーションキーがあることを確認します。またはアクティベーションキーを作成します。アクティベーションキーの作成方法については、[7章 アクティベーションキーの設定](#) を参照するか、またはコマンド **man subscription-manager** を使用して各種のオプションを表示します。
- ※ インストールされている **subscription-manager** ユーティリティが 1.10 以上であることを確認します。パッケージは標準の Red Hat Enterprise Linux リポジトリで利用できます。

手順10.3 ホストを登録するには、以下を実行します。

1. ホストでターミナルを開き、root でログインします。
2. Red Hat Subscription Manager (RHSM) に関連するすべての古いホストデータをクリアします。

```
# subscription-manager clean
```

3. RHSM を使用してホストを登録します。

```
# subscription-manager register --org your_org_name --activationkey
your_activation_key
```

例10.1 登録後のコマンド出力:


```
# subscription-manager register --org MyOrg --activationkey TestKey-1
The system has been registered with id: 62edc0f8-855b-4184-b1b8-72a9dc793b96
```

ホストをコンテンツビューに登録するには、**--environment** オプションを使用します。たとえば、ホストを「開発」ライフサイクル環境のコンテンツビュー「MyView」に登録するには、以下を実行します。

```
# subscription-manager register --org your_org_name --environment
Development/MyView --activationkey your_activation_key
```



注記

Red Hat Enterprise Linux 6.3 ホストの場合、リリースバージョンはデフォルトで Red Hat Enterprise Linux 6 Server になり、これは 6.3 リポジトリを指している必要があります。

手順10.4 Red Hat Enterprise Linux 6.3 がリポジトリを指すように設定するには、以下を実行します。

1. Red Hat Satellite で、ホスト → コンテンツホスト を選択します。
2. 変更する必要があるホストの名前をクリックします。
3. コンテンツホストのコンテンツ セクションで、リリースバージョン の右側にある編集アイコンをクリックします。
4. リリースバージョン ドロップダウンメニューから「6.3」を選択します。
5. 保存 をクリックします。

10.6.2. Katello エージェントのインストール

以下の手順は、Katello エージェントを Satellite 6 に登録されているホストにインストールする方法を示しています。*katello-agent* パッケージは、**goferd** サービスを提供する *gofer* パッケージによって異なります。Red Hat Satellite Server または Capsule Server がコンテンツホストに適用できるエラータについての情報を提供できるように、このサービスを有効にしておく必要があります。

前提条件

Satellite バージョン 6.1 以降では、**Satellite Tools** リポジトリを有効にする必要があります。**Red Hat Common** リポジトリは使用されなくなっており、Satellite バージョン 6.1 以上との互換性はありません。

Satellite Tools リポジトリは必要なパッケージを提供するため、これを有効にし、Red Hat Satellite Server に同期させてホストで利用できるようにする必要があります。

手順10.5 Satellite Tools リポジトリが有効にされていることを確認するには、以下を実行します。

1. Satellite web UI を開き、コンテンツ → **Red Hat** リポジトリ に移動してから **RPM** タブをクリックします。
2. **Red Hat Enterprise Linux Server** の項目を見つけ、これを拡張します。
3. **Red Hat Satellite Tools 6.2 (for RHEL バージョン名 Server) (RPMs)** の項目を見つけ、これを拡張します。

Red Hat Satellite Tools 6.2 の項目が表示されていない場合、それらがカスタマーポータル

から取得されるサブスクリプションマニフェストに含まれていないためである可能性があります。これを修正するには、カスタマーポータルにログインしてからこれらのリポジトリを追加し、サブスクリプションマニフェストをダウンロードして、これを Satellite にインポートします。

4. リポジトリの名前の横にある **ENABLED? (有効化)** チェックボックスが選択されていることを確認します。選択されていない場合はこれを選択します。

ホスト上で実行されているすべてのサポートされているメジャーバージョンの Red Hat Enterprise Linux に対して **Satellite Tools** リポジトリを有効にします。

手順10.6 Katello エージェントをインストールするには、以下を実行します。

1. ホストでターミナルコンソールを開き、**root** ユーザーでログインします。
2. 以下のコマンドを使って **satellite-tools** リポジトリが有効にされていることを確認します。

```
# yum repolist enabled | grep -i satellite-tools
```

satellite-tools が有効にされていない場合は、以下のコマンドを使ってこれを有効にします。

```
# subscription-manager repos --enable=rhel-version-server-satellite-tools-6.2-rpms
```

3. 以下のコマンドを使って **katello-agent** RPM パッケージをインストールします。

```
# yum install katello-agent
```

katello-agent が正常にインストールされると、**goferd** サービスが自動的に起動し、有効にされます。

10.6.3. Puppet エージェントのインストールおよび設定

このセクションでは、ホストに Puppet エージェントをインストールし、これを設定する方法について説明します。Puppet エージェントを正常にインストールし、設定した後は、**ホスト → すべてのホスト** に移動し、Red Hat Satellite Server に表示されるすべてのホストの一覧表示が可能になります。

前提条件

Satellite バージョン 6.1 以降では、**Satellite Tools** リポジトリを有効にする必要があります。**Red Hat Common** リポジトリは使用されなくなっており、Satellite バージョン 6.1 以上との互換性はありません。

Satellite Tools リポジトリは必要なパッケージを提供するため、これを有効にし、Red Hat Satellite Server に同期させてホストで利用できるようにする必要があります。

手順10.7 Satellite Tools リポジトリが有効にされていることを確認するには、以下を実行します。

1. Satellite web UI を開き、コンテンツ → **Red Hat** リポジトリ に移動してから **RPM** タブをクリックします。
2. **Red Hat Enterprise Linux Server** の項目を見つけ、これを拡張します。
3. **Red Hat Satellite Tools 6.2 (for RHEL バージョン名 Server) (RPMs)** の項目を見つけ、これを拡張します。

Red Hat Satellite Tools 6.2 の項目が表示されていない場合、それらがカスタマーポータルから取得されるサブスクリプションマニフェストに含まれていないためである可能性があります。これを修正するには、カスタマーポータルにログインしてからこれらのリポジトリを追加し、サブスクリプションマニフェストをダウンロードして、これを Satellite にインポートします。

4. リポジトリの名前の横にある **ENABLED? (有効化)** チェックボックスが選択されていることを確認します。選択されていない場合はこれを選択します。

手順10.8 Puppet エージェントをインストールし、有効にするには、以下を実行します。

1. ホストでターミナルコンソールを開き、**root** ユーザーでログインします。
2. 以下のコマンドを使って **satellite-tools** リポジトリが有効にされていることを確認します。

```
# yum repolist enabled | grep -i satellite-tools
```

satellite-tools が有効にされていない場合は、以下のコマンドを使ってこれを有効にします。

```
# subscription-manager repos --enable=rhel-version-server-satellite-tools-6.2-rpms
```

3. 以下のコマンドを使用して Puppet エージェントの RPM パッケージをインストールします。

```
# yum install puppet
```

4. 起動時に開始する Puppet エージェントを設定します。
 - A. Red Hat Enterprise Linux 6 の場合は、以下を実行します。

```
# chkconfig puppet on
```

- B. Red Hat Enterprise Linux 7 の場合は、以下を実行します。

```
# systemctl enable puppet
```

前提条件

Puppet エージェントを設定する前に、以下の条件を満たしている必要があります。

- ※ ホストを Red Hat Satellite Server に登録しておく必要があります。
- ※ Satellite Tools リポジトリが有効にされている必要があります。
- ※ Puppet パッケージがホストにインストールされている必要があります。

手順10.9 Puppet エージェントを設定するには、以下を実行します。

1. **/etc/puppet/puppet.conf** ファイルでサーバーおよび環境の設定を指定して、Puppet エージェントを設定します。

```
# vi /etc/puppet/puppet.conf
```

```
[main]
# The Puppet log directory.
# The default value is '$vardir/log'.
```

```

logdir = /var/log/puppet

# Where Puppet PID files are kept.
# The default value is '$vardir/run'.
rundir = /var/run/puppet

# Where SSL certificates are kept.
# The default value is '$confdir/ssl'.
ssldir = /var/lib/puppet/ssl

...

[agent]
# The file in which puppetd stores a list of the classes
# associated with the retrieved configuration. Can be loaded in
# the separate ``puppet`` executable using the ``--loadclasses``
# option.
# The default value is '$confdir/classes.txt'.
classfile = $vardir/classes.txt
pluginsync = true
report = true
ignoreschedules = true
daemon = false
ca_server = satellite.example.com
server = satellite.example.com
environment = KT_Example_Org_Library_RHEL6Server_3

# Where puppetd caches the local configuration. An
# extension indicating the cache format is added automatically.
# The default value is '$confdir/localconfig'.
localconfig = $vardir/localconfig

...

```



重要

environment パラメーターをホストが属する Puppet 環境の名前に設定します。Puppet 環境は、ホストまたはホストグループに関連付けることのできる Puppet モジュールのコレクションです。

- ✧ ホストの Puppet 環境を検索するには、**ホスト → すべてのホスト** に移動し、ホストの表の **環境** 列を調べます。
- ✧ Puppet 環境をホストに割り当てるには、**ホスト → すべてのホスト** に移動してから、選択したホストの横にある **編集** をクリックします。
- ✧ Satellite Server で有効にされた Puppet 環境を一覧表示するには、**設定 → 環境** に移動します。さらに Satellite Server の **/etc/puppet/environments/** ディレクトリーで、Puppet 環境に関連付けられている Puppet モジュールおよびマニフェストを検索することもできます。

詳細は、[Red Hat Satellite Puppet Guide](#) を参照してください。

2. ホスト上で Puppet エージェントを実行します。

```
# puppet agent -t --server satellite.example.com
```

3. Satellite Server Web UI を使って Puppet クライアントの SSL 証明書に署名します。
 - a. Web UI から Satellite Server にログインします。
 - b. インフラストラクチャー → **Capsules (スマートプロキシ)** を選択します。
 - c. 必要な Capsule の「アクション」ドロップダウンリストから **証明書** を選択します。
 - d. 必要なホストの右側にある **Sign (署名)** をクリックします。
 - e. **puppet agent** コマンドを再度入力します。

```
# puppet agent -t --server satellite.example.com
```



注記

Puppet エージェントがホストに設定される場合、ホストは組織やロケーションに割り当てられないため **任意の組織** が選択されている場合にのみ **すべてのホスト** の下に一覧表示されます。ホストを組織に割り当てるには「[ホストの特定組織への割り当て](#)」を参照し、ホストをロケーションに割り当てるには、「[ホストの特定ロケーションへの割り当て](#)」を参照してください。

10.7. ホストのグループの変更

以下の手順では、ホストのグループを変更する方法を示します。

1. ホスト → **すべてのホスト** に移動します。
2. 変更する必要があるホストのチェックボックスを選択します。
3. 画面右上の **アクションの選択** メニューで、**グループの変更** を選択します。新規オプションのウィンドウが開きます。
4. **ホストグループの選択** メニューで、ホストのグループを選択します。
5. **送信** をクリックします。

10.8. ホストの環境の変更

以下の手順は、ホストの環境を変更する方法を示します。

1. ホスト → **すべてのホスト** に移動します。
2. 変更する必要があるホストのチェックボックスを選択します。
3. 画面右上の **アクションの選択** メニューで、**環境の変更** を選択します。新規オプションのウィンドウが開きます。
4. **環境の選択** メニューから、ホストの環境を選択します。
5. **送信** をクリックします。

10.9. ホストの管理

以下の手順では、ホストの状態を管理対象と非管理対象の間で切り換える方法について示します。ホストが管理対象として設定される場合、Satellite Server からのネットワーク設定、ストレージ、およびリソース割り当てパラメーターの調整が可能です。

Satellite がプロビジョニングするホストは、デフォルトでは管理対象です。Satellite でサポートされていないオペレーティングシステムを使用するシステムの設定管理についてのレポートを取得する必要がある場合は、ホストを非管理対象にすることが推奨されます。

1. **ホスト** → **すべてのホスト** に移動します。
2. ホストを選択します。
3. **編集** をクリックします。
4. **ホストの管理** または **ホストの管理解除** をクリックしてホストの状態を変更します。
5. **送信** をクリックして変更を保存します。

注記

ホストの状態を管理対象外から管理対象に変更する場合、ホストのページで **ビルド** をクリックしてホストを再ビルドする必要があります。

10.10. ホストの特定組織への割り当て

以下の手順は、ホストを特定の組織に割り当てる方法を示します。組織についての一般的な情報やそれらの設定方法については、『サーバー管理ガイド』の [組織、ロケーション、およびライフサイクル環境の設定](#) を参照してください。

1. **ホスト** → **すべてのホスト** に移動します。
2. 変更する必要があるホストのチェックボックスを選択します。
3. 画面右上の **アクションの選択** メニューで **組織の割り当て** を選択します。新規オプションのウィンドウが開きます。
4. **組織の選択** メニューに移動し、ホストの組織を選択します。**Fix Organization on Mismatch (組織の不一致についての修正)** チェックボックスを選択します。

注記

ドメインやサブネットなど、ホストに関連付けられているリソースがある一方で、それらのリソースがホストの割り当て先の組織に割り当てられていない場合、不一致が生じます。オプション **Fix Organization on Mismatch (組織の不一致についての修正)** により、このようなリソースが組織に追加されるため、このオプションは推奨される選択肢になります。一方、オプション **不一致により失敗** を選択すると、常にエラーメッセージが生成されます。たとえば、ホストのある組織から別の組織への再割り当ては、設定に実際の不一致がない場合でも失敗します。

5. **送信** をクリックします。

10.11. ホストの特定ロケーションへの割り当て

以下の手順では、ホストを特定のロケーションに割り当てる方法を示します。ロケーションについての一般的な情報およびそれらを設定する方法については、『サーバー管理ガイド』の [ロケーション](#) を参照してください。

1. **ホスト** → **すべてのホスト** に移動します。
2. 変更する必要があるホストのチェックボックスを選択します。
3. 画面右上の **アクションの選択** メニューで **ロケーションの割り当て** を選択します。新規オプションのウィンドウが開きます。
4. **ロケーションの選択** メニューに移動し、ホストのロケーションを選択します。**Fix Location on Mismatch** (ロケーションの不一致についての修正) チェックボックスを選択します。



注記

ドメインやサブネットなど、ホストに関連付けられているリソースがある一方で、それらのリソースがホストの割り当て先の組織に割り当てられていない場合、不一致が生じます。オプション **Fix Organization on Mismatch** (組織の不一致についての修正) により、このようなリソースが組織に追加されるため、このオプションは推奨される選択肢になります。一方、オプション **不一致により失敗** を選択すると、常にエラーメッセージが生成されます。たとえば、ホストのある組織から別の組織への再割り当ては、設定に実際の不一致がない場合でも失敗します。

5. **送信** をクリックしてロケーションのホストへの割り当てを完了します。

10.12. ブートストラップスクリプトによる新規ホストの **Satellite 6** への追加

6.2 に組み込まれているブートストラップスクリプトを使用して新規クライアントを Satellite 6 に登録することができます。

ブートストラップスクリプトは、コンテンツの登録、製品の証明書、および Puppet 設定を処理します。ブートストラップスクリプトには、Red Hat Enterprise Linux システムが登録されているかどうか (RHN、Satellite 5、SAM、RHSM)、または登録されている場合でもシステムが Satellite 6 にサブスクライブしているかどうかにかかわらず、その Red Hat Enterprise Linux システムを対象とするという利点があります。

ブートストラップスクリプトのパッケージ *katello-client-bootstrap* は、デフォルトで Satellite Server のベースシステムにインストールされ、スクリプト自体は `/var/www/html/pub/` ディレクトリーにインストールされ、クライアントで使用できるようにされます。これは、以下の形式の URL を使用してアクセスできます:

```
satellite6.example.com/pub/bootstrap.py
```

手順10.10 ブートストラップのクライアントへのインストール

以下のようにブートストラップスクリプトをクライアントシステムにインストールします。

1. **root** で、以下のようにスクリプトに適したディレクトリーに切り替えます。たとえば、以下のようになります。

```
# cd /usr/bin/
```


2. **root** で、スクリプトをダウンロードします。

```
# wget http://satellite6.example.com/pub/bootstrap.py
```

これにより、スクリプトが現行ディレクトリーにインストールされます。

3. スクリプトを実行可能にします。

```
# chmod +x bootstrap.py
```

4. スクリプトが実行できることを確認するには、以下のように使用ステートメントを表示します。

```
# ./bootstrap.py -h
```

5. **root** ディレクトリーに戻り、スクリプトが「./」なしで実行できることを確認します。

```
# cd
# bootstrap.py -h
```

手順10.11 Red Hat Enterprise Linux 6 システムの移行

前提条件:

- ※ ブートストラップスクリプトは、前述のようにインストールされています。
- ※ ホスト用のアクティベーションキーがあります。アクティベーションキーをセットアップするには、[Red Hat ホスト設定ガイド](#) を参照してください。

1. ご使用の環境に適した値を使用して、ブートストラップコマンドを以下のように入力します。

-s, --server オプションについては、Capsule の FQDN 名を指定します。この例では、統合 Capsule が使用されており、URL は Satellite の FQDN です。

```
# bootstrap.py --login=admin \
--server satellite6.example.com \
--location=location_label \
--organization=organization_label \
--hostgroup=hostgroup_label \
--activationkey=activation_key
```

スクリプトは、**-l, --login** オプションを使用して入力した Satellite ユーザーに対応するパスワードの入力を求めるプロンプトを出します。

2. スクリプトが実行され、進捗の通知が **stout** に送信されます。証明書の承認を求めるプロンプトを出す出力を確認します。以下は例になります。

```
[NOTIFICATION], [2016-04-26 10:16:00], [Visit the UI and approve this
certificate via Infrastructure->Capsules]
[NOTIFICATION], [2016-04-26 10:16:00], [if auto-signing is disabled]
[RUNNING], [2016-04-26 10:16:00], [/usr/bin/puppet agent --test --noop
--tags no_such_tag --waitforcert 10]
```

クライアントは、管理者が Puppet 証明書を承認するまで無期限に待機します。

- a. Web UI で、インフラストラクチャー → **Capsule (スマートプロキシ)** に移動します。

- b. **-s, --server** で指定された FQDN に対応する Capsule の名前の右側にある **証明書** を選択します。
 - c. アクション 列で **Sign (サイン)** を選択し、クライアントの Puppet 証明書を承認します。
 - d. ゲストに戻り、残りのブートストラップ処理が完了するのを確認します。
3. Web UI で、**ホスト → すべてのホスト** に移動し、クライアントが適切なホストグループに接続されていることを確認します。

10.13. 追加のネットワークインターフェースの設定

Red Hat Satellite は、単一ホストに対して複数のネットワークインターフェースの指定をサポートします。[「ホストの作成」](#) に説明されているように新規ホストを作成する場合や、既存ホストを編集する場合にこれらのインターフェースを設定することができます。

ホストに割り当てることのできるネットワークインターフェースにはいくつかのタイプがあります。新規インターフェースを追加する場合は、以下のいずれかを選択してください。

- ✦ **インターフェース:** 追加の物理または仮想インターフェースを指定できます。2 つのタイプの仮想インターフェースを作成できます。ホストが単一インターフェースで複数の (仮想) ネットワークと通信する必要がある場合は **VLAN** を使用します。これらのネットワークは相互にアクセスすることはできません。もう 1 つのタイプの仮想インターフェースは **alias** です。これは既存インターフェースに割り当てられる追加の IP アドレスです。詳細は、[「仮想インターフェースの追加」](#)、または [「物理インターフェースの追加」](#) を参照してください。
- ✦ **ボンディング:** ボンディングインターフェースを作成します。NIC ボンディングは複数のネットワークインターフェースを、単一デバイスとして表示され、単一 MAC アドレスを持つ単一インターフェースにバインディングする方法です。これにより、2 つ以上のネットワークインターフェースが 1 つのインターフェースとして機能し、帯域幅の拡大と冗長性を同時に提供します。詳細は、[「ボンディングインターフェースの追加」](#) を参照してください。
- ✦ **BMC:** ベースボード管理コントローラー (BMC) により、マシンの物理的な状態をリモートで監視し、管理することができます。BMC の詳細は、『Red Hat Satellite Installation Guide』の [Enabling Power Management on Managed Hosts](#) を参照し、BMC インターフェースの設定についての詳細は、[「ベースボード管理コントローラー \(BMC\) インターフェースの追加」](#) を参照してください。

注記

追加のインターフェースには、デフォルトで **管理対象** フラグが有効にされています。これは、新規インターフェースが選択したサブネットに関連付けられた DNS および DHCP Capsule Server によるプロビジョニング時に自動的に設定されることを意味します。これには、DNS および DHCP Capsule Server が適切に設定されたサブネットが必要です。ホストのプロビジョニングにキックスタートメソッドを使用する場合、管理対象インターフェースの設定ファイルはインストール後のフェーズで、`/etc/sysconfig/network-scripts/ifcfg-$interface_id` に自動的に作成されます。

注記

現在、仮想およびボンディングインターフェースには物理デバイスの MAC アドレスが必要です。そのため、これらのインターフェースの設定はベアメタルホストでのみ機能します。

10.13.1. 物理インターフェースの追加

以下の手順は、物理インターフェースをホストに追加する方法を示しています。

手順10.12 物理インターフェースを追加するには、以下を実行します。

1. ホスト → すべてのホスト に移動して利用可能なホストを表示します。
2. 編集するホストの横にある **編集** をクリックします。
3. インターフェース タブで、**インターフェースの追加** をクリックします。
4. **Type (タイプ)** メニューで **インターフェース オプション** を選択された状態にします。
5. 追加インターフェースの **MAC address (MAC)** を指定します。この設定は必須です。
6. **eth0** または **eth1.1** などのデバイスの **識別子** を指定します。識別子はボンディングインターフェース (割り当て済みデバイス フィールド。詳細は [手順10.14「ボンディングインターフェースを追加するには、以下を実行します。」](#) を参照してください)、VLAN およびエイリアス (割り当て先 フィールド。 [手順10.13「仮想インターフェースを追加するには、以下を実行します。」](#) を参照してください) に使用されます。
7. ホストの IP アドレスに関連付けられた **DNS 名** を指定します。Satellite はこの名前を選択したドメイン (「DNS A」フィールド) に関連付けられた Capsule Server、および選択したサブネット (「DNS PTR」フィールド) に関連付けられた Capsule Server に保存します。そのため、単一ホストに複数の DNS エントリーを持たせることができます。
8. **ドメイン** ドロップダウンメニューからドメインを選択します。ドメインを作成し、管理するには、**インフラストラクチャー → ドメイン** に移動します。
9. **サブネット** ドロップダウンメニューからサブネットを選択します。サブネットを作成し、管理するには、**インフラストラクチャー → サブネット** に移動します。
10. インターフェースの **IP address (IP)** を指定します。DHCP Capsule Server が割り当てられた管理対象インターフェースでは、DHCP リースを作成するためにこの設定が必要です。DHCP で有効にされた管理インターフェースでは IP アドレスの自動補完を行います。
11. インターフェースを管理するかどうかを決定します。**管理** チェックボックスが選択されている場合、インターフェース設定はプロビジョニング時に関連付けられた Capsule Server からプルされ、DNS および DHCP エントリーが作成されます。キックスタートのプロビジョニングを使用している場合、設定ファイルはインターフェース用に自動的に作成されます。
12. **仮想 NIC** チェックボックスを選択して仮想インターフェースを作成します。詳細は、[「仮想インターフェースの追加」](#) を参照してください。
13. **OK** をクリックしてインターフェース設定を保存してから、**送信** をクリックして変更をホストに適用します。

10.13.2. 仮想インターフェースの追加

以下の手順は、ホストの追加の仮想インターフェースを設定する方法を示しています。これは、VLAN またはエイリアスインターフェースのいずれかを使用することができます。

エイリアスインターフェースは既存インターフェースに割り当てる追加の IP アドレスです。以下に注意してください。

- ✧ エイリアスインターフェースは割り当てられているインターフェースから MAC アドレスを自動的に継承します。そのため、MAC アドレスを指定せずにエイリアスを作成することができます。

- ※ インターフェースは、ブートモードが **静的** に設定されたサブネットに指定する必要があります。

手順10.13 仮想インターフェースを追加するには、以下を実行します。

1. **ホスト** → **すべてのホスト** に移動して利用可能なホストを表示します。
2. 編集するホストの横にある **編集** をクリックします。
3. **インターフェース** タブで、**インターフェースの追加** をクリックします。
4. **Type (タイプ)** メニューで **インターフェース オプション** を選択された状態にします。
5. 一般的なインターフェース設定を指定します。適用できる設定オプションは、[「物理インターフェースの追加」](#) で説明されているように物理インターフェースのオプションと同じです。

管理対象の仮想インターフェースの **MAC アドレス** を指定し、プロビジョニング用の設定ファイルが適切に生成されるようにします。ただし、**MAC アドレス** は管理対象外の仮想インターフェースには不要です。

VLAN を作成する場合、**識別子** フィールドに **eth1.10** の形式で ID を指定します。エイリアスを作成する場合、**eth1:10** の形式で ID を使用します。

6. **仮想 NIC** チェックボックスを選択します。仮想インターフェースに固有の追加の設定オプションがその形式に追加されます。
 - ※ **タグ:** ネットワークのハイレベルのセグメント化を可能にするためにインターフェースごとにタグを指定できます。空白のままにすると、管理対象インターフェースは、このサブネットに VLAN ID が指定されている場合は関連付けられたサブネットの VLAN ID を継承します。このフィールドのユーザー定義のエントリはエイリアスインターフェースでは適用されません。
 - ※ **割り当て先:** 仮想インターフェースが属する物理インターフェースの識別子を指定します (例: eth1)。この設定は必須です。
7. **OK** をクリックしてインターフェース設定を保存してから、**送信** をクリックして変更をホストに適用します。

10.13.3. ボンディングインターフェースの追加

以下の手順は、ホストのボンディングインターフェースを設定する方法を示しています。

手順10.14 ボンディングインターフェースを追加するには、以下を実行します。

1. **ホスト** → **すべてのホスト** に移動して利用可能なホストを表示します。
2. 編集するホストの横にある **編集** をクリックします。
3. **インターフェース** タブで、**インターフェースの追加** をクリックします。
4. **Type (タイプ)** メニューから **ボンド** を選択します。また、タイプ固有の設定オプションがその形式に追加されます。
5. 一般的なインターフェース設定を指定します。適用可能な設定オプションは、[「物理インターフェースの追加」](#) に説明されているように物理インターフェースのオプションと同じです。ボンディングインターフェースは、**識別子** フィールドの **bond0** 形式のユーザー ID を使用します。**MAC address (Mac)** フィールドには単一 MAC アドレスを指定するので十分です。
6. ボンディングインターフェースに固有の設定オプションを指定するには、以下を実行します。

- ※ **モード**: フォールトトレランスおよび負荷分散のポリシーを定義する **ボンドモード** を選択します。それぞれのボンドモードの簡単な説明については、[表10.2「Red Hat Satellite で利用可能なボンディングモード」](#) を参照してください。
- ※ **割り当て済みデバイス**: 割り当てられたデバイスの識別子のコンマ区切りの一覧を指定します。物理インターフェースまたは VLAN を指定できます。
- ※ **ボンドオプション**: 設定オプションのコンマ区切りの一覧を指定します (例: **miimon=100**)。ボンディングインターフェースに指定できるいくつかの設定オプションがあります。詳細は、[Red Hat Enterprise Linux 7 Networking Guide](#) を参照してください。

7. **OK** をクリックしてインターフェース設定を保存してから、**送信** をクリックして変更をホストに適用します。

表10.2 Red Hat Satellite で利用可能なボンディングモード

ボンディングモード	説明
balance-rr	送受信は、ボンディングインターフェースで順次行われます。
active-backup	ボンディングインターフェースの中で最初に利用可能になったものから送受信が行われます。アクティブなボンディングインターフェースに障害がある場合に限り別のボンディングインターフェースが使用されます。
balance-xor	送信は選択されたハッシュポリシーに基づいて行われます。このモードでは、特定のピア用に宛先が指定されたトラフィックは常に同じインターフェースで送信されます。
broadcast	すべての送信はすべてのボンディングインターフェースで行われます。
802.a3	同じ設定を共有するアグリゲーショングループを作成します。アクティブなグループのすべてのインターフェースで送受信が行われます。
balance-tlb	送信トラフィックが各ボンディングインターフェースの現在の負荷に応じて配分されます。
balance-alb	受信ロードバランシングは ARP (Address Resolution Protocol) ネゴシエーションにより実現されています。

10.13.4. ベースボード管理コントローラー (BMC) インターフェースの追加

このセクションでは、ベースボード管理コントローラー (BMC) インターフェースを、この機能をサポートするホストに設定する方法について説明します。

前提条件

次に進む前に、以下の前提条件を満たしていることを確認します。

- ※ BMC は Capsule Server で有効にされている。必要な場合は [手順10.15「既存の Capsule Server で BMC パワー管理を有効にするには、以下を実行します。」](#) を参照してください。
- ※ `ipmitool` パッケージがインストールされている。
- ※ ホストの MAC アドレス、IP アドレスおよび BMC インターフェースのその他の詳細、およびこのインターフェースの適切な認証情報を確認している。

注記

BMC インターフェースが管理されている場合は BMC インターフェースの MAC アドレスのみが必要になります。これは DHCP 予約を作成するために必要になります。

手順10.15 既存の Capsule Server で BMC パワー管理を有効にするには、以下を実行します。

1. **satellite-installer** ルーチンを使用し、以下のオプションを指定した以下のコマンドを実行して Capsule Server の BMC パワー管理を設定します。

```
# satellite-installer --foreman-proxy-bmc=true --foreman-proxy-bmc-  
default-provider=ipmitool
```

2. Capsule Server の機能を更新します。
 - a. Satellite Web UI にログインし、インフラストラクチャー → **Capsule (スマートプロキシー)** に移動します。
 - b. 更新する必要がある機能のある Capsule Sever を特定します。右側のドロップダウンリストで、**更新** をクリックします。**機能** 列にある機能の一覧には BMC が含まれているはずで

手順10.16 BMC インターフェースを追加するには、以下を実行します。

1. ホスト → **すべてのホスト** に移動して利用可能なホストを表示します。
2. 編集するホストの横にある **編集** をクリックします。
3. **インターフェース** タブで、**インターフェースの追加** をクリックします。
4. **Type (タイプ)** メニューから **BMC** を選択します。タイプ固有の設定オプションがその形式に追加されます。
5. 一般的なインターフェース設定を指定します。適用できる設定オプションは、[「物理インターフェースの追加」](#) で説明されているように物理インターフェースのオプションと同じです。
6. BMC インターフェースに固有の設定オプションを指定するには、以下を実行します。
 - ✳ **ユーザー名、パスワード:** BMC で必要とされる認証資格情報を指定します。
 - ✳ **プロバイダー:** BMC プロバイダーを指定します。
7. **OK** をクリックしてインターフェース設定を保存してから、**送信** をクリックして変更をホストに適用します。

10.14. ホストの削除

以下の手順では、Red Hat Satellite からホストを削除する方法を説明します。

手順10.17 ホストを削除するには、以下を実行します。

1. ホスト → **すべてのホスト** をクリックするか、または **ホスト → コンテンツホスト** をクリックします。
2. 削除するホストを選択します。
3. **アクションの選択** をクリックして、ドロップダウンメニューから **ホストの削除** を選択します。
4. 確認のためのポップアップボックスが表示されます。**送信** を選択して Red Hat Satellite からホストを永久に削除します。

**警告**

仮想マシンに関連付けられているホストのレコードが削除されている場合、仮想マシンも削除されます。このような状況で仮想マシンが削除されることを防ぐには、ハイパーバーザーから仮想マシンを削除せずに Satellite との関連付けを解除します。

手順**10.18** 仮想マシンをハイパーバイザーから削除せずに **Satellite** との関連付けを解除するには、以下を実行します。

1. Satellite Web UI で、インフラストラクチャー ドロップダウンメニューにマウスを置いて **コンピュートリソース** をクリックします。
2. 仮想マシンの関連付けを解除するコンピュートリソースを選択します。コンピュートリソースは **名前** というラベルが付けられた列に一覧表示されます。
3. 画面の右上にある **Disassociate VMs (VM の関連付けの解除)** ボタンを選択します。

第11章 Satellite でのベアメタルホストの検出

Red Hat Satellite 6.2 にはすでにインストールされている Discovery プラグインが同梱されています。Discovery プラグインはプロビジョニングネットワーク上の不明ホストのベアメタル検出を自動化します。これらの新規ホストは Satellite Server に登録され、クライアントの Puppet エージェントは、シリアル ID、ネットワークインターフェース、メモリーおよびディスク情報などの Facter が収集するシステムのファクトをアップロードします。登録後は、ホストは Satellite Web UI の **検出されたホスト** ページに表示されます。事前に定義された検出ルールを使用して、プロビジョニングを手動 (Web UI、CLI、または API を使用) か、または自動で開始することができます。

Discovery プラグインは、プロビジョニングネットワークと Satellite Server インスタンスの両方に直接アクセスできる Satellite Capsule Server 経由で通信します。Satellite Server からホストを直接検出することができますが、Red Hat では以下の設定の使用を推奨しています。

```
Satellite Server (Satellite Server Discovery plug-in) <--> Satellite Capsule
(Satellite Capsule Discovery plug-in) <--> Discovered Host (Satellite
Discovery image)
```

Satellite Discovery プラグインは 3 つの異なるコンポーネントで構成されています。

Satellite Server Discovery プラグイン

これは Satellite Server で実行され、検出されたホストと使用できるように API および UI 機能を提供します。*tfm-rubygem-foreman_discovery* パッケージにはこのプラグインが含まれます。

Satellite Capsule Server Discovery プラグイン

これは、プロビジョニングネットワークで検出されたホストと Satellite Server 間の通信プロキシです。*rubygem-smart_proxy_discovery* パッケージにはこのプラグインが含まれます。

Satellite Discovery イメージ

これは、Red Hat Enterprise Linux をベースとする最小オペレーティングシステムです。このオペレーティングシステムはホスト上で PXE で起動し、初期のハードウェア情報を取得し、Satellite Server にチェックインするために使用されます。検出されたホストは、Anaconda で再起動するまで Satellite Discovery イメージを実行し続けます。その後にプロビジョニングプロセスを開始します。*foreman-discovery-image* パッケージにはこのイメージが含まれます。これは TFTP サービスを提供する Satellite Capsule Server にインストールされる必要があります。

11.1. Satellite Discovery プラグインの設定

以下のセクションでは、Satellite Discovery プラグインを設定する方法および Satellite Server で PXE 起動テンプレートを準備する方法について説明します。

11.1.1. Satellite Discovery イメージのデプロイ

Satellite Discovery イメージを含むパッケージを、TFTP サービスを提供する Satellite Capsule Server (Satellite Server 自体ではない) にインストールします。

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

このパッケージには、Linux カーネルと PXE で起動される検出されるホストに使用される起動可能な ISO ファイルとしての初期 RAM ディスクイメージが含まれます。以下のコマンドを実行してパッケージの内容を検査します。以下のような出力が生成されます。

```
$ rpm -ql foreman-discovery-image
/usr/share/foreman-discovery-image
/usr/share/foreman-discovery-image/fdi-image-rhel_7-2.1.0-20150212.1.iso
```

このパッケージのインストール時に、カーソルと ISO ファイルのイメージが TFTP ディレクトリーに抽出され、イメージおよびカーネルの最新バージョンへのシンボリックリンクが作成されます。PXE 起動のプロビジョニングテンプレートのシンボリックリンクを使用すると、*foreman-discovery-image* パッケージがアップグレードされるたびにテンプレートのバージョンを変更する必要はありません。以下は例になります。

```
$ find /var/lib/tftpboot/boot
/var/lib/tftpboot/boot
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-img
/var/lib/tftpboot/boot/fdi-image-rhel_7-2.1.0-20150212.1-mlinuz
/var/lib/tftpboot/boot/fdi-image-rhel_7-img
/var/lib/tftpboot/boot/fdi-image-rhel_7-mlinuz
```

上記の出力の最後の 2 行には、PXE 起動プロビジョニングテンプレートで使用するシンボリックリンクが含まれます。詳細は、[「PXE 起動の設定」](#) を参照してください。

注記

現時点では、Red Hat Enterprise Linux 6 の Satellite 6 インストールの場合でも Red Hat Enterprise Linux 7 Discovery イメージのみが提供されます。*foreman-discovery-image* パッケージのアップグレード時に実行中の検出されたホストがある場合、すぐにそれらすべてを再起動してイメージの更新されたバージョンをロードします。これは、Satellite 6 Web UI、CLI、または API で実行できます。

11.1.2. PXE 起動の設定

不明のホストがプロビジョニングネットワークで起動する場合、Satellite Server は、ローカルハードドライブから起動するという単一オプションを含む PXELinux ブートメニューを提供します。以下の手順では、ハードウェアの検出を有効にするためにこの動作を変更する方法について説明します。これには、PXE Linux グローバルデフォルトテンプレートの複数の変数の変更が必要になります。これらの変数については以下で説明します。

- ※ テンプレートの KERNEL および APPEND 行は、*foreman-discovery-image* パッケージのインストール時に作成されるシンボリックリンクを使用します ([「Satellite Discovery イメージのデプロイ」](#) を参照してください)。URL は `/var/lib/tftpboot/` ディレクトリーに相対しています。**APPEND** パラメーターが単一行に指定されていることを確認します。
- ※ **proxy.type** 変数は **proxy** (推奨) または **foreman** のいずれかに設定できます。変数が **proxy** に設定されている場合、すべての通信は Satellite Capsule Server を経由します。変数が **foreman** に設定されている場合、Satellite Server に直接通信されます。本章の例では、**proxy.type** が **proxy** に設定されていることを想定しています。

- ※ **proxy.url** 変数は、**proxy.type** 設定に応じて Satellite Capsule Server または Satellite Server の URL を指定します。**HTTP** および **HTTPS** 設定はどちらもサポートされます。Satellite Capsule Server にアクセスする場合 (**proxy.type=proxy**) のデフォルトポートは 9090 で、Satellite Server との直接通信の場合 (**proxy.type=foreman**) は 80 です。
- ※ **IPAPPEND 2** 設定は、プロビジョニングネットワークに接続されたインターフェースを検出します。イメージは、このオプションが削除または変更されている場合には正常に起動しません。

手順11.1 PXE 起動を設定するには、以下を実行します。

1. Satellite Web UI で、ホスト → プロビジョニングテンプレート に移動します。
2. **PXELinux global default** テンプレートを編集します。以下のメニューエントリーをテンプレートに追加します。

```

LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/fdi-image-rhel_7-vmlinuz
APPEND initrd=boot/fdi-image-rhel_7-img rootflags=loop
root=live:/fdi.iso rootfstype=auto ro rd.live.image acpi=force
rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 nomodeset
proxy.url=https://SATELLITE_CAPSULE_URL:9090 proxy.type=proxy
IPAPPEND 2

```

3. **ONTIMEOUT** 変数を変更して新規のメニューエントリーがデフォルトになるように設定します。

```
ONTIMEOUT discovery
```

4. プロビジョニングテンプレート ページの上部にある **PXE デフォルトのビルド** をクリックします。これにより、TFTP プロキシに **pxelinux.cfg/default** ファイルを再作成するようとの指示が出されます。デフォルトテンプレートが変更されるたびにこの手順を繰り返されるため、変更が TFTP Satellite Capsule Server にデプロイされることを確認できます。

上記の手順の代わりに、**proxy.url** 変数を PXE 起動テンプレートから省略することもできます。この場合、Discovery イメージは DNS 設定ファイルで **x-foreman.tcp** という名前の SRV レコードを検索します。この場合、**proxy.url** 変数は **proxy** に設定する必要があります。また、DNS サーバーも適切に設定されている必要があります。たとえば、以下の設定ステートメントは Capsule Server が HTTPS で使用されるように指定します。

```
_x-foreman._tcp SRV 0 5 9090 capsule
```

ここで、*capsule* は DNS 設定に含まれる Capsule Server の名前です。



注記

Satellite 6.2 は、ホストを検出できるすべてのサブネットに 1 つの Capsule Server URL のみを指定できます。テンプレートはサブネットごとに使用できないため、すべてのネットワークで DNS エイリアス名を使用します。または、代わりに SRV レコードを使用します。

**重要**

DHCP 設定からの DNS サーバーは、**BOOTIF** 変数で指定されるインターフェースの場合にのみ考慮に入れます。**BOOTIF** は、PXE テンプレートの **IPAPPEND** 変数で自動的に設定されます。つまり、システムに複数の NIC が設定されている場合、DNS は起動元のインターフェースでのみ機能します。

11.1.3. グローバル Discovery 設定の確認

Satellite Web UI で Discovery プラグインに関連するグローバル設定を確認することができます。管理 → 設定 に移動して、**Discovered** タブを開きます。以下の設定に留意してください。

discovery_organization、discovery_location

これらの変数は、検出されたホストを配置する場所を指定します。デフォルトでは、検出されたホストは最初に作成された組織とロケーションの下に自動的に配置されます。

discovery_fact

この変数は検出されたホストの MAC アドレスを判別するために使用される受信ファクトを指定します。デフォルトでは、PXELinux BOOTIF カーネルコマンドラインのオプションが使用されます。

discovery_auto

この変数は、指定されたルールに基づいて自動的なプロビジョニングを有効にします。デフォルトでは「false」に設定されます。Red Hat では、**discovery_auto** を有効にする前に設定を手動のプロビジョニングでテストすることを推奨しています。詳細は、[「検出されたホストのプロビジョニング」](#) を参照してください。

discovery_fact_column

この変数により、検出されたホストの一覧の追加行に Facter がレポートするファクトを追加できます。

11.2. Satellite Capsule Server Discovery プラグインの設定

foreman_url 設定が Satellite Capsule Server 設定ファイルにあることを確認します。設定は以下のように表示されます。

```
# grep foreman_url /etc/foreman-proxy/settings.yml
:foreman_url: https://satellite.example.com
```

satellite-installer コマンドはこの変数を自動的に設定しますが、Red Hat ではホストが正常に反応し、通信をブロックするファイアウォールのルールがないことを確認することを推奨しています。

11.2.1. Discovery 用サブネットの設定

検出されたホストを持つすべてのサブネットが Satellite Capsule Server で通信できるように設定する必要があります。Satellite Web UI で **インフラストラクチャー → サブネット** に移動し、ホストの検出を実行する必要がある各サブネットに必要な Capsule Server を選択し、これが Discovery Capsule Server に接続されていることを確認します。

Capsule Server で Discovery プラグインが有効にされていることを確認するには、インフラストラクチャー → **Capsules (スマートプロキシ)** に移動します。Discovery プラグインが Capsule Server に関連付けられている機能の一覧の表示されるはずです。更新 をクリックして一覧を最新の状態であることを確認します。

11.2.2. Discovery プラグインでの Hammer の使用

Discovery プラグインで **hammer** コマンドを使用するには、以下のように `/etc/hammer/cli.modules.d/foreman_discovery.yml` で Discovery プラグインを有効にする必要があります。

```
:foreman_discovery:
  :enable_module: true
```

hammer が使用するファイルおよびディレクトリーの詳細については、[hammer configuration directories](#) [5] を参照してください。

11.2.3. ユーザーの各種パーミッションの確認

最初の起動時に、Satellite Capsule Server Discovery プラグインは **Discovery** というロールを作成します。このロールを管理者以外のユーザーに割り当て、それらのユーザーが Discovery プラグインを使用できるようにします。または、**perform_discovery** パーミッションを既存ロールに割り当てることもできます。ロールおよびパーミッションの詳細については、『サーバー管理ガイド』の [ユーザーの作成および管理](#) を参照してください。

11.3. 検出されたホストのプロビジョニング

Satellite Server および Capsule Server の両方で Discovery プラグインを正常に設定した後に、ベアメタルホストを自動的に検出することができます。これを実行するには、「[PXE 起動の設定](#)」で説明されているように PXE 設定テンプレートで設定されたプロビジョニングネットワークのマシンを起動します。マシンは Satellite Server に自動的に登録され、Satellite Web UI の **ホスト → 検出されたホスト** の一覧に表示されます。

検出されたホストは手動でプロビジョニングすることも、自動プロビジョニングを設定することもできます。

11.3.1. ホストの手動プロビジョニング

以下の手順では、Satellite Web UI で検出されたホストを手動でプロビジョニングする方法を説明します。

手順11.2 検出されたホストを手動でプロビジョニングするには、以下を実行します。

1. **ホスト → 検出されたホスト** に移動します。
2. プロビジョニングするホストを選択し、**プロビジョニング** をクリックします。
3. ホストの **編集** ページで、必要な詳細を入力し、**保存** をクリックします。

ホストの設定の保存時に、Satellite は TFTP サーバーのホストの PXELinux ファイルを変更し、検出されたホストを再起動します。次に選択したオペレーティングシステムのインストーラーを起動し、最終的にはインストールしたオペレーティングシステムを起動します。

既存ホストのプロビジョニングを再度実行する場合は、マシンからオペレーティングシステムを削除してこれを再起動します。その後ホストは **検出されたホスト** ページに再度表示されます。

11.3.2. 検出されたホストの使用停止

Red Hat Satellite で特定のホストを管理する必要がなくなった場合は、ホストを使用停止にして検出されないようにする必要があります。

手順11.3 検出されたホストを使用停止にするには、以下を実行します。

1. ホストをシャットダウンします。
2. ホスト → 検出されたホスト に移動します。
3. 名前 列で、使用停止にするホストを検出し、対応する **編集** ドロップダウンメニューから **削除** を選択します。

11.3.3. ホストの自動プロビジョニング

Satellite 6.2 では、ホストグループをプロビジョニングされたホストに割り当て、プロビジョニングを自動的にトリガーするプロビジョニングルールを定義することができます。

手順11.4 プロビジョニングルールを作成するには、以下を実行します。

1. 設定 → **Discovery rules** に移動します。
2. **新規ルール** をクリックします。プロビジョニングルールの以下のパラメーターを指定します。
 - ※ **名前** はルールの一覧に表示されるルールの名前です。この名前には、スペースや非英数字を含めることができません。
 - ※ **検索** は、特定のルールに検出されたホストを一致させる検索ステートメントです。スコープ指定構文を使用してこれを定義することができます。スコープ指定検索の例については、[「スコープ指定の検索構文」](#) を参照してください。
 - ※ **ホストグループ** は、プロビジョニングプロセスを開始する前に一致するホストに割り当てられるホストグループです。選択したホストグループには必要なパラメーターがすべて設定されていることを確認します。必要なパラメーターにはアスタリスク (*) のマークが付けられます。
 - ※ **ホスト名** は、人間の判読できるホスト名を一致するホストの割り当てるパターンを定義します。空白のままにされている場合は、ホスト名はデフォルトで「macMACADDRESS」形式で割り当てられます。プロビジョニングテンプレートに使用される構文と同じ構文がこの場合にも使用されます。詳細および例については、[「ホスト名のパターン」](#) を参照してください。
 - ※ **ホストの制限** はルールに基づくプロビジョニングされるホストの最大数です。制限に達すると、ルールは 1 つ以上のホストが削除されるまで有効になりません。通常のユースケースとして、ホスト名やホストグループなどのプロビジョニングパラメーターをエントリーごとに変更する必要がある場合に、サーバーラックまたは行ごとにルールを使用します。この値をゼロ (0) に設定すると制限なしに設定することができます。
 - ※ **優先度** は、ルールの実行順序を指定します。値はゼロ以上である必要があります。値が低いほど優先度が高くなります。2 つのルールの優先度が同じ場合には、最初に検出されるルールが適用されます。
 - ※ **有効化** は、ルールを一時的に有効または無効にするオプションを提供します。
3. **送信** をクリックしてルールを保存します。

デフォルトで、Satellite はホストの自動検出を有効にしません。以下の手順では、**discovery_auto** 変数を有効にし、指定されたルールに基づいて自動プロビジョニングを行う方法について説明します。

手順11.5 自動プロビジョニングを有効にするには、以下を実行します。

手順**11.5** 自動プロビジョニングを有効にするには、以トを実行します。

1. Satellite Web UI で、管理 → 設定 → **Discovered** に移動します。
2. 名前 列で **discovery_auto** を見つけ、その値を **true** に設定します。
3. 保存 をクリックします。

ルールを定義した後に、Red Hat では、ホストに対して **Auto discover** ボタンを使用してホストを検出し、ルールを適用することを推奨します。これにより、グローバルオプションを有効にせずに自動プロビジョニングがトリガーされます。

11.3.4. スコープ指定の検索構文

このセクションでは、選択したパラメーターに応じて検出されたホストをフィルターするスコープ指定の検索構文を使用する方法を説明します。これは自動プロビジョニングのルールを作成する際に便利です ([「ホストの自動プロビジョニング」](#) を参照してください)。

Satellite Web UI の検索フィールドは自動補完に対応するため、検索構文の作成を容易にします。たとえば、ホスト → 検出されたホスト ページで検索パターンをテストすることができます。以下は通常の検索クエリーの例になります。

- facts.architecture = x86_64
- facts.bios_vendor ~ 'Dell*'
- facts.macaddress = "aa:bb:cc:dd:ee:ff"
- facts.macaddress_eth0 = "aa:bb:cc:dd:ee:ff"
- facts.ipaddress_eth1 ~ "192.168.*"
- facts.architecture ^ (x86_64,i386)



注記

スコープ指定検索のキャレット記号 (^) は「in」(SQL と同じ用法) を意味し、正規表現に使用される「starts with」を意味しません。スコープ指定の検索演算子の詳細の一覧については、https://github.com/wvanbergen/scoped_search/blob/master/lib/scoped_search/query_language/tokenizer.rb を参照してください。

Satellite 6.2 では、すべてのファクトは文字列のため、数値比較を実行することはできません。ただし、3つの重要なファクトが抽出され、数字に変換されます。それらについては、[表11.1「数値比較を可能にするファクト」](#) で説明されています。

表**11.1** 数値比較を可能にするファクト

検索パラメーター	説明	使用法の例
cpu_count	CPU の数	cpu_count >= 8
disk_count	割り当てられたディスクの数	disk_count < 10
disks_size	ディスク空き容量の合計 (MiB 単位)	disks_size > 1000000

11.3.5. ホスト名のパターン

このセクションでは、自動プロビジョニングのルールを作成する際に使用できるホスト名のパターンを一覧表示します ([「ホストの自動プロビジョニング」](#) を参照してください)。

ターゲットホスト名のテンプレートパターンには、プロビジョニングテンプレート (ERB) と同じ構文が使用されます。ドメインは自動的に追加されます。**@host** 属性のほかに、ランダムな整数の **rand()** 関数を利用できます。以下は例になります。

- application-server-<%= rand(99999) %>
- load-balancer-<%= @host.facts['bios_vendor'] + '-' + rand(99999) %>
- wwwsrv-<%= @host.hostgroup.name %>
- minion-<%= @host.discovery_rule.name %>
- db-server-<%= @host.ip.gsub('.', '-') + '-' + @host.hostgroup.subnet.name %>>



重要

ホスト名のパターンを作成する際に、作成されるホスト名が固有の名前であることを確認してください。ホスト名は数字で始めることができません。Facter (MAC アドレス、BIOS、またはシリアル ID) で提供される固有情報を使用するのか、またはホスト名をランダム化することは適切な方法です。

11.3.6. コマンドラインでの Discovery プラグインの使用

hammer コマンドを使用して検出に関連する特定のタスクを実行することができます。**hammer -h** コマンドを実行して設定を確認します。

```
$ hammer -h | grep discovery
discovery          Manipulate discovered hosts.
discovery_rule     Manipulate discovered rules.
```

hammer discovery -h コマンドを使用して利用可能なオプションを表示します。たとえば、以下のコマンドを使用して検出されるホストを再起動できます (以下は ID を 130 と想定した場合)。

```
$ hammer discovery reboot -id 130
Host reboot started
```

11.4. 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を使って Satellite Discovery イメージを拡張することができます。イメージで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。

最初に、以下のディレクトリ構造を作成します。

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

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

- ✧ **autostart.d** ディレクトリーには、起動時のホストが Satellite に登録される前に、イメージによって POSIX の順序で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 変数に追加されます。ここにバイナリーファイルを配置し、これらを autostart スクリプトで使用できます。
- ✧ **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します。
- ✧ **lib** ディレクトリーは LD_LIBRARY_PATH 変数に追加され、**lib/ruby** は RUBYLIB 変数に追加され、**/bin** のバイナリーファイルを正常に実行できるようにします。

新規ディレクティブおよびオプションが既存の環境変数 (PATH、LD_LIBRARY_PATH、RUBYLIB および FACTERLIB) に追加されます。スクリプトの内容へのパスを明示的に指定する必要がある場合、zip コンテンツがイメージ上の **/opt/extension** ディレクトリーに抽出されます。

上記のディレクトリー構造を作成した後に、これを以下のコマンドで zip アーカイブにパッケージ化します。

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

複数の zip ファイルを作成できますが、これらは Discovery イメージの同じ場所に抽出されるため、ファイル名が同じ場合に後の zip のファイルが先のファイルを上書きします。

Discovery イメージで使用される拡張子が認識されるように、TFTP サーバーに zip ファイルを検出イメージと共に配置してから、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、以下の構文を使用します。

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

PXE テンプレートの更新についての詳細は、[「PXE 起動の設定」](#) を参照してください。

11.5. Satellite 検出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** に正常に表示されない場合、以下の設定領域を調べてエラーの切り分けをします。

- ✧ デフォルト PXE Linux テンプレートの再デプロイを試行します。
- ✧ 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。
- ✧ デフォルト PXE Linux テンプレートで **proxy.url** および **proxy.type** オプションを確認します。
- ✧ DNS がそのイメージに対して適切に機能しているか確認するか、またはデフォルト PXE Linux テンプレートの **proxy.url** オプションで 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 500 MB 以上のメモリーがあることを確認します。メモリーがこれより小さくなると、イメージがインメモリーで抽出される必要があるために各種のランダムなカーネルパニックのエラーが生じる可能性があります。

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

検出されたホストの最初の仮想コンソールはシステムログのために予約されます。とく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**: 初回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**: ネットワーク情報

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

```
fdi.ssh=1 fdi.rootpw=redhat
```

[5] <https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md#locations>

第12章 Satellite ホストでのジョブの実行

Red Hat Satellite は任意のコマンドをホストで実行することをサポートします。これは **リモート実行** と呼ばれています。リモート実行はデフォルトで Satellite Server で有効にされますが、必要なすべての Capsule Server では手動で有効にする必要があります。通信は Capsule Server 経由で行われます。これは、Satellite Server にはターゲットホストへの直接のアクセスが不要であり、多数のホストを管理するために拡張可能であることを意味します。コマンドはプロビジョニングテンプレートやパーティションテーブルと同様の方法でカスタマイズできます。いくつかのジョブテンプレートがデフォルトで含まれおり、これらを使用してコマンドを実行できます。[「ジョブテンプレートのセットアップ」](#) を参照してください。



注記

Capsule Server のベースシステムは Satellite Server の内部 Capsule のクライアントであるため、このセクションは Capsule Server を含む Satellite Server に接続されるホストのすべてのタイプに適用されます。

コマンドは 1 度に複数のホストに対して実行でき、デプロイメントに適した変数をコマンドで使用できます。変数の値はホストのファクト、Smart Class パラメーター、Smart 変数またはホストパラメーターで設定できます。さらに、コマンドの実行時にテンプレートのカスタム値を指定することができます。[「ジョブの実行」](#) を参照してください。

以下の一覧は、リモート実行を使用する方法のいくつかの例を示しています。

- ソフトウェアパッケージのインストール、更新、または削除
- 設定管理エージェントのブートストラップ
- Puppet、Salt または Chef 実行のトリガー

デフォルトでは、それぞれの Capsule はリモート実行機能が無効にされた状態でインストールされます。Capsule Server でリモート実行を使用するには、これを有効にする必要があります。有効にするには、以下のコマンドを実行します。

```
# satellite-installer --scenario capsule --enable-foreman-proxy-plugin-remote-execution-ssh
```

リモート実行が Capsule Server で実行されていることを確認するには、Web UI で **インフラストラクチャー → Capsules (スマートプロキシ)** に移動します。Capsule Server は、**SSH が実行する機能** 列に一覧表示されるはずです。

デフォルトで、Satellite Server は Katello エージェントではなくリモート実行を使用するように設定されています。必要な場合、これらの設定は、まずカスタムジョブテンプレートを作成し、次に Web UI の **管理 → リモート実行機能** に移動してこれらの新規テンプレートを選択して変更できます。変更するそれぞれのアクションについて、ラベルを選択してから使用するジョブテンプレートを選択します。

12.1. リモートコマンドのセキュアな接続の確立

リモート実行に使用される SSH キーは Capsule のインストール時に自動的に作成され、設定は `/etc/foreman-proxy/settings.d/remote_execution_ssh.yml` ファイルにあります。それらには以下のオプションが含まれます。

ssh_identity_file

SSH キーをロードするファイルです。デフォルトでは `/usr/share/foreman-`

`proxy/.ssh/id_rsa_foreman_proxy` に設定されます。

local_working_dir

リモート実行に必要なスクリプトを実行するために Satellite または Capsule で使用されるディレクトリーです。デフォルトでは `/var/tmp` に設定されます。

remote_working_dir

リモート実行ジョブを実行するために使用されるクライアントシステムのディレクトリーです。デフォルトでは、`/var/tmp` に設定されます。



注記

クライアントシステムで `noexec` が `/var/` ボリュームまたはファイルシステムに設定されている場合、`remote_working_dir` を変更します。そうしないとスクリプトを実行できず、リモート実行ジョブが失敗します。

代替ディレクトリーを使用することが必要な場合、`new_place` などの新規ディレクトリーを作成し、デフォルトのディレクトリーから SELinux コンテキストをコピーします。以下は例になります。

```
# chcon --reference=/var new_place
```

SELinux ラベルの使用についての詳細は、『SELinux User's and Administrator's Guide』の [Maintaining SELinux Labels](#) セクションを参照してください。

リモート実行のための SSH キーの配布

リモート実行を有効にするには、公開 SSH キーを Capsule から管理するホストに送信します。Capsule からターゲットホストに公開鍵を送信する方法として 3 つの方法があります。

- ※ キーを手動で配布するには、Capsule で以下のコマンドを実行します。

```
# ssh-copy-id -i ~foreman-proxy/.ssh/id_rsa_foreman_proxy.pub  
root@target.example.com
```

ここで、`target.example.com` はターゲットホストのホスト名です。管理するターゲットホストのそれぞれについてこれを繰り返し実行します。

- ※ Satellite API を使用して公開鍵を Capsule から直接ダウンロードするには、それぞれのターゲットホストに対して以下のコマンドを実行します。

```
# curl https://myproxy.example.com:9090/ssh/pubkey >>  
~/.ssh/authorized_keys
```

ここで、`myproxy.example.com` は Capsule のホスト名を表します。

- ※ 公開鍵を新規にプロビジョニングされたホストに組み込むには、以下の行を含めるように **Kickstart default finish** テンプレートなどを変更します。

```
<%= snippet 'remote_execution_ssh_keys' %>
```

12.2. リモートコマンドの設定および実行

リモートホストで実行するコマンドはジョブテンプレートとして定義する必要があります。ジョブテンプレートを定義した後はこれを複数回使用することができます。

12.2.1. ジョブテンプレートのセットアップ

Satellite はジョブの実行に使用できる数多くのデフォルトジョブテンプレートを提供します。それらは、[ホスト → ジョブテンプレート](#) の下にあります。デフォルトテンプレートの中でそれぞれのニーズに合うテンプレートを見つけた場合は、[「ジョブの実行」](#)に進みます。

デフォルトテンプレートは独自のテンプレートを作成するためのベースとして使用することもできます。デフォルトジョブテンプレートは編集用にロックされるので、テンプレートを変更するにはまずこのクローンを作成する必要があります。

手順12.1 ジョブテンプレートのクローンを作成するには、以下を実行します。

1. **ホスト → ジョブテンプレート** に移動します。
2. **新規ジョブテンプレート** をクリックします。または、**アクション** 列のドロップダウンメニューで **クローン** を選択して既存のテンプレートを変更できます。
3. ジョブテンプレートを設定します。
 - a. **テンプレート** タブで、ジョブテンプレートの固有の名前を入力します。**デフォルト** を選択すると、テンプレートをすべての組織およびロケーションで利用可能にできます。**テンプレートエディター** を使用してテンプレートを手動で入力することも、**参照** をクリックしてテキストファイルからアップロードすることもできます。テンプレートは **Embedded Ruby (ERB)** テンプレート構文を使用します。詳細は、[「詳細テンプレートの作成」](#) を参照してください。パワー関連のアクションを実行する場合などには詳細なテンプレートが必要になります。**Power Action - SSH Default** テンプレートをカスタムテンプレートに組み込む方法については、[例12.4「テンプレートにパワー操作を組み込む」](#) を参照してください。
 - b. **Job (ジョブ)** タブで、ジョブカテゴリーを定義 (独自のカテゴリーを定義するか、[表 12.1「デフォルトのジョブテンプレートカテゴリー」](#) に一覧表示されているデフォルトカテゴリーから選択する) したり、実効ユーザーを定義したりすることができます。これらの設定は、ジョブの起動時にも設定できます ([手順12.2「リモートジョブを実行するには、以下を実行します。」](#) を参照してください)。また、テンプレートコマンドの入力パラメーターを定義できます。これらのパラメーターはジョブの実行時に要求されます。
 - c. 残りのタブでは、テンプレートのタイプ、組織およびロケーションの設定、およびテンプレート履歴の表示を実行できます。
4. **送信** をクリックします。ページの更新時に、新規のテンプレートがジョブテンプレートの一覧に表示されるはずです。



重要

ホストの編集ページの **パラメーター** タブでのみ表示できるパラメーターはジョブテンプレートの入力パラメーターとして使用できることに注意してください。

表12.1 デフォルトのジョブテンプレートカテゴリー

ジョブテンプレートのカテゴリー

Packages	パッケージ関連のアクションを実行するためのテンプレートです。デフォルトで、インストール、更新、および削除アクションが含まれています。
Puppet	ターゲットホストで Puppet ホストを実行するためのテンプレートです。
Power	パワー関連のアクションを実行するためのテンプレートです。デフォルトで、再起動およびシャットダウンアクションが含まれます。
Commands	リモートホストでカスタムコマンドを実行するためのテンプレートです。
Services	サービス関連のアクションを実行するためのテンプレートです。デフォルトで、開始、停止、再起動、およびステータスアクションが含まれます。
Katello	コンテンツ関連のアクションを実行するためのテンプレートです。これらのテンプレートは主として Satellite Web UI の各種の場所で使用されます (たとえば、コンテンツホストの一括操作のための UI など) が、エラータのインストールなどの各種操作を実行するために個別に使用できます。

例12.1 restorecon テンプレートの作成

この例は、**Run Command - restorecon** というテンプレートを作成する方法を示します。これは、ターゲットホストのすべてのファイル用にデフォルトの **SELinux** コンテキストを復元します。

1. ホスト → ジョブテンプレート に移動します。新規ジョブテンプレート をクリックします。
2. 名前 フィールドに **Run Command - restorecon** を挿入します。デフォルト を選択してテンプレートをすべての組織で利用できるようにします。以下のテキストを テンプレートエディター に追加します。

```
restorecon -RvF <%= input("directory") %>
```

`<%= input("directory") %>` の文字列は、ジョブの呼び出し時にユーザー定義のディレクトリーに置き換えられます。

3. **Job (ジョブ)** タブで、以下のアクションを実行します。
 - a. ジョブカテゴリー を **Commands** に設定します。
 - b. **入力を追加** をクリックして、ジョブのカスタマイズを可能にします。ディレクトリー を **Name (名前)** フィールドに挿入します。入力名は **テンプレートエディター** で指定される値と一致している必要があります。
 - c. **Required** をクリックし、コマンドがユーザー指定のパラメーターなしでは実行されないようにします。
 - d. **Input type** ドロップダウンリストから **ユーザー入力** を選択します。また、**Target directory for restorecon** などのようにジョブの呼び出しの際に表示される **Description (説明)** を指定します。
4. **送信** をクリックします。

このテンプレートに基づいてジョブを実行する方法については、[例12.2「複数のホストでの restorecon テンプレートの実行」](#) を参照してください。

12.2.2. ジョブの実行

このセクションでは、1 つ以上のホストに対してジョブテンプレートに基づくジョブを実行する方法を説明します。

手順12.2 リモートジョブを実行するには、以下を実行します。

1. **ホスト → すべてのホスト** に移動し、ジョブのターゲットホストを選択します。検索フィールドを使用してホストの一覧の範囲を制限することができます。
2. 画面右上の **アクションを選択** メニューから **ジョブを実行** を選択します。これにより、**ジョブ呼び出し** ページに移動します。または、ターゲットが 1 つのホストのみである場合、その名前をクリックし、ホスト情報ページで **ジョブを実行** をクリックします。**実行** ボタンを使用して **ジョブテンプレート** ページからジョブを起動することもできます。
3. **ジョブ呼び出し** ページで、主なジョブ設定を定義します。
 - a. 使用する **ジョブカテゴリー** および **ジョブテンプレート** を選択します。これらの設定は必須です。
 - b. オプションとして、**ブックマーク** の一覧で保存された検索文字列を選択してターゲットホストを指定します。
 - c. オプションとして、**Search query** を入力し、ターゲットホストの範囲をさらに狭めることができます。**解決** 行には、クエリーの影響を受けるホストの数が表示されます。更新ボタンを押して、クエリーの変更後の数の再計算を実行します。プレビューアイコンにはターゲットホストが一覧表示されます。
 - d. 残りの設定は、選択したジョブテンプレートによって異なります。カスタムパラメーターをテンプレートに追加する方法については、[手順12.1「ジョブテンプレートのクローンを作成するには、以下を実行します。」](#) を参照してください。
4. オプションとして、**詳細フィールドを表示** をクリックします。一部の詳細設定はジョブテンプレートによって異なります。以下は一般的な設定です。
 - ✦ **実効ユーザー**: ジョブを実行するためにユーザーを定義します。デフォルトは SSH ユーザーです。
 - ✦ **同時実行レベル**: 1 度に実行されるジョブの最大数を定義します。**タイムスパン** では、定義された期間にジョブの実行を分散できます。これらの設定は、インフラストラクチャー上での大規模なジョブ実行の負荷分散に役立ちます。
 - ✦ **クエリーのタイプ**: 検索クエリーが評価される時を定義します。これは、クエリーをスケジュールされるタスクに対して常に最新の状態に保つのに役立ちます。
5. ジョブをすぐに実行する場合は、**スケジュール** が **今すぐ実行** に設定されていることを確認します。さらに 1 回限りの将来のジョブを定義したり、再帰的に実行されるジョブをセットアップすることもできます。再帰的に実行されるジョブについては、開始日と終了日、実行の回数と頻度を定義することができます。また cron 構文を施与うして繰り返しを定義することもできます。
6. **送信** をクリックします。これにより、**Job Overview (概要)** ページが表示され、ジョブの完了時にはジョブのステータスも表示されます。

例12.2 複数のホストでの restorecon テンプレートの実行

以下の例では、複数のホストで [例12.1「restorecon テンプレートの作成」](#) で作成されたテンプレートに基づいて、複数のホストに対してジョブを実行する方法を示します。このジョブは **/home/** ディレクトリの下にあるすべてのファイルで SELinux コンテキストを復元します。

1. **ホスト → すべてのホスト** に移動し、ターゲットホストを選択します。**アクションの選択** ドロップダウンリストから **ジョブを実行** を選択します。
2. **ジョブ呼び出し** ページで、**Commands** ジョブカテゴリーを選択し、**Run Command - restorecon** ジョブテンプレートを選択します。

3. **directory** フィールドに **/home** を入力します。
4. **スケジュール** を **今すぐ実行** に設定します。
5. **送信** をクリックします。**ジョブ呼び出し** ページに移動します。ここでジョブ実行のステータスを監視できます。

12.2.3. ジョブの監視

実行中のジョブの進捗を監視できます。これは、トラブルシューティングが必要になる場合に役立ちます。

手順12.3 ジョブを監視するには、以下を実行します。

1. ジョブのページに移動します。このページは、**今すぐ実行** が設定されているジョブをトリガーすると自動的に表示されます。スケジュールされたジョブを監視するには、**モニター → ジョブ** に移動して、検査するジョブ実行を選択します。
2. 「ジョブ」ページで **ホスト** タブをクリックします。これにより、ジョブが実行されているホストの一覧が表示されます。
3. **ホスト** 列で、検査するホストの名前をクリックします。これにより、ジョブの実行をリアルタイムでモニターできる **Detail of Commands** (コマンドの詳細) ページが表示されます。
4. いつでも **ジョブに戻る** をクリックして **Job Details** (ジョブの詳細) ページに戻ることができます。

12.2.4. 詳細テンプレートの作成

ジョブテンプレートの作成時に、既存テンプレートを **テンプレートエディター** フィールドにインポートできます。これは **レンダリング** と呼ばれています。これにより、テンプレートを組み合わせたり、一般的なテンプレートからより具体的なテンプレートを作成したりすることができます。

以下のテンプレートを使ってデフォルトのテンプレートを組み合わせ、**httpd** サービスを Red Hat Enterprise Linux システムにインストールし、これを起動できます。

```
<%= render_template 'Package Action - SSH Default', :action => 'install',
:package => 'httpd' %>
<%= render_template 'Service Action - SSH Default', :action => 'start',
:service_name => 'httpd' %>
```

上記のテンプレートはレンダリングされるテンプレートのパラメーター値を直接指定します。ユーザーがジョブ実行時にレンダリングされたテンプレートの入力を定義できるようにする **input()** メソッドを使用することもできます。たとえば、以下の構文を使用できます。

```
<%= render_template 'Package Action - SSH Default', :action => 'install',
:package => input("package") %>
```

上記のテンプレートを使用して、レンダリングされたテンプレートからパラメーター定義をインポートする必要があります。これを実行するには、**ジョブ** タブに移動し、**外部入力セットを追加** をクリックしてから **Target template** (ターゲットテンプレート) ドロップダウンリストでレンダリングされたテンプレートを選択します。すべてのパラメーターをインポートするか、またはコンマ区切りの一覧を指定することができます。

例12.3 restorecon テンプレートのレンダリング

この例は、[例12.1「restorecon テンプレートの作成」](#)で作成される **Run command - restorecon** テンプレートに派生するテンプレートを作成する方法を示しています。このテンプレートでは、ジョブ実行時のユーザー入力不要で、ターゲットホストの **/home/** ディレクトリー下のすべてのファイルで SELinux コンテキストを復元します。

[「ジョブテンプレートのセットアップ」](#) で説明されている新規テンプレートを作成し、テンプレートエディター 画面で以下の文字列を指定します。

```
<%= render_template("Run Command - restorecon", :directory => "/home") %>
```

例12.4 テンプレートにパワー操作を組み込む

以下の例では、再起動などのパワー操作を実行するためのジョブテンプレートをセットアップする方法を示します。この手順は、Satellite が再起動時に切断の例外をエラーとして解釈するのを防ぐため、ジョブのリモート実行が正常に機能します。

[「ジョブテンプレートのセットアップ」](#) で説明されている新規テンプレートを作成し、テンプレートエディター 画面で以下の文字列を指定します。

```
<%= render_template("Power Action - SSH Default", :action => "restart") %>
```

12.3. グローバル設定

Satellite のリモート実行機能は、その動作を設定するために使用できる数多くのグローバル設定を提供しています。それらは [表12.2「リモート実行用のグローバル設定」](#) に一覧表示されています。これらの設定を確認し、更新するには、**管理 → 設定** に移動し、**Remote Execution** タブをクリックします。

表12.2 リモート実行用のグローバル設定

パラメーター名	説明
remote_execution_effective_user	これは、任意のジョブについてのデフォルトの有効ユーザーです。ジョブが実行される際に、プロセスの有効ユーザーはそれに応じて変更されます (sudo による変更など)。このオプションはジョブテンプレートおよびジョブ呼び出し別にオーバーライドできます。
remote_execution_effective_user_method	ターゲットホストで有効ユーザーを設定するために使用するメソッドを指定します。現時点で su および sudo がサポートされています。
remote_execution_fallback_proxy	ホストでリモート実行が設定された Capsule を検索します。これはホストがサブネットではないか、またはサブネットにリモート実行が有効にされた Capsule がない場合に役立ちます。
remote_execution_global_proxy	ホストに割り当てられた Capsule 外にあるリモート実行 Capsule を検索します。ロケーションまたは組織が有効にされている場合、検索はホストの組織またはロケーションに限定されます。

パラメーター名	説明
remote_execution_ssh_user	Capsule が SSH でターゲットに接続される際に使用されるデフォルトユーザーです。 remote_execution_ssh_user 変数を設定し、これをホスト別にオーバーライドできます。 この設定は、ホスト、ホストグループ、オペレーティングシステム、ドメイン、ロケーションまたは組織別に行われます。さらに、 remote_execution_effective_user とは異なるユーザーを指定することもできます。
remote_execution_sync_templates	データベースのシード処理時にジョブテンプレートをディスクと同期させるかどうかを定義します。



重要

/etc/foreman/settings.yaml 設定ファイルでグローバルパラメーターを設定できますが、このファイルに追加する手動の変更は次の **satellite-installer** 実行時に上書きされます。そのため、Red Hat では、これらのパラメーターを web UI で変更することを推奨します。または、コンソールから **foreman-rake config** コマンドを使用することができます。

12.3.1. リモート実行用の Capsule の選択

リモート実行では、Capsule Server がホストで指定されたジョブを実行することが必要になります。デフォルトで **remote execution provider** 機能が有効にされたホストの組織およびロケーション内の Capsule はこれらのジョブを実行できると見なされています。**remote_execution_global_proxy** 変数を **false** に設定すると、この動作を無効にできます。これは、ネットワークの分離のためにすべての Capsule を使用できない場合などのより複雑な環境で必要になる場合があります。この設定では、Capsule のプールを各サブネットに割り当て、ジョブの負荷はそれらの間で分散されます。

または、**remote_execution_fallback_proxy** 変数を **true** に設定してフォールバックモードを有効にできます。この設定では、Capsule でもリモート実行が設定されている場合に、リモート実行に Puppet マスターなどのホストに関連付けられた Capsule が使用されます。

12.4. リモート実行用のパーミッションの委任

ターゲットにするホストを含め、インフラストラクチャー内で実行するジョブとそれを実行するユーザーを制御できます。リモート実行機能は 2 つの組み込みロールを提供します。

- ✦ **Remote Execution Manager (リモート実行マネージャー)**: このロールは、すべてのリモート実行機能および機能性へのアクセスを許可します。
- ✦ **Remote Execution User (リモート実行ユーザー)**: このロールは、ジョブの実行のみを許可します。ジョブテンプレートを変更するパーミッションは提供されません。

Remote Execution User (リモート実行ユーザー) ロールのクローンを作成し、そのフィルターをカスタマイズして詳細度を高めることができます。**view_job_templates** パーミッションでフィルターを調整する場合、ユーザーは一致するジョブテンプレートに基づいてジョブを確認し、トリガーすることのみが可能です。**view_hosts** および **view_smart_proxies** パーミッションを使用すると、ロールに表示されるホストまたは Capsule を制限できます。

execute_template_invocation パーミッションはジョブの実行が開始される直前にチェックされる特殊なパーミッションです。このパーミッションは、特定のホストで実行できるジョブテンプレートを定義します。これにより、パーミッションの指定時に詳細度をさらに高めることができます。ロールおよびパー

ミッションの使用についての詳細は、『Server Administration Guide』の[Creating and Managing Roles](#)を参照してください。

以下の例は、**`execute_template_invocation`** パーミッションのフィルターを示しています。

```
name = Reboot and host.name = staging.example.com
name = Reboot and host.name ~ *.staging.example.com
name = "Restart service" and host_group.name = webservers
```

上記の例の最初の行により、ユーザーは選択したホストで **Reboot** テンプレートを実行できます。2 番目の行は、**`.staging.example.com`** で終わる名前を持つホストのプールを定義します。3 番目の行はテンプレートをホストグループにバインドします。



注記

ユーザーに割り当てるパーミッションは時間の経過と共に変更されます。ユーザーに将来実行するスケジュールされたジョブがあり、パーミッションが変更された場合、パーミッションがジョブ実行の直前にチェックされるため、これによる実行の失敗が生じる可能性があります。

第13章 ホストコレクションの設定

ホストコレクションアプリケーションタブは、管理者による以下の実行を可能にするシステム管理ツールです。

- ※ ホストのコレクションへの追加。
- ※ パッケージやエラータ、またはパッケージグループをホストコレクションの全ホストメンバーに一括インストール。
- ※ 特定のパッケージやエラータ、または特定のパッケージグループの全ホストメンバーへの更新。

13.1. ホストコレクションの作成

以下の手順では、ホストコレクションから作成する方法を示します。

手順13.1 ホストコレクションを作成するには、以下を実行します。

1. **ホスト → ホストコレクション** をクリックします。
2. **新しいホストコレクション** をクリックします。
3. ホストコレクションの名前と説明を追加します。
4. **無制限のコンテンツホスト** のチェックを外し、グループに許可されるホストの最大数を指定します。または、無制限のホストがホストコレクションに加わるのを許可する場合はこのボタンのチェックを付けたままにします。
5. **保存** をクリックします。

13.2. ホストのホストコレクションへの追加

以下の手順では、ホストをホストコレクションに追加する方法を示します。

前提条件

ホストをホストコレクションに追加するには、ホストを Red Hat Satellite に登録する必要があります。ホストの登録方法については、[「ホストの登録」](#) を参照してください。

手順13.2 ホストをホストコレクションに追加するには、以下を実行します。

1. **ホスト → ホストコレクション** をクリックします。
2. ホストの追加先となるホストコレクションをクリックします。
3. **ホスト** タブの **追加** サブタブを選択します。
4. テーブルから追加するホストを選択してから、**選択を追加** をクリックします。

13.3. コンテンツのホストコレクションへの追加

以下の手順は、Red Hat Satellite でコンテンツをホストコレクションに追加する方法を示しています。

13.3.1. パッケージのホストコレクションへの追加 |

以下の手順では、パッケージをホストコレクションに追加する方法を示します。

前提条件

- ※ 追加するコンテンツが既存リポジトリのいずれかで利用可能であるか、またはリポジトリにない場合は、この手順を開始する前にリポジトリに追加しておく必要があります。
- ※ コンテンツはホストの割り当てられる環境にプロモートする必要があります。

手順**13.3** パッケージをホストコレクションに追加するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. パッケージの追加先となるホストコレクションをクリックします。
3. コレクションの各種アクション タブで、パッケージのインストール、削除、および更新 をクリックします。
4. すべてのパッケージを更新するには、**Update All Packages** ボタンをクリックしてデフォルトメソッドを使用します。または、ボタン右側のドロップダウンアイコンを選択して使用するメソッドを選択します。**リモート実行 - 最初にカスタマイズ** メニューエントリを選択すると、**ジョブ呼び出し** ページに移動します。ここでアクションをカスタマイズすることができます。
5. 必要に応じて **パッケージ** または **パッケージグループ** ラジオボタンを選択します。
6. パッケージまたはパッケージグループの名前をフィールドに指定してから、以下のいずれかをクリックします。
 - ※ **インストール** — デフォルトメソッドを使用して新規パッケージをインストールします。または、ボタンの右側にあるドロップダウンアイコンを選択して使用するメソッドを選択します。**リモート実行 - 最初にカスタマイズ** メニューエントリを選択すると、**ジョブ呼び出し** ページに移動します。ここでアクションをカスタマイズすることができます。
 - ※ **更新** — デフォルトメソッドを使用してホストコレクションの既存パッケージを更新します。または、ボタン右側のドロップダウンアイコンを選択して使用するメソッドを選択します。**リモート実行 - 最初にカスタマイズ** メニューエントリを選択すると、**ジョブ呼び出し** ページに移動します。ここで、アクションをカスタマイズすることができます。

13.3.2. エラータのホストコレクションへの追加

以下の手順では、エラータをホストコレクションに追加する方法を示します。

前提条件

- ※ 追加するエラータは既存リポジトリのいずれかで利用可能であるか、またはリポジトリにない場合は、この手順を開始する前にリポジトリに追加しておく必要があります。
- ※ エラータはホストの割り当てられる環境にプロモートする必要があります。

手順**13.4** エラータをホストコレクションに追加するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. エラータの追加先となるホストコレクションを選択します。
3. コレクションの各種アクション タブで、エラータのインストール をクリックします。

4. ホストコレクションに追加するエラータを選択し、**選択をインストール** ボタンをクリックしてデフォルトのメソッドを使用します。または、ボタン右側のドロップダウンアイコンを選択して使用するメソッドを選択します。**リモート実行 - 最初にカスタマイズ** メニューエントリーを選択すると、**ジョブ呼び出し** ページに移動します。ここでアクションをカスタマイズすることができます。

13.4. ホストコレクションからのコンテンツの削除

以下の手順では、パッケージをホストコレクションから削除する方法を示します。

手順13.5 コンテンツをホストコレクションから削除するには、以下を実行します。

1. **ホスト → ホストコレクション** をクリックします。
2. パッケージを削除するホストコレクションをクリックします。
3. **コレクションの各種アクション** タブで、**パッケージのインストール、削除、および更新** をクリックします。
4. 必要に応じて **パッケージ** または **パッケージグループ** ラジオボタンを選択します。
5. パッケージまたはパッケージグループの名前をフィールドに指定します。
6. **削除** ボタンをクリックし、デフォルトメソッドを使用してパッケージまたはパッケージグループを削除します。または、ボタン右側にあるドロップダウンアイコンを選択して使用するメソッドを選択します。**リモート実行 - 最初にカスタマイズ** メニューエントリーを選択すると、**ジョブの呼び出し** ページに移動します。ここでアクションをカスタマイズすることができます。

13.5. ホストコレクションのライフサイクル環境またはコンテンツビューの変更

以下の手順では、ホストコレクションの割り当てられたライフサイクル環境またはコンテンツビューを変更する方法を示します。

手順13.6 ホストコレクションのライフサイクル環境またはコンテンツビューを変更するには、以下を実行します。

1. **ホスト → ホストコレクション** をクリックします。
2. ライフサイクル環境またはコンテンツビューを変更する必要があるホストコレクションを選択します。
3. **コレクションの各種アクション** タブで、**割り当て済みのライフサイクル環境またはコンテンツビューの変更** をクリックします。
4. ホストコレクションに割り当てるライフサイクル環境を選択します。
5. 必要なコンテンツビューをドロップダウンリストから選択します。
6. **割り当て** をクリックします。

13.6. ホストコレクションからのホストの削除

以下の手順では、ホストをホストコレクションから削除する方法を示します。

手順13.7 ホストをホストコレクションから削除するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. 必要なホストコレクションを選択します。
3. ホスト タブで、一覧/削除 サブタブを選択します。
4. ホストコレクションから削除するホストを選択し、**選択を削除** をクリックします。

13.7. ホストコレクションの削除

以下の手順では、ホストコレクションを削除する方法を示します。

手順13.8 ホストコレクションを削除するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. 削除するホストコレクションを選択します。
3. **削除** ボタンをクリックします。警告ボックスが表示されます。

ホストコレクション **ホストコレクション名** を削除してもよろしいですか？

4. **削除** をクリックします。

13.8. ホストコレクションのクローン作成

以下の手順では、ホストコレクションのクローンを作成する方法を示します。

手順13.9 ホストコレクションのクローンを作成するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. 左側のパネルで、クローン作成するホストコレクションをクリックします。
3. **コレクションのコピー** をクリックします。
4. クローン作成されたコレクションの名前を指定します。
5. **作成** をクリックします。

13.9. ホストコレクションの詳細の確認

以下の手順では、ホストコレクションの詳細を確認する方法を説明します。

手順13.10 ホストコレクションの詳細を確認するには、以下を実行します。

1. ホスト → ホストコレクション をクリックします。
2. 確認するホストコレクションを選択し、**詳細** タブに移動します。

付録A テンプレート作成の参照

Embedded Ruby (ERB) は、プレーンテキストと Ruby コードを組み合わせるテンプレートをベースにしてテキストファイルを生成するためのツールです。Red Hat Satellite は、[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」](#)、[Red Hat Satellite Provisioning Guide](#))、リモート実行 [ジョブテンプレート](#) ([12章Satellite ホストでのジョブの実行](#))、パーティションテーブルのテンプレート ([「パーティションテーブル」](#))、および [スマートパラメーター](#) ([「スマート変数の設定」](#)) で ERB 構文を使用します。このセクションでは、ERB テンプレートで使用できる Satellite 固有の関数および変数と使用例のいくつかについての概要を示します。Red Hat Satellite で提供されるデフォルトテンプレート (ホスト → プロビジョニングテンプレート、ホスト → ジョブテンプレート) には ERB 構文のサンプルが十分にあることも留意してください。

ホストのプロビジョニングまたはリモートジョブの実行時に、ERB のコードが実行され、変数はホスト固有の値に置き換えられます。このプロセスは、[レンダリング](#) と呼ばれています。Satellite Server ではセーフモードのレンダリングオプションがデフォルトで有効にされており、これにより、有害なコードがテンプレートから実行されることを防ぐことができます。

A.1. ERB テンプレートの作成

以下は ERB 構文を要約したポイントになります。

- ✧ **<% %>**: Ruby コードを ERB テンプレート内で囲むマークです。コードはテンプレートのレンダリング時に実行されます。これには Ruby の制御フロー構造および Satellite 固有の関数および機能を含めることができます。たとえば、以下のようになります。

```
<% if @host.operatingsystem.family == "Redhat" &&
@host.operatingsystem.major.to_i > 6 %>
systemctl <%= input("action") %> <%= input("service") %>
<% else %>
service <%= input("service") %> <%= input("action") %>
<% end -%>
```

- ✧ **<%= %>**: コード出力はテンプレートに挿入されます。これは変数の置き換えに便利です。たとえば、以下のようになります。

```
echo <%= @host.name %>
```

- ✧ **<% -%>**, **<%= -%>**: デフォルトで改行文字は、行末で閉じられている場合に Ruby ブロックの後に挿入されます。この動作を抑制するには、囲みマークを変更します。以下のテンプレートは例になります。

```
curl <%= @host.ip -%>
/mydir
```

上記のテンプレートは以下のようにレンダリングされます。

```
curl <%= @host.ip %>/mydir
```

これはレンダリングされるテンプレートの行数を減らすために使用されます (Ruby 構文で許可される場合)。

- ✧ **<%# %>**: テンプレートのレンダリング時に無視されるコメントを囲むマークです。

```
<%# A comment %>
```

A.2. ERB テンプレートのトラブルシューティング

Satellite web UI では、特定ホストについてのテンプレートのレンダリングを検証するための 2 つの方法を提供しています。

- ✱ **テンプレートエディターによる直接的な方法:** テンプレートの編集時に (ホスト → パーティションテーブル、ホスト → プロビジョニングテンプレート、または ホスト → ジョブテンプレート)、テンプレートタブで **プレビュー** をクリックしてからドロップダウンメニューでホストを選択します。次にテンプレートは選択したホストのパラメーターを使用してテキストフィールドにレンダリングされます。プレビューが失敗した場合は、そこからテンプレートの問題を特定することができます。
- ✱ **ホストの詳細ページを使用する方法:** ホスト → すべてのホスト でホストを選択し、テンプレート タブをクリックしてホストに関連付けられたテンプレートを一覧表示します。選択したテンプレートの横にあるドロップダウンメニューから **確認** を選択し、そのレンダリングされたバージョンを表示します。

A.3. Satellite 固有の関数および変数

このセクションでは、ERB テンプレート用の Satellite 固有の関数および変数を一覧表示します。それらの中にはどのテンプレートで使用できるものもあれば、使用が制限されるものもあります。たとえば、ジョブテンプレートは @host 変数のみを受け入れ、[表A.4「キックスタート固有の変数」](#) の変数はキックスタートテンプレートでのみ適用できます。

以下の表に記載されている関数は、すべての種類のテンプレートで 사용할 ことができます。

表A.1 汎用的な関数

名前	説明
indent(n)	コードブロックを n スペース分インデントします。インデントされていないスニペットテンプレートの使用時に便利です。
foreman_url(kind)	完全な URL を、ホストでレンダリングされた指定タイプのテンプレートに返します。たとえば、「provision」タイプのテンプレートは通常 http://HOST/unattended/provision にあります。
snippet(name)	指定されたスニペットテンプレートをレンダリングします。プロビジョニングテンプレートをネスト化するのに便利です。
snippets(file)	Foreman データベースの指定されたスニペットをレンダリングし、データベースにない場合は unattended/snippets/ ディレクトリーからこれをロードします。
snippet_if_exists(name)	指定されたスニペットをレンダリングし、指定された名前を持つスニペットが見つからない場合は省略します。

例A.1 スニペットおよびインデント関数の使用

以下の構文は、**subscription_manager_registration** スニペットをテンプレートにインポートし、4 スペース分インデントします。

```
<%= indent 4 do
  snippet 'subscription_manager_registration'
end %>
```

以下の関数はジョブテンプレートで使用できます。使用例については、[「詳細テンプレートの作成」](#) を参照

してください。

表A.2 ジョブテンプレート固有の関数

名前	説明
input(input_name)	ジョブ実行時に指定された入力の値を返します。
render_template(name, parameters)	汎用的な snippet() 関数と同様に指定されたテンプレートを返しますが、引数をテンプレートに渡すことができます。

以下の変数は、テンプレート内でのホストの使用を可能にします。

表A.3 ホスト固有の変数および関数

名前	説明
@host.architecture	ホストのアーキテクチャーです。
@host.bond_interfaces	すべてのボンディングインターフェースの配列を返します。 注記 を参照してください。
@host.capabilities	システムプロビジョニングの方法には、ビルド (キックスタートなど) またはイメージのいずれかを使用できます。
@host.certname	ホストの SSL 証明書名です。
@host.diskLayout	ホストのディスクレイアウトです。オペレーティングシステムから継承できます。
@host.domain	ホストのドメインです。
@host.environment	ホストの Puppet 環境です。
@host.facts	Facter からファクトの Ruby ハッシュを返します。たとえば、出力の 'ipaddress' ファクトにアクセスするには、@host.facts['ipaddress'] を指定します。
@host.grub_pass	ホストの GRUB パスワードを返します。
@host.hostgroup	ホストのホストグループです。
@host.info['parameters']	ホストパラメーターについての情報が含まれる Ruby ハッシュを返します。たとえば、@host.info['parameters']['lifecycle_environment'] を使用してホストのライフサイクル環境を取得します。
@host.image_build?	ホストがイメージを使用してプロビジョニングされる場合は true を返します。
@host.interfaces	プライマリーインターフェースを含む利用可能なすべてのホストインターフェースの配列が含まれます。 注記 を参照してください。
@host.interfaces_with_identifier('IDs')	指定された ID を持つインターフェースの配列を返します。複数の ID の配列を入力として渡すことができます (例: @host.interfaces_with_identifier(['eth0', 'eth1'])). 注記 を参照してください。
@host.ip	ホストの IP アドレスです。
@host.location	ホストの位置です。
@host.mac	ホストの MAC アドレスです。
@host.managed_interfaces	管理対象インターフェースの配列を返します (BMC およびボンディングインターフェースを除く)。 注記 を参照してください。
@host.medium	割り当てられたオペレーティングシステムのインストールメディアです。
@host.name	ホストの完全名です。
@host.operatingsystem.family	オペレーティングシステムファミリーです。

名前	説明
@host.operatingsystem.major	割り当てられたオペレーティングシステムのメジャーバージョンの番号です。
@host.operatingsystem.minor	割り当てられたオペレーティングシステムのマイナーバージョンの番号です。
@host.operatingsystem.name	割り当てられたオペレーティングシステムの名前です。
@host.operatingsystem.boot_files_uri(@host.medium,@host.architecture)	カーネルおよび initrd への完全パスで、アレイを返します。
@host.os.medium_uri(@host)	プロビジョニングに使用される URI です (インストールメディアに設定されるパス)。
@host.param_false?(name)	指定された名前のホストパラメーターが false と評価される場合に false を返します。
@host.param_true?(name)	指定された名前のホストが true と評価される場合に true を返します。
@host.params['parameter_name']	指定されたパラメーターの値を返します。
@host.primary_interface	ホストのプライマリーインスタンスを返します。
@host.provider	コンピュータリソースプロバイダーです。
@host.provision_interface	ホストのプロビジョニングインターフェースを返します。インターフェースオブジェクトを返します。
@host.ptable	パーティションテーブル名です。
@host.puppetmaster	ホストが使用する必要のある Puppet マスターです。
@host.pxe_build?	ホストがネットワークまたは PXE を使用してプロビジョニングされる場合に true を返します。
@host.shortname	ホストの省略名です。
@host.sp_ip	BMC インターフェースの IP アドレスです。
@host.sp_mac	BMC インターフェースの MAC アドレスです。
@host.sp_name	BMC インターフェースの名前です。
@host.sp_subnet	BMC ネットワークのサブネットです。
@host.subnet.dhcp	DHCP プロキシがこのホストに設定されている場合は true を返します。
@host.subnet.dns_primary	ホストのプライマリー DNS サーバーです。
@host.subnet.dns_secondary	ホストのセカンダリー DNS サーバーです。
@host.subnet.gateway	ホストのゲートウェイです。
@host.subnet.mask	ホストのサブネットマスクです。
@host.url_for_boot(:initrd)	このホストに関連付けられる initrd イメージへの完全パスです。変数を補間しないので推奨されません。
@host.url_for_boot(:kernel)	このホストに関連付けられたカーネルへの完全パスです。変数を補間しないので推奨されません。boot_files_uri が優先されます。
@provisioning_type	プロビジョニングのタイプに応じて「host」または「hostgroup」と等しくなります。
@static	ネットワーク設定が静的な場合、 true を返します。
@template_name	レンダリングされるテンプレートの名前です。
grub_pass	md5pass 引数でラップされる GRUB パスワードを返します (例: --md5pass=#{@host.grub_pass})。
ks_console	ポートを使用して組み立てられる文字列およびカーネル行に追加できるボーレートを返します (例: console=ttyS1,9600)。
root_pass	システムに設定される root パスワードを返します。



注記

@host.interfaces または **@host.bond_interfaces** などのネットワークインターフェースに関連するホスト変数は、アレイで分類されるインターフェースデータを返します。特定インターフェースのパラメーター値を抽出するには、Ruby メソッドを使用してアレイを解析します。たとえば、アレイの最初のインスタンスについての情報を取得し、これをキックスタートテンプレートで使用するには、以下を実行します。

```
<% myinterface = @host.interfaces.first %>
IPADDR="<%= myinterface.ip %>"
NETMASK="<%= myinterface.subnet.mask %>"
GATEWAY="<%= myinterface.subnet.gateway %>"
```

インターフェース名のアレイを抽出するなどの目的でインターフェースアレイを繰り返すことができます。以下を使用します。

```
<% ifnames = []
@host.interfaces.each do |i|
  ifnames.push(i.name)
end %>
```

例A.2 ホスト固有変数の使用

以下の例では、ホストで Puppet および Puppetlabs リポジトリが有効にされているかどうかをチェックします。

```
<%
pm_set = @host.puppetmaster.empty? ? false : true
puppet_enabled = pm_set || @host.param_true?('force-puppet')
puppetlabs_enabled = @host.param_true?('enable-puppetlabs-repo')
%>
```

以下の例では、パッケージ関連の決定に使用できるホストのオペレーティングシステムのマイナーおよびメジャーバージョンを取得する方法を示します。

```
<%
os_major = @host.operatingsystem.major.to_i
os_minor = @host.operatingsystem.minor.to_i
%>

<% if ((os_minor < 2) && (os_major < 14)) -%>
...
<% end -%>
```

以下の例では、ホストのスニペットで DHCP ブートモードが有効な場合に 'kickstart_networking_setup' スニペットをインポートします。


```
<% subnet = @host.subnet %>
<% if subnet.respond_to?(:dhcp_boot_mode?) -%>
<%= snippet 'kickstart_networking_setup' %>
<% end -%>
```

一般的な Ruby メソッドのほとんどはホスト固有の変数で使用できます。たとえば、ホストの IP アドレスの最後のセグメントを抽出するには、以下を使用できます。

```
<% @host.ip.split('.').last %>
```

以下の変数は、キックスターンプロビジョニングテンプレート内で使用されるように設計されています。

表A.4 キックスタート固有の変数

名前	説明
@arch	ホストのアーキテクチャー名です。@host.architecture.name と同じです。
@dynamic	使用されているパーティションテーブルが %pre スクリプト (表の最初の行に #Dynamic オプションがある) の場合 true を返します。
@epel	epel-release rpm の正しいバージョンを自動インストールするコマンドです。%post スクリプトで使用されます。
@mediapath	URL コマンドを提供する詳細なキックスタート行です。
@osver	オペレーティングシステムのメジャーバージョンの番号です。 @host.operatingsystem.major と同じです。