



Red Hat Satellite 6.14

Red Hat Satellite の管理

ユーザーと権限の管理、組織と場所の管理、Satellite のバックアップと復元、
Satellite のメンテナンスなど

Red Hat Satellite 6.14 Red Hat Satellite の管理

ユーザーと権限の管理、組織と場所の管理、Satellite のバックアップと復元、Satellite のメンテナンスなど

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

法律上の通知

Copyright © 2024 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本ガイドでは、Red Hat Satellite 6 Server を設定および管理する手順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 を正常にインストールしておく必要があります。

目次

RED HAT ドキュメントへのフィードバック (英語のみ)	5
第1章 RED HAT SATELLITE へのアクセス	6
1.1. KATELLO ルート CA 証明書のインポート	6
1.2. SATELLITE へのログイン	6
1.3. RED HAT IDENTITY MANAGEMENT 認証情報を使用した SATELLITE HAMMER CLI へのログイン	7
1.4. RED HAT IDENTITY MANAGEMENT 認証情報を使用した FIREFOX ブラウザーでの SATELLITE WEB UI へのログイン	8
1.5. RED HAT IDENTITY MANAGEMENT 認証情報を使用した CHROME ブラウザーでの SATELLITE WEB UI へのログイン	9
1.6. SATELLITE WEB UI のナビゲーションタブ	10
1.7. パスワードの変更	11
1.8. 管理ユーザーパスワードのリセット	11
1.9. ログインページでのカスタムメッセージの設定	12
第2章 RED HAT SATELLITE の起動および停止	13
第3章 SATELLITE SERVER のクローン	14
3.1. クローン作成プロセスの概要	14
3.2. 前提条件	14
3.3. PULP データの考慮事項	15
3.4. SATELLITE SERVER のクローン	16
第4章 事前定義済みプロファイルを使用した SATELLITE SERVER の調整	19
第5章 内部 SATELLITE データベースから外部データベースへの移行	21
5.1. 外部データベースとして POSTGRESQL を使用する際の注意点	21
5.2. 外部データベース用のホストの準備	22
5.3. POSTGRESQL のインストール	23
5.4. 外部データベースへの移行	24
第6章 ANSIBLE COLLECTIONS を使用した SATELLITE の管理	26
6.1. SATELLITE ANSIBLE モジュールのインストール	26
6.2. SATELLITE ANSIBLE モジュールの表示	26
第7章 組織の管理	27
7.1. 組織の作成	27
7.2. 組織コンテキストの設定	28
7.3. 組織デバッグ証明書の作成	29
7.4. 組織デバッグ証明書を使用したりポジトリリーコンテンツの閲覧	29
7.5. 組織の削除	30
第8章 ロケーションの管理	32
8.1. ロケーションの作成	32
8.2. 複数のロケーションの作成	32
8.3. ロケーションコンテキストの設定	33
8.4. ロケーションの削除	33
第9章 ユーザーとロールの管理	34
9.1. ユーザーの管理	34
9.2. SSH キーの管理	37
9.3. PERSONAL ACCESS TOKEN の管理	38
9.4. ユーザーグループの作成と管理	39
9.5. ロールの作成および管理	40

9.6. SATELLITE で利用可能な事前定義済みロール	43
9.7. 詳細なパーミッションフィルタリング	44
第10章 メール通知の設定	48
10.1. 電子メール通知の種類	48
10.2. 電子メール通知の設定	48
10.3. メールの配信テスト	49
10.4. 電子メール通知のテスト	49
10.5. ホストの電子メール通知設定の変更	50
第11章 SATELLITE SERVER および CAPSULE SERVER のバックアップ	51
11.1. バックアップサイズの予測	51
11.2. SATELLITE SERVER または CAPSULE SERVER の完全バックアップの実行	53
11.3. PULP コンテンツなしでのバックアップの実行	54
11.4. 増分バックアップの実行	55
11.5. 週次の完全バックアップ後の日次増分バックアップ例	56
11.6. オンラインバックアップの実行	56
11.7. スナップショットバックアップの実行	57
11.8. バックアップを実行する際のホワイトリスト化とスキップの手順	58
第12章 バックアップからの SATELLITE SERVER または CAPSULE SERVER の復元	59
12.1. 完全バックアップからの復元	59
12.2. 増分バックアップからの復元	60
12.3. 仮想マシンのスナップショットを使用した CAPSULE SERVER のバックアップと復元	60
第13章 SATELLITE SERVER または CAPSULE SERVER の名前の変更	62
13.1. SATELLITE SERVER の名前の変更	62
13.2. CAPSULE SERVER の名前の変更	63
第14章 SATELLITE SERVER のメンテナンス	66
14.1. 監査レコードの手動削除	66
14.2. 監査レコードの自動削除	66
14.3. 監査レコードの匿名化	66
14.4. レポートのレコード削除	66
14.5. 未使用タスクのクリーニング機能の設定	67
14.6. タスクレコードの削除	67
14.7. ID によるタスクの削除	68
14.8. 完全なディスクからのリカバリー	68
14.9. SATELLITE SERVER または CAPSULESERVER のベースオペレーティングシステムでのパッケージの管理	69
14.10. POSTGRESQL 領域の確保	70
14.11. オンデマンドリポジトリからのスペースの再利用	70
第15章 カスタム SSL 証明書の更新	72
15.1. SATELLITE SERVER でのカスタム SSL 証明書の更新	72
15.2. CAPSULE SERVER でのカスタム SSL 証明書の更新	73
第16章 問題のログとレポート	75
16.1. デバッグに役立つログレベルの引き上げ	75
16.2. HAMMER CLI を使用してログレベルを設定する	78
16.3. ジャーナルまたはファイルベースのログへのログの設定	79
16.4. 個別のロガーを使用した選択的デバッグ	80
16.5. 個別のロガーの概要	80
16.6. サービスのステータスの取得	81
16.7. サービスの再起動:	82

16.8. ログ情報を処理するためのユーティリティー	82
16.9. ログファイルディレクトリー	82
16.10. システムジャーナルメタデータ	83
第17章 リソースの監視	85
17.1. RED HAT SATELLITE コンテンツダッシュボードの使用	85
17.2. RSS 通知の設定	89
17.3. SATELLITE SERVER の監視	89
17.4. CAPSULE SERVER の監視	89
第18章 WEBHOOK の使用	92
18.1. WEBHOOK への移行	92
18.2. WEBHOOK のインストール	93
18.3. WEBHOOK テンプレートの作成	93
18.4. WEBHOOK の作成	94
18.5. 利用可能な WEBHOOK イベント	95
18.6. シェルフック	98
18.7. SHELLHOOKS プラグインのインストール	99
18.8. WEBHOOK を使用したシェルフックスクリプトへの引数の指定	99
18.9. CURL を使用したシェルフックスクリプトへの引数の指定	99
18.10. 引数を入力するためのシェルフックの作成	100
第19章 検索およびブックマーク機能	102
19.1. 検索クエリーの構築	102
19.2. フリーテキスト検索の使用	104
19.3. ブックマークの管理	104
付録A 管理設定	106
A.1. 一般設定	106
A.2. SATELLITE タスク設定	107
A.3. テンプレート同期設定	108
A.4. 検出された設定	110
A.5. ブートディスク設定	112
A.6. RED HAT クラウド設定	113
A.7. コンテンツ設定	113
A.8. 認証設定	118
A.9. メール設定	120
A.10. 通知設定	121
A.11. プロビジョニング設定	121
A.12. ファクト設定	125
A.13. 管理設定	127
A.14. リモート実行設定	127
A.15. ANSIBLE 設定	129

RED HAT ドキュメントへのフィードバック (英語のみ)

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

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

前提条件

- [Red Hat アカウント](#) が登録されている。

手順

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

第1章 RED HAT SATELLITE へのアクセス

Red Hat Satellite のインストールと設定が終わったら、Satellite web UI を使用して Satellite にログインし、追加の設定を行います。

1.1. KATELLO ルート CA 証明書のインポート

Satellite に初めてログインする場合は、デフォルトの自己署名証明書を使用していることを通知する警告が表示され、ルート CA 証明書がブラウザーにインポートされるまでこのブラウザーを Satellite に接続できない可能性があります。以下の手順を使用して、Satellite でルート CA 証明書を見つけ、ブラウザーにインポートします。

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

前提条件

- Red Hat Satellite がインストールされ、設定されていること。

手順

1. Satellite Server の完全修飾ドメイン名を特定します。

```
# hostname -f
```

2. Web ブラウザーで以下の完全修飾ドメイン名を指定して、Satellite Server の **pub** ディレクトリーにアクセスします。

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

3. Satellite に初めてアクセスする場合は、信頼できない接続を警告するメッセージが Web ブラウザーに表示されます。自己署名証明書を承認し、Satellite の URL をセキュリティ例外として追加し、設定を上書きします。この手順は、使用しているブラウザーによって異なる場合があります。セキュリティ例外を承認する前に Satellite の URL が有効であることを確認します。
4. **katello-server-ca.crt** を選択します。
5. 証明書を認証局としてブラウザーにインポートして信頼し、Web サイトを特定します。

CLI 手順

1. Satellite CLI から、Satellite web UI へのアクセスに使用するマシンに、**katello-server-ca.crt** ファイルをコピーします。

```
# scp /var/www/html/pub/katello-server-ca.crt username@hostname:remotefile
```

2. ブラウザーで、**katello-server-ca.crt** 証明書を認証局としてインポートして信頼し、Web サイトを識別します。

1.2. SATELLITE へのログイン

さらに設定するには、Web ユーザーインターフェイスを使用して Satellite にログインします。

前提条件

- ブラウザーに Katello ルート CA 証明書がインストールされていること。詳細は、[「Katello ルート CA 証明書のインポート」](#) を参照してください。

手順

1. Web ブラウザーで以下の完全修飾ドメイン名を指定して、Satellite Server にアクセスします。

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

2. 設定プロセスで作成したユーザー名とパスワードを入力します。設定プロセス時にユーザーを作成されなかった場合は、デフォルトのユーザー名 **admin** が使用されます。ログインに問題がある場合は、パスワードをリセットできます。詳細は、[「管理ユーザーパスワードのリセット」](#) を参照してください。

1.3. RED HAT IDENTITY MANAGEMENT 認証情報を使用した SATELLITE HAMMER CLI へのログイン

このセクションでは、Red Hat Identity Management (IdM) ログインとパスワードを使用して Satellite Hammer CLI にログインする方法について説明します。

前提条件

- Satellite Server を Red Hat Identity Management に登録して認証に Red Hat Identity Management を使用するように設定する。具体的には、Satellite Web UI と Satellite API の両方へのアクセスを有効にしている。詳細は、[オンラインネットワーク環境への Satellite Server のインストール](#) の [Red Hat Identity Management の使用](#) を参照してください。
- この手順を実行するホストが Red Hat Identity Management 認証情報を使用してユーザーを Satellite Hammer CLI にログインするように設定されている。詳細は、[オンラインネットワーク環境への Satellite Server のインストール](#) の [Red Hat Identity Management ユーザー認証を使用するための Hammer CLI の設定](#) を参照してください。
- ホストが Red Hat Identity Management クライアントである。
- Red Hat Identity Management サーバーが実行中で、ホストから到達可能である

手順

1. Satellite ユーザーに代わって Kerberos Ticket-Granting Ticket (TGT) を取得します。

```
$ kinit idm_user
```



警告

Red Hat Identity Management を認証プロバイダーとして設定するときに、Satellite API と Satellite Web UI の両方へのアクセスを有効にした場合、ユーザーが Kerberos TGT を受信した後に攻撃者が API セッションを取得できるようになります。ユーザーが以前にブラウザーなど、どこにも Satellite ログイン認証情報を入力していなかった場合でも、攻撃は可能です。

2. 自動ネゴシエート認証が有効になっていない場合は、TGT を使用して Hammer を手動で認証します。

```
$ hammer auth login negotiate
```

3. オプション: コレクション内にある、キャッシュされたすべての Kerberos チケットを破棄します。

```
$ kdestroy -A
```

Kerberos チケットを破棄した後も、ログインしたままです。

検証

- **Hammer** コマンドを使用して、システムが再度認証を要求しないようにします。

```
$ hammer host list
```



注記

Hammer からログアウトするには、**hammer auth logout** と入力します。

1.4. RED HAT IDENTITY MANAGEMENT 認証情報を使用した FIREFOX ブラウザーでの SATELLITE WEB UI へのログイン

このセクションでは、Firefox ブラウザーで、Red Hat Identity Management (IdM) ログインとパスワードを使用して Satellite Web UI にログインする方法について説明します。

前提条件

- Satellite Server を Red Hat Identity Management に登録し、認証に Red Hat Identity Management を使用するようにサーバーを設定している。詳細は、[オンラインネットワーク環境への Satellite Server のインストール](#) の [Red Hat Identity Management の使用](#) を参照してください。
- Firefox ブラウザーを使用して Satellite Web UI にログインしているホストが Red Hat Identity Management クライアントである。
- Red Hat Identity Management ログインとパスワードが有効である。

- Red Hat では、最新の安定した Firefox ブラウザーを使用することを推奨。
- Firefox ブラウザーが Single Sign-On (SSO) 用に設定されている。詳細は、[Red Hat Enterprise Linux での認証と認可の設定](#) の [Single Sign-On に Kerberos を使用するための Firefox の設定](#) を参照してください。
- Red Hat Identity Management サーバーが実行中で、ホストから到達可能である

手順

1. Red Hat Identity Management 認証情報を使用して、自分用の Kerberos Ticket-Granting ticket (TGT) を取得します。

```
$ kinit idm_user
Password for idm_user@EXAMPLE.COM:
```

2. ブラウザーのアドレスバーに、Satellite Server の URL を入力します。
自動的にログインされます。



注記

あるいは、最初の 2 つの手順を省略して、Satellite Web UI に表示されるフィールドにログインとパスワードを入力します。これは、Satellite Web UI にアクセスしているホストが Red Hat Identity Management クライアントではない場合には、唯一のオプションです。

1.5. RED HAT IDENTITY MANAGEMENT 認証情報を使用した CHROME ブラウザーでの SATELLITE WEB UI へのログイン

このセクションでは、Chrome ブラウザーを使用して、Red Hat Identity Management のログイン名とパスワードを使用して Satellite Web UI にログインする方法について説明します。

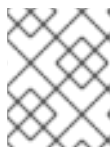
前提条件

- Satellite Server を Red Hat Identity Management に登録し、認証に Red Hat Identity Management を使用するようにサーバーを設定している。詳細は、[オンラインネットワーク環境への Satellite Server のインストール](#) の [Red Hat Identity Management の使用](#) を参照してください。
- Chrome ブラウザーを使用して Satellite Web UI にログインしているホストが、Red Hat Identity Management クライアントである。
- Red Hat Identity Management ログインとパスワードが有効である。
- Red Hat では、最新の安定した Chrome ブラウザーを使用することを推奨。
- Red Hat Identity Management サーバーが実行中で、ホストから到達可能である

手順

1. Chrome ブラウザーで Kerberos 認証を使用できるようにします。

```
$ google-chrome --auth-server-whitelist="example.com" --auth-negotiate-delegate-whitelist="example.com"
```

**注記**

ドメイン全体を許可リストに登録する代わりに、特定の Satellite Server を許可リストに登録することもできます。

- Red Hat Identity Management 認証情報を使用して、自分用の Kerberos Ticket-Granting Ticket (TGT) を取得します。

```
$ kinit idm_user
Password for idm_user@_EXAMPLE.COM:
```

- ブラウザーのアドレスバーに、Satellite Server の URL を入力します。
自動的にログインされます。

**注記**

あるいは、最初の 3 つの手順を省略して、Satellite Web UI に表示されるフィールドにログインとパスワードを入力することもできます。これは、Satellite Web UI にアクセスしているホストが Red Hat Identity Management クライアントではない場合には、唯一のオプションです。

1.6. SATELLITE WEB UI のナビゲーションタブ

ナビゲーションタブを使用して、Satellite Web UI を参照します。

ナビゲーションタブ	説明
すべてのコンテキスト	このタブをクリックすると、組織とロケーションが変更されます。組織やロケーションが選択されていない場合、デフォルト組織は 任意の組織 に、デフォルトロケーションは 任意のロケーション になります。このタブを使用して異なる値に変更します。
監視	サマリーダッシュボードおよびレポートを表示します。
Content	コンテンツ管理ツールを提供します。コンテンツビュー、アクティベーションキー、ライフサイクル環境などが含まれます。
Hosts	ホストインベントリおよびプロビジョニング設定ツールを提供します。
設定	一般的な設定ツール、およびホストグループや Puppet データを含むデータを提供します。
Infrastructure	Satellite が環境と対話する方法を設定するツールを提供します。
ユーザー名	ユーザーが個人情報を編集できるユーザー管理機能を提供します。
	環境に対する重要な変更が管理者に通知されるようにイベントの通知が表示されます。

ナビゲーションタブ	説明
管理	一般設定のほかに、ユーザーおよび RBAC 設定などの詳細設定を提供します。

1.7. パスワードの変更

以下の手順は、パスワードを変更する方法を示しています。

手順

1. Satellite Web UI で、右上隅のユーザー名をクリックします。
2. メニューから **マイアカウント** を選択します。
3. **現在のパスワード** フィールドに現在のパスワードを入力します。
4. **パスワード** フィールドに新しいパスワードを入力します。
5. **確認** フィールドに新しいパスワードを再入力します。
6. **Submit** をクリックして新しいパスワードを保存します。

1.8. 管理ユーザーパスワードのリセット

以下の手順を使用して、管理者パスワードを無作為に生成された文字にリセットするか、新しい管理者パスワードを設定します。

管理ユーザーパスワードのリセット:

1. Satellite Server がインストールされているベースのオペレーティングシステムにログインします。
2. 以下のコマンドを実行してパスワードをリセットします。

```
# foreman-rake permissions:reset
Reset to user: admin, password: qwJxBptxb7Gfcjj5
```

3. このパスワードを使用して、Satellite Web UI でパスワードをリセットします。
4. Satellite Server の `~/.hammer/cli.modules.d/foreman.yml` ファイルを編集し、新規パスワードを追加します。

```
# vi ~/.hammer/cli.modules.d/foreman.yml
```

`~/.hammer/cli.modules.d/foreman.yml` ファイルを更新しない限り、Hammer CLI では新規パスワードを使用できません。

新規の管理ユーザーパスワードの設定:

1. Satellite Server がインストールされているベースのオペレーティングシステムにログインします。

2. パスワードをリセットするには、以下のコマンドを入力します。

```
# foreman-rake permissions:reset password=new_password
```

3. Satellite Server の `~/.hammer/cli.modules.d/foreman.yml` ファイルを編集し、新規パスワードを追加します。

```
# vi ~/.hammer/cli.modules.d/foreman.yml
```

`~/.hammer/cli.modules.d/foreman.yml` ファイルを更新しない限り、Hammer CLI では新規パスワードを使用できません。

1.9. ログインページでのカスタムメッセージの設定

手順

1. Satellite web UI で、**Administer** > **Settings** に移動して、**General** タブをクリックします。
2. **ログインページフッターテキスト** の横にある編集ボタンをクリックして、ログインページに表示させるテキストを入力します。たとえば、自社で必須とされる警告メッセージなどにすることができます。
3. **Save** をクリックします。
4. Satellite web UI からログアウトして、ログインページで Satellite バージョン番号の下にカスタムテキストが表示されることを確認します。

第2章 RED HAT SATELLITE の起動および停止

Satellite には、コマンドラインから Satellite サービスを管理するための **satellite-maintain service** コマンドが含まれています。このコマンドは Satellite のバックアップの作成時に役に立ちます。バックアップ作成に関する詳細は、[11章 Satellite Server および Capsule Server のバックアップ](#) を参照してください。

satellite-installer コマンドを使用して Satellite をインストールした後に、すべての Satellite サービスは自動的に起動されて有効になります。これらのサービスのリストを表示するには、以下のコマンドを実行します。

```
# satellite-maintain service list
```

実行中のサービスのステータスを確認するには、以下のコマンドを実行します。

```
# satellite-maintain service status
```

Satellite サービスを停止するには、以下を実行します。

```
# satellite-maintain service stop
```

Satellite サービスを開始するには、以下を実行します。

```
# satellite-maintain service start
```

Satellite サービスを再起動するには、以下を実行します。

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

第3章 SATELLITE SERVER のクローン

Satellite Server のクローンを作成してインスタンスを作成し、インスタンスの別のマシンまたはオペレーティングシステムへのアップグレードや移行をテストできます。これは、アップグレードまたは移行中の柔軟性を高めるための任意のステップです。

Capsule Server で Satellite クローンツールは使用できません。代わりに、既存の Capsule Server をバックアップし、それをターゲットサーバーに復元してから、Capsule Server を再設定する必要があります。



注記

Satellite Server の新規インスタンスを作成する場合は、バックアップを復元した後に以前のインスタンスを廃止します。クローン作成されたインスタンスは、実稼働環境で並行して実行することは想定されていません。

用語

次の用語を理解するようにしてください。

ソースサーバー

クローンの起点。

ターゲットサーバー

ファイルをコピーしてソースサーバーのクローンを作成する新しいサーバー。

3.1. クローン作成プロセスの概要

1. ソースサーバーをバックアップします。
2. ソースサーバーからターゲットサーバーにクローンを作成します。
3. ソースサーバーの電源を切断します。
4. 新規ホスト名とターゲットサーバーの IP アドレスが一致するように、ターゲットサーバー上のネットワーク設定を更新します。
5. 非推奨の Katello Agent を使用しているコンテンツホストがある場合は、それらすべてのコンテンツホストとその Capsule で **goferd** サービスを再起動して、接続を更新します。
6. 新規ターゲットサーバーをテストします。

3.2. 前提条件

Satellite Server のクローンを作成するには、以下のリソースが用意されていることを確認します。

- ターゲットサーバーとなる Red Hat Enterprise Linux 8 の最小インストール。Red Hat Enterprise Linux 8 ソフトウェアグループやサードパーティーのアプリはインストールしないでください。サーバーがすべての必要な仕様に準拠していることを確認してください。詳細は、[オンラインネットワーク環境での Satellite Server のインストールのインストールのための環境準備](#) を参照してください。
- **satellite-maintain backup** スクリプトを使用して作成した Satellite Server のバックアップ。Pulp データありでも、なしでもバックアップを使用できます。

- ターゲットサーバーの Satellite のサブスクリプション。

クローンを開始する前に、以下の条件を満たしていることを確認してください。

- ターゲットサーバーが分離ネットワーク上にあること。これにより、Capsule Server とホスト間で必要のない通信を回避できます。
- ターゲットサーバーのストレージ容量が、ソースサーバーと少なくとも同じである。

カスタマイズした設定ファイル

satellite-installer ツールまたは Satellite バックアッププロセスのマネージドでないソースサーバーで、設定をカスタマイズしている場合には、これらのファイルを手動でバックアップする必要があります。

3.3. PULP データの考慮事項

Pulp データを含めずに、Satellite Server のクローンを作成できます。ただし、クローンした環境を機能させるためには、Pulp データが必要です。ターゲットサーバーに、Pulp データがない場合には、Satellite は完全に機能しません。

Pulp データをターゲットサーバーに転送するには、2つのオプションがあります。

- Pulp データを含むバックアップを使用したクローン作成
- Pulp データなしのバックアップを使用してクローンを作成し、ソースサーバーから手動で **/var/lib/pulp** をコピーします。

pulp_data.tar ファイルが 500 GB 以上の場合や、**pulp_data.tar** ファイルが 100 GB 以上で、NFS など、速度の遅いストレージシステムを使用している場合には、デプロイメント時にメモリーエラーが発生する可能性があるため、バックアップに **pulp_data.tar** を含めないようにしてください。ソースサーバーからターゲットサーバーに **pulp_data.tar** ファイルをコピーするようにしてください。

Pulp データなしでバックアップする方法

「[Satellite Server のクローン](#)」の手順の内容に従います。ただし、Pulp データが含まれるクローンを作成する手順を、以下の手順に置き換えてください。

1. PostgreSQL データベースをアクティブにし、Pulp データは除外してバックアップを実行します。

```
# satellite-maintain backup offline --skip-pulp-content \
--assume-yes /var/backup
```

2. Satellite サービスを停止して無効にします。

```
# satellite-maintain service stop
# satellite-maintain service disable
```

3. Pulp データをターゲットサーバーにコピーします。

```
# rsync --archive --partial --progress --compress \
/var/lib/pulp/ target_server.example.com:/var/lib/pulp/
```

「[ターゲットサーバーのクローン作成](#)」に進みます。

3.4. SATELLITE SERVER のクローン

以下の手順を使用して、Satellite Server のクローンを作成します。この手順の一部として大量のデータをコピーして転送する必要があるため、完了までにかかり時間がかかる可能性がある点に注意してください。

3.4.1. クローン作成のためのソースサーバーの準備

ソースサーバーで、以下の手順を実行します。

1. Satellite サブスクリプションのプール ID を確認します。

```
# subscription-manager list --consumed \
--matches 'Red Hat Satellite'|grep "Pool ID:"|awk '{print $3}'
```

後で使用できるように、**プール ID** をメモしてください。

2. Red Hat Satellite のサブスクリプションを削除します。

```
# subscription-manager remove --serial=$(subscription-manager list \
--consumed \
--matches 'Red Hat Satellite'|grep "Serial:"|awk '{print $2}')
```

3. Pulp データのサイズを判断します。

```
# du -sh /var/lib/pulp/
```

4. Pulp データが 500 GB 未満の場合は、PostgreSQL のデータベースをアクティブにし、Pulp データを含めてバックアップを実行します。Pulp データが 500 GB 以上の場合には、以下の手順は省略して、続行する前に「[Pulp データの考慮事項](#)」の手順を実行してください。

```
# satellite-maintain backup offline --assumeyes /var/backup
```

5. Satellite サービスを停止して無効にします。

```
# satellite-maintain service stop
# satellite-maintain service disable
```

「[ターゲットサーバーのクローン作成](#)」に進みます。

3.4.2. ターゲットサーバーのクローン作成

サーバーのクローンを作成するには、ターゲットサーバーで以下の手順を実行してください。

1. **satellite-clone** ツールはデフォルトで、**/backup/** をバックアップフォルダーとして使用します。別のフォルダーにコピーする場合には、**/etc/satellite-clone/satellite-clone-vars.yml** ファイルの **backup_dir** 変数を更新してください。
2. 更新前の Satellite のバックアップファイルを、移行先のサーバーの **/backup/** フォルダーに配置します。共有ストレージをマウントするか、移行先のサーバーの **/backup/** フォルダーにバックアップファイルをコピーします。
3. ソースサーバーの電源を切断します。

- 以下のコマンドを入力してカスタマーポータルに登録して、サブスクリプションのアップロードし、必要なサブスクリプションだけを有効化します。

```
# subscription-manager register your_customer_portal_credentials
# subscription-manager attach --pool=pool_ID
# subscription-manager repos --disable=*
# subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=satellite-maintenance-6.14-for-rhel-8-x86_64-rpms
# dnf module enable satellite-maintenance:el8
```

- satellite-clone** パッケージをインストールします。

```
# dnf install satellite-clone
```

satellite-clone ツールをインストールした後に、独自のデプロイメントに合わせて `/etc/satellite-clone/satellite-clone-vars.yml` ファイルで設定を調節します。

- satellite-clone** ツールを実行します。

```
# satellite-clone
```

- DHCP、DNS、TFTP、およびリモート実行サービスを再設定します。ソースの Satellite Server と競合しないように、クローンプロセスにより、ターゲットの Satellite Server でこのサービスが無効になります。
- Satellite Web UI で DHCP、DNS、TFTP を再設定し、有効にします。詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の [Satellite Server での外部サービスの設定](#) を参照してください。
- ユーザー名 **admin** とパスワード **changeme** で、Satellite Web UI にログインします。すぐに、管理者パスワードを変更して認証情報のセキュリティを確保します。
- 正しい組織が選択されていることを確認します。
- Satellite Web UI で、**Content > Subscriptions** に移動します。
- マニフェストの管理** をクリックします。
- Refresh** をクリックしてから **Close** をクリックし、サブスクリプションのリストに戻ります。
- 利用可能なサブスクリプションが正しいことを確認します。
- `/usr/share/satellite-clone/logs/reassociate_capsules.txt` ファイルの説明に従い、Capsules とライフサイクル環境間の関係を復元します。
- ターゲットサーバーの IP アドレスと新規ホスト名が一致するように、DNS など、ネットワーク設定を更新します。**satellite-clone** ツールにより、ホスト名をソースサーバーのホスト名に変更します。ホスト名を別のものに変更する場合には、**satellite-change-hostname** ツールを使用してください。詳細は、Red Hat Satellite の管理 の Satellite または Capsule サーバーの名前の変更 を参照してください。
- ソースサーバーが **virt-who** デーモンを使用する場合は、ターゲットサーバーにインストールして設定します。ソースサーバーのディレクトリー `/etc/virt-who.d/` にある **virt-who** 設定ファイルをすべて、ターゲットサーバーにある同じディレクトリーに移動します。詳細は、Red Hat

Satellite での仮想マシンのサブスクリプションの設定を参照してください。以下の章を使用してアップグレードを行った後に、安全にソースサーバーの使用を停止することができます。

第4章 事前定義済みプロファイルを使用した SATELLITE SERVER の調整

Satellite のデプロイメントに 5000 台を超えるホストが含まれる場合には、事前定義済みの tuning プロファイルを使用して Satellite のパフォーマンスを向上できます。

Capsule では tuning プロファイルを使用できない点に注意してください。

Satellite が管理するホストの数と利用可能なハードウェアリソースに応じて、プロファイルの1つを選択できます。

tuning プロファイルは、`/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes` ディレクトリにあります。

`--tuning` オプションを指定して `satellite-installer` コマンドを実行した場合には、デプロイメント設定が以下の順番で Satellite Server に適用されます。

1. `/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` ファイルで定義したデフォルトの tuning プロファイル
2. `/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/` ディレクトリで定義され、デプロイメントに適用する tuning プロファイル
3. オプション: `/etc/foreman-installer/custom-hiera.yaml` ファイルを設定した場合、Satellite はこれらの設定を適用します。

`/etc/foreman-installer/custom-hiera.yaml` ファイルで定義した設定は、tuning プロファイルで定義した設定を上書きすることに注意してください。

したがって、tuning プロファイルを適用する前に、`/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml` のデフォルトの tuning プロファイルに定義されている設定、適用する tuning プロファイル、および `/etc/foreman-installer/custom-hiera.yaml` ファイルを比較して、重複する設定内容を `/etc/foreman-installer/custom-hiera.yaml` ファイルから削除する必要があります。

default

管理対象ホスト数: 0–5000

RAM: 20G

CPU コア数: 4

medium

管理対象ホスト数: 5001–10000

RAM: 32G

CPU コア数: 8

large

管理対象ホスト数: 10001–20000

RAM: 64G

CPU コア数: 16

extra-large

管理対象ホスト数: 20001–60000

RAM: 128G

CPU コア数: 32

extra-extra-large

マネージドホスト数: 60000+

RAM: 256G

CPU コア数: 48+

手順

1. オプション: Satellite Server で、**custom-hiera.yaml** ファイルを設定した場合、**/etc/foreman-installer/custom-hiera.yaml** ファイルを **custom-hiera.original** にバックアップします。**/etc/foreman-installer/custom-hiera.yaml** ファイルが破損した場合には、バックアップファイルを使用して、ファイルを元の状態に戻します。

```
# cp /etc/foreman-installer/custom-hiera.yaml \
/etc/foreman-installer/custom-hiera.original
```

2. オプション: Satellite Server で **custom-hiera.yaml** ファイルを設定した場合、**/usr/share/foreman-installer/config/foreman.hiera/tuning/common.yaml** のデフォルト tuning プロファイルの定義と、**/usr/share/foreman-installer/config/foreman.hiera/tuning/sizes/** に適用する tuning プロファイルを確認します。**/etc/foreman-installer/custom-hiera.yaml** ファイルの設定内容と比較して、**/etc/foreman-installer/custom-hiera.yaml** ファイルで重複設定を削除します。
3. 適用するプロファイルに対して、**--tuning** オプションを指定して **satellite-installer** コマンドを入力します。たとえば、medium tuning プロファイル設定を適用するには、以下のコマンドを入力します。

```
# satellite-installer --tuning medium
```


第5章 内部 SATELLITE データベースから外部データベースへの移行

Red Hat Satellite のインストールプロセスの一部として、**satellite-installer** コマンド、PostgreSQL のデータベースを Satellite と同じサーバー上にインストールします。デフォルトの内部データベースを使用している場合で、サーバーの負荷を軽減するために外部データベースを使い初める必要がある場合は、内部データベースを外部データベースに移行することが可能です。

Satellite Server のデータベースが内部または外部なのかを確認するには、データベースのステータスをクエリーすることができます。

PostgreSQL の場合は、以下のコマンドを実行します。

```
# satellite-maintain service status --only postgresql
```

Red Hat では、外部データベースのメンテナンスのサポートやそのためのツールは提供していません。これにはバックアップ、アップグレード、データベースのチューニングが含まれます。外部データベースをサポートし、管理する自社のデータベース管理者が必要です。

デフォルトの内部データベースから外部データベースに移行するには、以下の手順を完了する必要があります。

1. 「[外部データベース用のホストの準備](#)」 外部データベースをホストするように Red Hat Enterprise Linux 8 サーバーを準備します。
2. 「[PostgreSQL のインストール](#)」 Satellite、Pulp および Candlepin を所有する専用ユーザーを作成し、PostgreSQL を Satellite、Pulp および Candlepin のデータベースで準備します。
3. 「[外部データベースへの移行](#)」 新規データベースを参照するように **satellite-installer** のパラメーターを編集し、**satellite-installer** を実行します。

5.1. 外部データベースとして POSTGRESQL を使用する際の注意点

Foreman、Katello、および Candlepin は PostgreSQL データベースを使用します。PostgreSQL を外部データベースとして使用する場合は、以下の情報を参照してお使いの Satellite 設定にこのオプションが適しているかどうかを判別してください。Satellite は PostgreSQL バージョン 12 をサポートします。

外部 PostgreSQL の利点

- Satellite 上の空きメモリーと空き CPU が増えます。
- PostgreSQL データベースで **shared_buffers** を高い値に設定しても、Satellite 上の他のサービスの妨げるリスクがありません。
- Satellite 操作にマイナスの影響をもたらすことなく PostgreSQL サーバーのシステムを調整する柔軟性が得られます。

外部 PostgreSQL のマイナス点

- デプロイメントの複雑性が増し、問題解決がより困難になります。
- 外部 PostgreSQL サーバーの場合は、パッチおよびメンテナンス対象に新たなシステムが加わることになります。

- Satellite または PostgreSQL データベースサーバーのいずれかにハードウェアまたはストレージ障害が発生すると、Satellite が機能しなくなります。
- Satellite Server とデータベースサーバーの間でレイテンシーが発生すると、パフォーマンスに影響が出ます。

お使いの Satellite 上の PostgreSQL データベースが原因でパフォーマンスの低下が生じている可能性がある場合は、[Satellite 6: How to enable postgres query logging to detect slow running queries](#) を参照して時間のかかっているクエリーがあるかどうか判定します。1 秒以上かかるクエリーがある場合は、通常、大規模インストールのパフォーマンスが原因であることが多く、外部データベースに移行しても問題解決が期待できません。時間のかかっているクエリーがある場合は、Red Hat サポートチームまでお問い合わせください。

5.2. 外部データベース用のホストの準備

新しくプロビジョニングされたシステムに最新の Red Hat Enterprise Linux 8 をインストールして、外部データベースをホストします。

Red Hat Enterprise Linux のサブスクリプションでは、外部データベースと Satellite を併用する場合に、正しいサービスレベルアグリーメントが提供されません。外部データベースに使用するベースオペレーティングシステムにも、Satellite サブスクリプションをアタッチする必要があります。

前提条件

- 準備したホストが Satellite の [ストレージ要件](#) を満たしている。

手順

1. [Satellite Infrastructure サブスクリプションの割り当て](#) の手順に従って、Satellite サブスクリプションをサーバーに割り当てます。
2. すべてのリポジトリを無効にし、以下のリポジトリのみを有効にします。

```
# subscription-manager repos --disable '*'
# subscription-manager repos \
--enable=satellite-6.14-for-rhel-8-x86_64-rpms \
--enable=satellite-maintenance-6.14-for-rhel-8-x86_64-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms
```

3. 次のモジュールを有効にします。

```
# dnf module enable satellite:el8
```



注記

モジュール **satellite:el8** を有効にすると、**postgresql:10** および **ruby:2.5** との競合について警告が表示されます。これは、これらのモジュールが Red Hat Enterprise Linux 8 でデフォルトのモジュールバージョンに設定されているためです。モジュール **satellite:el8** には、モジュール **postgresql:12** および **ruby:2.7** への依存関係があり、**satellite:el8** モジュールで有効になります。これらの警告はインストールプロセスの失敗の原因にはならないため、安全に無視できます。Red Hat Enterprise Linux 8 のモジュールとライフサイクルストリームの詳細は、[Red Hat Enterprise Linux Application Streams のライフサイクル](#) を参照してください。

5.3. POSTGRESQL のインストール

インストール可能な PostgreSQL は、内部データベースのインストール中に **satellite-installer** ツールでインストールされたものと同じバージョンの PostgreSQL のみになります。Satellite は PostgreSQL バージョン 12 をサポートします。

手順

1. PostgreSQL をインストールするには、以下のコマンドを入力します。

```
# dnf install postgresql-server postgresql-evr postgresql-contrib
```

2. PostgreSQL を初期化するには、以下のコマンドを入力します。

```
# postgresql-setup initdb
```

3. **/var/lib/pgsql/data/postgresql.conf** ファイルで以下を行います。

```
# vi /var/lib/pgsql/data/postgresql.conf
```

Satellite で機能するには、外部 PostgreSQL のデフォルト設定を調整する必要があることに注意してください。基本的に推奨される外部データベース設定の調整は次のとおりです。

- checkpoint_completion_target: 0.9
- max_connections: 500
- shared_buffers: 512MB
- work_mem: 4MB

4. **#** を削除して、着信接続をリッスンするようにします。

```
listen_addresses = '*'
```

5. **/var/lib/pgsql/data/pg_hba.conf** ファイルを編集します。

```
# vi /var/lib/pgsql/data/pg_hba.conf
```

6. 以下の行をファイルに追加します。

```
host all all Satellite_ip/32 md5
```

7. PostgreSQL サービスを起動し、有効にするには、以下のコマンドを実行します。

```
# systemctl enable --now postgresql
```

8. 外部 PostgreSQL サーバーで **postgresql** ポートを開きます。

```
# firewall-cmd --add-service=postgresql
# firewall-cmd --runtime-to-permanent
```

9. **postgres** ユーザーに切り替え、PostgreSQL クライアントを起動します。

```
$ su - postgres -c psql
```

10. 3 つのデータベースと専用のロールを作成します。1 つは Satellite 用、1 つは Candlepin 用、もう 1 つは Pulp 用です。

```
CREATE USER "foreman" WITH PASSWORD 'Foreman_Password';
CREATE USER "candlepin" WITH PASSWORD 'Candlepin_Password';
CREATE USER "pulp" WITH PASSWORD 'Pulpcore_Password';
CREATE DATABASE foreman OWNER foreman;
CREATE DATABASE candlepin OWNER candlepin;
CREATE DATABASE pulpcore OWNER pulp;
```

11. Pulp データベースに接続します。

```
postgres=# \c pulpcore
You are now connected to database "pulpcore" as user "postgres".
```

12. **hstore** エクステンションを作成します。

```
pulpcore=# CREATE EXTENSION IF NOT EXISTS "hstore";
CREATE EXTENSION
```

13. **postgres** ユーザーをログアウトします。

```
# \q
```

14. Satellite Server から、データベースにアクセスできることをテストします。接続に成功した場合には、コマンドは **1** を返します。

```
# PGPASSWORD='Foreman_Password' psql -h postgres.example.com -p 5432 -U
foreman -d foreman -c "SELECT 1 as ping"
# PGPASSWORD='Candlepin_Password' psql -h postgres.example.com -p 5432 -U
candlepin -d candlepin -c "SELECT 1 as ping"
# PGPASSWORD='Pulpcore_Password' psql -h postgres.example.com -p 5432 -U pulp -
d pulpcore -c "SELECT 1 as ping"
```

5.4. 外部データベースへの移行

既存のデータをバックアップおよび転送してから、**satellite-installer** コマンドを使用して、外部の PostgreSQL データベースに接続するように Satellite を設定します。

前提条件

- Red Hat Enterprise Linux サーバーに PostgreSQL サーバーをインストールおよび設定していること。

手順

- Satellite Server で Satellite サービスを停止します。

```
# satellite-maintain service stop
```

- PostgreSQL サービスを起動します。

```
# systemctl start postgresql
```

- 内部データベースのバックアップを作成します。

```
# satellite-maintain backup online \
--preserve-directory \
--skip-pulp-content \
/var/migration_backup
```

- データを新規外部データベースに転送します。

```
PGPASSWORD='Foreman_Password' pg_restore -h postgres.example.com -U foreman -
d foreman < /var/migration_backup/foreman.dump
PGPASSWORD='Candlepin_Password' pg_restore -h postgres.example.com -U
candlepin -d candlepin < /var/migration_backup/candlepin.dump
PGPASSWORD='Pulpcore_Password' pg_restore -h postgres.example.com -U pulp -d
pulpcore < /var/migration_backup/pulpcore.dump
```

- satellite-installer** コマンドを使用して Satellite が新規データベースを参照するように更新します。

```
# satellite-installer --scenario satellite \
--foreman-db-database foreman \
--foreman-db-host postgres.example.com \
--foreman-db-manage false \
--foreman-db-password Foreman_Password \
--foreman-db-username foreman \
--foreman-proxy-content-pulpcore-manage-postgresql false \
--foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \
--foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \
--foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password \
--foreman-proxy-content-pulpcore-postgresql-user pulp \
--katello-candlepin-db-host postgres.example.com \
--katello-candlepin-db-name candlepin \
--katello-candlepin-db-password Candlepin_Password \
--katello-candlepin-db-user candlepin \
--katello-candlepin-manage-db false
```

第6章 ANSIBLE COLLECTIONS を使用した SATELLITE の管理

Satellite Ansible Collections は、Satellite API と対話する Ansible モジュールセットです。Satellite Ansible Collections を使用して、Satellite の多くの側面を管理および自動化することができます。

6.1. SATELLITE ANSIBLE モジュールのインストール

この手順を使用して、Satellite Ansible モジュールをインストールします。

手順

- 次のコマンドを使用してパッケージをインストールします。

```
# satellite-maintain packages install ansible-collection-redhat-satellite
```

6.2. SATELLITE ANSIBLE モジュールの表示

以下を実行すると、インストールされている Satellite Ansible モジュールを表示できます。

```
# ansible-doc -l redhat.satellite
```

または、[Red Hat Ansible Automation Platform](#) で Satellite Ansible モジュールの完全なリストとその他の関連情報を確認することもできます。

すべてのモジュールは **redhat.satellite** namespace にあり、**redhat.satellite._module_name_** という形式で参照できます。たとえば、**activation_key** モジュールに関する情報を表示するには、以下のコマンドを入力します。

```
$ ansible-doc redhat.satellite.activation_key
```

第7章 組織の管理

組織は、所有者、目的、コンテンツ、セキュリティーレベルなどの区分に基づいて Red Hat Satellite リソースを論理グループに分割します。Red Hat Satellite で複数の組織を作成して管理し、Red Hat サブスクリプションを分割して各組織に割り当てることができます。これにより、1つの管理システムで複数の個別の組織のコンテンツを管理できるようになります。以下に、組織管理の例をいくつか示します。

単一の組織

システム管理チェーンが単純な小企業。この場合は、ビジネスに組織を1つ作成し、コンテンツをその組織に割り当てることができます。

複数の組織

複数の小規模な事業単位を所有する大規模な企業。たとえば、別のシステム管理とソフトウェア開発グループを持つ会社です。この場合、所属する会社および各企業が所有する各事業単位の組織を作成できます。これにより、システムインフラストラクチャーが別々に維持されます。こうすることで、ニーズに合わせて各組織にコンテンツを割り当てることができます。

外部組織

他の組織の外部システムを管理する会社たとえば、クラウドコンピューティングと Web ホストリソースを顧客に提供する会社です。この場合は、会社の独自のシステムインフラストラクチャーの組織と、外部の各ビジネスに組織を作成できます。必要に応じて、各組織にコンテンツを割り当てることができます。

Red Hat Satellite のデフォルトのインストールには、**Default Organization** と呼ばれるデフォルトの組織があります。

新規ユーザー

新規ユーザーにデフォルトの組織が割り当てられていない場合には、そのアクセスは制限されます。ユーザーにシステム権限を付与するには、ユーザーをデフォルトの組織に割り当てます。ユーザーが次に Satellite にログインする際に、ユーザーのアカウントに適切なシステム権限が付与されます。

7.1. 組織の作成

以下の手順を使用して組織を作成します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#)を参照してください。

手順

1. Satellite Web UI で、**Administer > Organizations** に移動します。
2. **新規組織** をクリックします。
3. **名前** フィールドに、組織の名前を入力します。
4. **ラベル** フィールドには、組織の一意識別子を入力します。これは、コンテンツストレージのディレクトリーなど、特定のアセットを作成し、マッピングするために使用されます。文字、数字、アンダースコア、およびダッシュを使用し、スペースは使用しないでください。
5. オプション: Simple Content Access (SCA) を有効にしない場合は、**Simple Content Access** チェックボックスをオフにします。SCA の詳細は、[Simple Content Access](#) を参照してください。



注記

Red Hat は、エンタイトルメントモードが非推奨であるため、SCA を無効にすることを推奨しません。

6. オプション: **Description** フィールドに、組織の説明を入力します。
7. **Submit** をクリックします。
8. ホストに組織が割り当てられていない場合は、組織に追加するホストを選択し、**Proceed to Edit** をクリックします。
9. **編集** ページで、組織に追加するインフラストラクチャーリソースを割り当てます。これには、ネットワークリソース、インストールメディア、キックスタートテンプレートなどのパラメーターが含まれます。**管理 > 組織** に移動して、編集する組織を選択すると、いつでもこのページに戻ることができます。
10. **Submit** をクリックします。

CLI 手順

1. 組織を作成するには、以下のコマンドを入力します。

```
# hammer organization create \
--name "My_Organization" \
--label "My_Organization_Label" \
--description "My_Organization_Description"
```



注記

この方法で作成された組織では、デフォルトで Simple Content Access (SCA) が有効になっています。SCA を無効にする場合は、コマンドに **--simple-content-access false** パラメーターを追加します。エンタイトルメントモード (SCA を使用しない) は非推奨であるため、Red Hat は SCA を無効にすることを推奨していません。

2. オプション: 組織を編集するには、**hammer organization update** コマンドを入力します。たとえば、以下のコマンドはコンピュータリソースを組織に割り当てます。

```
# hammer organization update \
--name "My_Organization" \
--compute-resource-ids 1
```

7.2. 組織コンテキストの設定

ホストに使用する組織や関連するリソースを定義する組織のコンテキスト

手順

組織メニューは、Satellite Web UI の左上にあるメニューバーの最初のメニュー項目です。現在の組織を選択していない場合には、メニューには**任意の組織**が表示されます。**Any Organization** をクリックして、組織を選択します。

CLI 手順

CLI を使用する場合は、**--organization "My_Organization"** または **--organization-label "My_Organization_Label"** オプションのいずれかを追加します。以下に例を示します。

```
# hammer subscription list \  
--organization "My_Organization"
```

このコマンドは、**My_Organization** に割り当てられたサブスクリプションを出力します。

7.3. 組織デバッグ証明書の作成

組織のデバッグ証明書が必要な場合は、以下の手順を使用します。

手順

1. Satellite Web UI で、**Administer > Organizations** に移動します。
2. デバッグ証明書を生成する組織を選択します。
3. **Generate and Download** をクリックします。
4. 証明書ファイルを安全なロケーションに保存します。

プロビジョニングテンプレート用の証明書のデバッグ

デバッグ証明書は、テンプレートダウンロードがダウンロードされている組織に存在しない場合には、テンプレートのダウンロードのプロビジョニング用に自動的に生成されます。

7.4. 組織デバッグ証明書を使用したリポジトリコンテンツの閲覧

Web ブラウザーを使用して、またはその組織のデバッグ証明書がある場合は、API を使用して組織のリポジトリコンテンツを表示できます。

前提条件

- 組織の証明書を作成してダウンロードしました。詳細は、[「組織デバッグ証明書の作成」](#) を参照してください。

手順

1. 証明書の秘密鍵と公開鍵を2つのファイルに分割します。
 - a. たとえば、デフォルトの組織の X.509 証明書を開きます。

```
$ vi 'Default Organization-key-cert.pem'
```
 - b. **-----BEGIN RSA PRIVATE KEY-----** から **-----END RSA PRIVATE KEY-----** までのファイルのコンテンツを **key.pem** ファイルにコピーします。
 - c. **-----BEGIN CERTIFICATE-----** から **-----END CERTIFICATE-----** までのファイルのコンテンツを **cert.pem** ファイルにコピーします。
2. ブラウザーを使用するには、まず X.509 証明書をブラウザーがサポートする形式に変換してから、証明書をインポートする必要があります。

Firefox をご利用の場合

1. 次のコマンドを使用して、証明書を PKCS12 形式に変換します。

```
$ openssl pkcs12 -keypbe PBE-SHA1-3DES -certpbe PBE-SHA1-3DES -export -in cert.pem
-inkey key.pem -out My_Organization_Label.pfx -name My_Organization
```

2. Firefox ブラウザーで、**編集 > 設定 > 詳細タブ** に移動します。
3. **証明書の表示** を選択し、**証明書タブ** をクリックします。
4. **Import** をクリックして、読み込む **.pfx** ファイルを選択します。
5. アドレスバーに次の URL を入力して、すべてのリポジトリのアクセス可能なパスを参照し、その内容を確認します。

```
https://satellite.example.com/pulp/content/
```

CURL ユーザーの場合

- CURL で組織のデバッグ証明書を使用するには、以下のコマンドを入力します。

```
$ curl -k --cert cert.pem --key key.pem \
https://satellite.example.com/pulp/content/My_Organization_Label/Library/content/dist/rhel/
server/7/7Server/x86_64/os/
```

cert.pem および **key.pem** へのパスが正しい絶対パスであることを確認します。そうしないと、コマンドは警告なしで失敗します。Pulp は組織ラベルを使用するため、組織ラベルを URL に入力する必要があります。

7.5. 組織の削除

組織は、ライフサイクル環境またはホストグループに関連付けられていない場合に削除できます。削除予定の組織にライフサイクル環境またはホストグループが関連付けられている場合は、**Administer > Organizations** に移動して関連の組織をクリックします。インストール中に作成されたデフォルトの組織は、Satellite 環境で関連付けられていないホストへのプレースホルダーであるため、削除しないでください。環境には常に 1 つ以上の組織が必要です。

手順

1. Satellite Web UI で、**Administer > Organizations** に移動します。
2. 削除する組織名の右側にあるリストから **削除** を選択します。
3. **OK** をクリックして組織を削除します。

CLI 手順

1. 以下のコマンドを入力して、削除する組織の ID を取得します。

```
# hammer organization list
```

出力から、削除する組織の ID をメモします。

2. 以下のコマンドを入力して組織を削除します。

```
# hammer organization delete --id Organization_ID
```

第8章 ロケーションの管理

ロケーションは、リソースをグループ化してホストを割り当てる方法を提供する組織と同様に機能します。組織とロケーションには、以下の概念の違いがあります。

- ロケーションは、物理または地理的な設定に基づいています。
- ロケーションには階層構造があります。

8.1. ロケーションの作成

以下の手順を使用して、ホストとリソースをロケーション別に管理できるようにロケーションを作成します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

手順

1. Satellite Web UI で、**Administer > Locations** に移動します。
2. **New Location** をクリックします。
3. オプション: **Parent** リストから、親ロケーションを選択します。これにより、ロケーションの階層が作成されます。
4. **Name** フィールドに、ロケーションの名前を入力します。
5. オプション: **Description** フィールドに、ロケーションの説明を入力します。
6. **Submit** をクリックします。
7. ホストにロケーションが割り当てられていない場合は、新しいロケーションに割り当てるホストを追加し、**Proceed to Edit** をクリックします。
8. ロケーションに追加するインフラストラクチャリソースを割り当てます。これには、ネットワークリソース、インストールメディア、キックスタートテンプレートなどのパラメーターが含まれます。**Administer > Locations** に移動して、編集するロケーションを選択すると、いつでもこのページに戻ることができます。
9. **Submit** をクリックして変更を保存します。

CLI 手順

- 以下のコマンドを実行してロケーションを作成します。

```
# hammer location create \  
--description "My_Location_Description" \  
--name "My_Location" \  
--parent-id "My_Location_Parent_ID"
```

8.2. 複数のロケーションの作成

以下の Bash スクリプトでは、3つのロケーション (ロンドン、ミュンヘン、ボストン) を作成して、これらのロケーションを Example Organization に割り当てます。

```
ORG="Example Organization"
```

```
LOCATIONS="London Munich Boston"
```

```
for LOC in ${LOCATIONS}
do
  hammer location create --name "${LOC}"
  hammer location add-organization --name "${LOC}" --organization "${ORG}"
done
```

8.3. ロケーションコンテキストの設定

ホストに使用するロケーションや関連するリソースを定義するロケーションのコンテキスト

手順

ロケーションメニューは、Satellite Web UI の左上にあるメニューバーの 2 番目のメニュー項目です。現在のロケーションを選択していない場合には、メニューには **任意のロケーション** と表示されます。**任意のロケーション** をクリックして、使用するロケーションを選択します。

CLI 手順

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

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

このコマンドは、**My_Location** のロケーションに関連付けられたホストを一覧表示します。

8.4. ロケーションの削除

ロケーションは、ライフサイクル環境またはホストグループに関連付けられていない場合に削除できます。削除するロケーションにライフサイクル環境またはホストグループが関連付けられている場合は、**Administer > Locations** に移動して、関連するロケーションをクリックします。インストール中に作成されたデフォルトのロケーションは、Satellite 環境で関連付けられていないホストへのプレースホルダーであるため、削除しないでください。環境には常に 1 つ以上のロケーションが必要です。

手順

1. Satellite Web UI で、**Administer > Locations** に移動します。
2. 削除するロケーションの名前の右側にあるリストから**削除**を選択します。
3. **OK** をクリックして、ロケーションを削除します。

CLI 手順

1. 以下のコマンドを入力して、削除するロケーションの ID を取得します。

```
# hammer location list
```

出力から、削除するロケーションの ID をメモします。

2. 以下のコマンドを実行してロケーションを削除します。

```
# hammer location delete --id Location ID
```

第9章 ユーザーとロールの管理

ユーザーは、システムを使用する各個人の一連の詳細情報を定義します。組織と環境をユーザーに関連付けることで、新しいエンティティを作成する際にこれらのデフォルト値を自動的に使用することができます。また、ユーザーには1つ以上のロールを割り当てることもでき、ユーザーには組織と環境を参照および管理する権限が与えられます。ユーザーの使用に関する詳細は、[「ユーザーの管理」](#)を参照してください。

複数のユーザーのパーミッションは、ユーザーグループでまとめることにより一括して管理できます。また、ユーザーグループ自体をさらにグループ化してパーミッションの階層を作成できます。ユーザーグループの作成の詳細は、[「ユーザーグループの作成と管理」](#)を参照してください。

ロールでは、一連のパーミッションおよびアクセスレベルを定義します。各ロールには、ロールに許可されたアクションを指定する1つ以上のパーミッションフィルターが含まれます。アクションは、**リソースタイプ**に従ってグループ化されます。ロールが作成されたら、そのロールにはユーザーとユーザーグループを関連付けることができます。この場合は、ユーザーの大きなグループに同じ一連のパーミッションセットを割り当てることができます。Satellite では、事前定義された一連のロールが提供され、[「ロールの作成および管理」](#)で説明されているようにカスタムロールおよびパーミッションフィルターを作成することもできます。

9.1. ユーザーの管理

管理者は、Satellite ユーザーを作成、変更、および削除できます。また、異なる **ロール** をユーザーやユーザーのグループに割り当てて、アクセスパーミッションを設定することもできます。

9.1.1. ユーザーの作成

この手順を使用してユーザーを作成します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#)を参照してください。

手順

1. Satellite Web UI で、**Administer > Users** に移動します。
2. **Create User** をクリックします。
3. **ログイン** フィールドにユーザーのユーザー名を入力します。
4. **名** および **姓** フィールドに、ユーザーの実名を入力します。
5. **Email アドレス** フィールドにメールアドレスを入力します。
6. **説明** フィールドには、新規ユーザーの説明を加えます。
7. **言語** リストからユーザー用の言語を選択します。
8. **タイムゾーン** リストからタイムゾーンを選択します。
デフォルトでは、Satellite Server はユーザーのブラウザーの言語とタイムゾーンを使用します。
9. ユーザーのパスワードを設定します。
 - a. **認証先** リストから、ユーザー認証に使用するソースを選択します。
 - **内部**: Satellite Server 内でのユーザー管理を有効にします。

- **EXTERNAL:** オンラインネットワーク環境からの Satellite Server のインストールの [外部認証の設定](#) で説明されているように、外部認証を設定します。

b. パスワード フィールドに初回パスワードを入力して、**確認** フィールドで再入力します。

10. **送信** をクリックしてユーザーを作成します。

CLI 手順

- 以下のコマンドを実行してユーザーを作成します。

```
# hammer user create \
--auth-source-id My_Authentication_Source \
--login My_User_Name \
--mail My_User_Mail \
--organization-ids My_Organization_ID_1,My_Organization_ID_2 \
--password My_User_Password
```

--auth-source-id 1 の設定では、ユーザーは内部で認証されますが、外部認証を指定することもできます。**--admin** オプションを追加して、管理者権限をユーザーに付与します。組織 ID を指定する必要はありません。**update** サブコマンドを使用してユーザーの詳細を変更できます。

ユーザー関連のサブコマンドに関する情報は、**hammer user --help** の入力してください。

9.1.2. ユーザーへのロールの割り当て

この手順を使用して、ロールをユーザーに割り当てます。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

手順

1. Satellite Web UI で、**Administer > Users** に移動します。
2. ロールを割り当てるユーザーの **ユーザー名** をクリックします。



注記

ユーザーアカウントが表示されない場合は、現在適切な組織を表示しているかどうかを確認します。Satellite の全ユーザーをリスト表示するには、**デフォルトの組織** をクリックしてから **任意の組織** をクリックします。

3. **ロケーション** タブをクリックして、ロケーションが割り当てられていない場合は選択します。
4. **組織** タブをクリックして、組織が割り当てられていることを確認します。
5. **ロール** タブをクリックして利用可能なロールのリストを表示します。
6. **ロール** リストから割り当てるロールを選択します。
利用可能な全パーミッションを付与するには、**管理** チェックボックスを選択します。
7. **Submit** をクリックします。

ユーザーに割り当てられたロールを参照するには、**ロール** タブをクリックします。割り当てられたロールは、**選択された項目** に表示されます。割り当てたロールを削除するには、**選択された項目** でロール名をクリックします。

CLI 手順

- ユーザーにロールを割り当てるには、次のコマンドを入力します。

```
# hammer user add-role --id user_id --role role_name
```

9.1.3. 別のユーザーアカウントへの切り替え

管理者は、別のユーザーとして Satellite Web UI に一時的にログオンすることにより、テストおよびトラブルシューティングの目的で他の認証済みユーザーに切り替えることができます。別のユーザーに切り替える場合、管理者は、同じメニューを含め、切り替えたユーザーとまったく同じシステム内のアクセス権限を持ちます。

監査は、管理者が別のユーザーとして実行するアクションを記録するために作成されます。ただし、管理者が別のユーザーとして実行するすべてのアクションは、切り替え後のユーザーで実行されたものとして記録されます。

前提条件

- Satellite の管理者権限を持つユーザーとして Satellite Web UI にログオンしていること。

手順

1. Satellite Web UI で、**Administer > Users** に移動します。
2. 切り替えるユーザーの右側の **アクション** コラムのリストから、**ユーザー切り替え** を選択します。

切り替えセッションを停止するには、メインメニューの右上にある切り替えアイコンをクリックします。

9.1.4. API のみのユーザーの作成

Satellite API とのみ対話できるユーザーを作成できます。

前提条件

- ユーザーを作成し、それらにロールを割り当てました。このユーザーは内部で承認されている必要があることに注意してください。詳細は、[xref:Assigning_Roles_to_a_User_admin](#) を参照してください。

手順

1. 管理者として Satellite にログインします。
2. **Administer > Users** に移動して、ユーザーを選択します。
3. **User** タブで、パスワードを設定します。このパスワードを保存したり、他の人と通信したりしないでください。コンソールで疑似ランダム文字列を作成できます。

```
# openssl rand -hex 32
```

4. ユーザーの Personal Access Token を作成します。詳細は、[「Personal Access Token の作成」](#) を参照してください。

9.2. SSH キーの管理

ユーザーに SSH キーを追加すると、プロビジョニング中に SSH キーのデプロイメントが可能になります。プロビジョニング中に SSH キーをデプロイする方法の詳細は、[ホストのプロビジョニングの プロビジョニング中の SSH キーのデプロイ](#) を参照してください。

SSH キーおよび SSH キーの作成に関する情報は、[Red Hat Enterprise Linux 8 基本的なシステム設定の設定の SSH ベースの認証の使用](#) を参照してください。

9.2.1. ユーザーの SSH キー管理

この手順を使用して、ユーザーの SSH キーを追加または削除します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

前提条件

- Red Hat Satellite 管理ユーザーとして Satellite Web UI にログインするか、SSH キーの追加には `create_ssh_key` パーミッションを有効にしたユーザーとして、キーの削除には `destroy_ssh_key` パーミッションを有効にしたユーザーとしてログインすること。

手順

- Satellite Web UI で、**Administer** > **Users** に移動します。
- ユーザー名** コラムから必要となるユーザーのユーザー名をクリックします。
- SSH キー** タブをクリックします。
 - SSH キーの追加
 - 公開 SSH キーのコンテンツをクリップボードに用意します。
 - SSH キーの追加** をクリックします。
 - キー** フィールドに公開 SSH キーのコンテンツをクリップボードから貼り付けます。
 - 名前** フィールドに SSH キーの名前を入力します。
 - Submit** をクリックします。
 - SSH キーの削除
 - 削除する SSH キーの列にある **削除** をクリックします。
 - 確認プロンプトで **OK** をクリックします。

CLI 手順

SSH キーをユーザーに追加するには、公開 SSH キーファイルへのパスを指定するか、クリップボードにコピーする公開 SSH キーのコンテンツへのパスが必要です。

- 公開 SSH キーファイルがある場合は、次のコマンドを入力します。

```
# hammer user ssh-keys add \  
--user-id user_id \  
--name key_name \  
--key-file ~/.ssh/id_rsa.pub
```

- SSH 公開キーのコンテンツがある場合は、次のコマンドを入力します。

```
# hammer user ssh-keys add \
--user-id user_id \
--name key_name \
--key ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTltbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHHS2KmNylYa2
7Qaa7EHp+2l99ucGStx4P77e03ZvE3yVRJEFikpoP3MJtYYfle8k
1/46MTIZo9CPTX4CYUHeN8= host@user
```

ユーザーの SSH キーを削除するには、次のコマンドを入力します。

```
# hammer user ssh-keys delete --id key_id --user-id user_id
```

ユーザーにアタッチされた SSH キーを表示するには、次のコマンドを入力します。

```
# hammer user ssh-keys info --id key_id --user-id user_id
```

ユーザーにアタッチされた SSH キーを表示するには、以下のコマンドを入力します。

```
# hammer user ssh-keys list --user-id user_id
```

9.3. PERSONAL ACCESS TOKEN の管理

Personal Access Token を使用すると、パスワードを使用せずに、API リクエストを認証できます。Personal Access Token の有効期限を設定できます。有効期限が切れる前に、有効期限が切れると判断した場合は、トークンを取り消すことができます。

9.3.1. Personal Access Token の作成

この手順を使用して、Personal Access Token を作成します。

手順

1. Satellite Web UI で、**Administer > Users** に移動します。
2. Personal Access Token を作成するユーザーを選択します。
3. **Personal Access Tokens** タブで、**Add Personal Access Token** をクリックします。
4. Personal Access Token の **Name** を入力します。
5. オプション: **Expires** 日付を選択して、有効期限を設定します。有効期限を設定しない場合、Personal Access Token は、取り消されないかぎり、期限切れになりません。
6. **Submit** をクリックします。**Personal Access Tokens** タブで、Personal Access Token を使用できるようになりました。



重要

ページを離れるか、新しい Personal Access Token を作成すると、再びアクセスできなくなるため、Personal Access Token を必ず保存してください。**Copy to clipboard** をクリックして、Personal Access Token をコピーできます。

検証

1. Satellite Server に API リクエストを送信し、Personal Access Token で認証します。

```
# curl https://satellite.example.com/api/status --user  
My_Username:My_Personal_Access_Token
```

2. ステータス **200** のレスポンスを受け取る必要があります。次に例を示します。

```
{"satellite_version":"6.14.0","result":"ok","status":200,"version":"3.5.1.10","api_version":2}
```

Personal Access Tokens タブに戻ると、Personal Access Token の横に更新された **Last Used** 時刻が表示されます。

9.3.2. Personal Access Token の取り消し

この手順を使用して、有効期限が切れる前に、Personal Access Token を取り消します。

手順

1. Satellite Web UI で、**Administer** > **Users** に移動します。
2. Personal Access Token を取り消すユーザーを選択します。
3. **Personal Access Tokens** タブで、取り消す Personal Access Token を見つけます。
4. 取り消す Personal Access Token の横にある **Actions** 列の **取り消し** をクリックします。

検証

1. Satellite Server に API リクエストを送信し、失効した Personal Access Token で認証を試みます。

```
# curl https://satellite.example.com/api/status --user  
My_Username:My_Personal_Access_Token
```

2. 次のエラーメッセージが表示されます。

```
{  
  "error": {"message": "Unable to authenticate user My_Username"}  
}
```

9.4. ユーザーグループの作成と管理

9.4.1. ユーザーグループ

Satellite を使用すると、ユーザーのグループに権限を割り当てることができます。また、ユーザーグループを他のユーザーグループのコレクションとして作成することもできます。外部認証ソースを使用している場合は、[オンラインネットワーク環境での Satellite Server のインストールの 外部ユーザーグループの設定](#) で説明されているように、Satellite ユーザーグループを外部ユーザーグループにマップできます。

ユーザーグループは組織コンテキストで定義されます。したがって、ユーザーグループにアクセスする前に組織を選択する必要があります。

9.4.2. ユーザーグループの作成

以下の手順を使用してユーザーグループを作成します。

手順

1. Satellite Web UI で、**Administer > User Groups** に移動します。
2. **ユーザーグループの作成** をクリックします。
3. **ユーザーグループ** タブで、新規ユーザーグループの名前を指定し、グループメンバーを選択します。
 - **ユーザーグループ** のリストから、以前に作成したユーザーグループを選択します。
 - **ユーザー** のリストからユーザーを選択します。
4. **ロール** タブで、ユーザーグループに割り当てるロールを選択します。または、**管理者** チェックボックスを選択して利用可能なすべてのパーミッションを割り当てます。
5. **Submit** をクリックします。

CLI 手順

- ユーザーグループを作成するには、次のコマンドを入力します。

```
# hammer user-group create \  
--name My_User_Group_Name \  
--role-ids My_Role_ID_1,My_Role_ID_2 \  
--user-ids My_User_ID_1,My_User_ID_2
```

9.4.3. ユーザーグループの削除

以下の手順を使用して、Satellite からユーザーグループを削除します。

手順

1. Satellite Web UI で、**Administer > User Groups** に移動します。
2. 削除するユーザーグループの右側にある **Delete** をクリックします。
3. **Confirm** をクリックしてユーザーグループを削除します。

9.5. ロールの作成および管理

Satellite では、標準的なタスクに十分なパーミッションとなる事前定義済みローラー式が提供されます (「[Satellite で利用可能な事前定義済みロール](#)」を参照)。また、カスタムロールを設定し、このカスタムロールに1つ以上のパーミッションフィルターを割り当てることもできます。パーミッションフィルターでは、特定のリソースタイプに許可されるアクションを定義します。特定の Satellite プラグインによりロールが自動的に作成されます。

9.5.1. ロールの作成

以下の手順を使用してロールを作成します。

手順

1. Satellite Web UI で、**Administer** > **Roles** に移動します。
2. **Create Role** をクリックします。
3. ロールの **名前** を記入します。
4. **送信** をクリックして、新しいロールを保存します。

CLI 手順

- 以下のコマンドを実行してロールを作成します。

```
# hammer role create --name My_Role_Name
```

ロールにはパーミッションを含める必要があります。ロールの作成後は、[「ロールへのパーミッションの追加」](#)に進んでください。

9.5.2. ロールのクローン作成

Satellite Web UI を使用してロールのクローンを作成します。

手順

1. Satellite web UI で、**Administer** > **Roles** に移動し、必要なロールの右側にあるドロップダウンメニューから **Clone** を選択します。
2. ロールの **名前** を記入します。
3. **送信** をクリックしてロールのクローンを作成します。
4. クローンされたロールの名前をクリックし、**フィルター** に移動します。
5. 必要に応じて、パーミッションを編集します。
6. **送信** をクリックして、新しいロールを保存します。

9.5.3. ロールへのパーミッションの追加

以下の手順を使用して、ロールにパーミッションを追加します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#)を参照してください。

手順

1. Satellite Web UI で、**Administer** > **Roles** に移動します。
2. 必要なロールの右側にあるドロップダウンリストから **フィルターの追加** を選択します。
3. ドロップダウンリストから **リソースタイプ** を選択します。**(その他)** グループには、どのリソースグループにも関連付けられていないパーミッションが含まれます。

4. 選択するパーミッションを **パーミッション** リストからクリックします。
5. **リソースタイプ** での選択により、**無制限** と **上書き** のチェックボックスが表示されます。**無制限** チェックボックスはデフォルトで選択され、選択されたタイプの全リソースにパーミッションが適用されます。**無制限** チェックボックスを無効にすると、**検索** フィールドが有効になります。このフィールドでは、Satellite 検索構文を使用してさらにフィルタリングを指定できます。詳細は、「[詳細なパーミッションフィルタリング](#)」を参照してください。**上書き** チェックボックスを有効にすると、新たなロケーションと組織を追加して、それらのロケーションや組織のリソースタイプにこのロールがアクセスできるようになります。また、すでに関連付けられたロケーションや組織をリソースタイプから削除して、アクセスを制限することもできます。
6. **Next** をクリックします。
7. **Submit** をクリックして変更を保存します。

CLI 手順

1. 利用可能な全パーミッションを表示します。

```
# hammer filter available-permissions
```

2. ロールにパーミッションを追加します。

```
# hammer filter create \
--permission-ids My_Permission_ID_1,My_Permission_ID_2 \
--role My_Role_Name
```

ロールとパーミッションパラメーターの詳細は、**hammer role --help** および **hammer filter --help** コマンドを入力します。

9.5.4. ロールのパーミッションの表示

Satellite Web UI を使用してロールのパーミッションを表示します。

手順

1. Satellite Web UI で、**Administer > Roles** に移動します。
2. 必要なロールの右側にある **フィルター** をクリックして、**フィルター** ページを開きます。

フィルター ページでは、リソースタイプ別にグループ化されたロールに割り当てられたパーミッションの表が表示されます。また、このページでは、Satellite システムで使用できるパーミッションとアクションの完全な表を生成できます。詳細は、「[パーミッションの完全テーブルの作成](#)」を参照してください。

9.5.5. パーミッションの完全テーブルの作成

Satellite CLI を使用してパーミッションテーブルを作成します。

手順

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

```
# satellite-maintain packages install foreman-console
```

2. 以下のコマンドで Satellite コンソールを起動します。

```
# foreman-rake console
```

コンソールに以下のコードを挿入します。

```
f = File.open('/tmp/table.html', 'w')

result = Foreman::AccessControl.permissions {|a,b| a.security_block <=>
  b.security_block}.collect do |p|
  actions = p.actions.collect { |a| "<li>#{a}</li>" }
  "<tr><td>#{p.name}</td><td><ul>#{actions.join("")}</ul></td><td>#{p.resource_type}</td>
</tr>"
end.join("\n")

f.write(result)
```

上記の構文により、パーミッションの表が作成され、**/tmp/table.html** ファイルに保存されます。

3. **Ctrl + D** を押して、Satellite コンソールを終了します。 **/tmp/table.html** の最初の行に以下のテキストを挿入します。

```
<table border="1"><tr><td>Permission name</td><td>Actions</td><td>Resource type</td>
</tr>
```

/tmp/table.html の最後に以下のテキストを追加します。

```
</table>
```

4. Web ブラウザーで **/tmp/table.html** を開いて、表を確認します。

9.5.6. ロールの削除

以下の手順を使用して、Satellite からロールを削除します。

手順

1. Satellite Web UI で、**Administer > Roles** に移動します。
2. 削除するロールの右側にあるドロップダウンリストから **Delete** を選択します。
3. **Confirm** をクリックしてロールを削除します。

9.6. SATELLITE で利用可能な事前定義済みロール

以下の表は、Satellite で事前定義されたロールがユーザーに付与するパーミッションの概要を示しています。

事前定義されたロールとそれらが付与するパーミッションの完全なセットを確認するには、Satellite Web UI に特権ユーザーとしてログインし、**Administer > Roles** に移動します。詳細は、「[ロールのパーミッションの表示](#)」を参照してください。

事前定義されたロール	ロールが提供するパーミッション	関連情報
Auditor	監査ログを表示します。	
デフォルトロール	タスクとジョブの呼び出しを表示します。	Satellite は、このロールをシステム内のすべてのユーザーに自動的に割り当てます。
Manager	グローバル設定を表示および編集します。	
Organization admin	組織を管理するパーミッションを除くすべてのパーミッション。	<p>組織ごとに定義された管理者ロール。このロールでは、他の組織のリソースは表示できません。</p> <p>このロールのクローンを作成して組織を割り当てると、その組織の管理をユーザーに委譲できます。</p>
Site manager	<p>さまざまな項目のパーミッションを表示します。</p> <p>インフラストラクチャー内のホストを管理するパーミッション。</p>	Manager ロールの制限バージョン。
System admin	<p>管理 > 設定 でグローバル設定を編集します。</p> <p>ユーザー、ユーザーグループ、ロールを表示、作成、編集、破棄します。</p> <p>組織とロケーションを参照、作成、編集、破棄、割り当てますが、その中のリソースは参照しません。</p>	このロールが指定されたユーザーは、ユーザーを作成したり、そのユーザーに全ロールを割り当てたりできます。このロールは信頼できるユーザーにのみ割り当ててください。
Viewer	Satellite の構造、ログ、レポート、統計情報の全要素の設定を表示します。	

9.7. 詳細なパーミッションフィルタリング

「[ロールへのパーミッションの追加](#)」の説明にあるように、Red Hat Satellite では、リソースタイプの選択済みインスタンスに対する設定済みユーザーパーミッションを制限できます。これらの詳細なフィルターは Satellite データベースに対するクエリーであり、ほとんどのリソースタイプでサポートされています。

9.7.1. 詳細なパーミッションフィルターの作成

以下の手順を使用して、詳細なフィルターを作成します。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

Satellite では、検索条件はアクション作成には適用されません。たとえば、検索フィールドで `create_locations` アクションを `name = "Default Location"` 式で制限しても、ユーザーが新しく作成されたロケーションにカスタム名を割り当てることができないわけではありません。

手順

フィルターの編集 ページの **検索** フィールドにクエリーを指定します。アクティブにするフィールドに対して **無制限** チェックボックスを選択解除します。クエリーの形式は以下のようになります。

field_name operator value

- **field_name** は、問い合わせるフィールドを示します。利用可能なフィールド名の範囲はリソースタイプによって異なります。たとえば、**Partition Table** リソースタイプでは、クエリーパラメーターとして **family**、**layout**、および **name** が提供されます。
- **operator** は、**field_name** と **value** との間の比較タイプを指定します。適用可能な演算子の概要は、[「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。
- **value** は、フィルタリングに使用される値です。この値は、組織の名前などです。2つの種類のワイルドカード文字がサポートされ、アンダースコア (`_`) は単一の文字を置換し、パーセント記号 (`%`) はゼロ以上の文字を置換します。

ほとんどのリソースタイプに対して、**検索** フィールドは利用可能なパラメーターを示すドロップダウンリストを提供します。このリストは、検索フィールドにカーソルを置くと表示されます。多くのリソースタイプに対しては、**and** や **not**、**has** といった論理演算子を使用してクエリーを組み合わせることができます。

CLI 手順

- 詳細なフィルターを作成するには、以下のように **--search** オプションを指定して **hammer filter create** を入力し、パーミッションフィルターで絞り込みます。

```
# hammer filter create \
--permission-ids 91 \
--search "name ~ ccv*" \
--role qa-user
```

このコマンドは、名前が **ccv** で始まるコンテンツビューのみを表示、作成、編集、破棄するパーミッションを **qa-user** ロールに追加します。

9.7.2. 詳細なパーミッションフィルターの使用例

管理者は、選択されたユーザーが環境パスの特定の部分を変更することを許可できます。以下のフィルターを使用すると、アプリケーションライフサイクルの開発段階にあるコンテンツを使用して作業できますが、実稼働環境にプッシュされるとそのコンテンツにはアクセスできなくなります。

9.7.2.1. ホストリソースタイプのパーミッションの適用

以下のクエリーは、`host-editors` という名前のグループのホストに対してのみ、ホストのリソースタイプに指定されたパーミッションを適用します。

`hostgroup = host-editors`

以下のクエリは、XXXX, Yyyy、または zzzz の文字列に名前が一致するレコードを返します。

```
name ^ (XXXX, Yyyy, zzzz)
```

また、選択された環境に対するパーミッションを制限することもできます。これを行うには、**検索** フィールドに環境名を指定します。以下に例を示します。

```
Dev
```

検索 ch フィールドでより詳細なパーミッションフィルターを使用すると、特定の組織またはロケーションにユーザーパーミッションを制限できます。ただし、リソースタイプによっては、**ロケーション** および **組織** タブを提供する **上書き** チェックボックスが、GUI の代わりとなります。これらのタブでは、利用可能な組織とロケーションのリストから選択できます。詳細は、[「組織固有のマネージャーロールの作成」](#) を参照してください。

9.7.2.2. 組織固有のマネージャーロールの作成

Satellite web UI を使用して **org-1** という名前の単一の組織に制限されたマネージャーロールを作成する方法を示します。

手順

1. Satellite Web UI で、**Administer > Roles** に移動します。
2. 既存の **Organization admin** ロールをクローンします。**フィルター** ボタンの横にあるドロップダウンリストから **クローン** を選択します。この結果、クローンされたロールの名前 (たとえば、**org-1 admin**) を挿入するよう求められます。
3. ロールに関連付けるロケーションと組織をクリックします。
4. **送信** をクリックしてロールを作成します。
5. **org-1 admin** をクリックしてから **フィルター** をクリックし、関連付けられたフィルターを確認します。デフォルトのフィルターはほとんどのケースで機能します。ただし、必要に応じて、**編集** をクリックして各フィルターのプロパティを変更することもできます。フィルターによっては、ロールを追加のロケーションと組織のリソースにアクセスできるようにする場合には、**上書き** オプションを有効にできます。たとえば、**ドメイン** リソースタイプを選択して **上書き** オプションを選択し、**ロケーション** と **組織** タブを使用して追加のロケーションと組織を選択すると、このロールに関連付けられていない追加のロケーションと組織のドメインにこのロールがアクセスできるようになります。また、**New filter** をクリックして、新しいフィルターをこのロールに関連付けることもできます。

9.7.3. 詳細な検索に対してサポートされる演算子

表9.1 論理演算子

演算子	説明
and	検索条件を組み合わせます。
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必要です。

表9.2 記号演算子

演算子	説明
=	Is equal to 。テキストフィールドで使用する等価比較。大文字と小文字の区別あり。
!=	Is not equal to 。= 演算子の反転。
~	Like 。テキストフィールドで使用するキーワード出現検索。大文字と小文字の区別なし。
!~	Not like 。~ 演算子の反転。
^	In 。テキストフィールドで使用する等価比較。大文字と小文字を区別した検索。これは、 Is equal to 比較とは別の SQL クエリーを生成し、複数値の比較をより効率的に行えます。
!^	Not in 。^ 演算子の反転。
>, >=	Greater than、greater than or equal to 。数値フィールドのみに対応します。
<, <=	Less than、less than or equal to 。数値フィールドのみに対応します。

第10章 メール通知の設定

メール通知は Satellite Server が定期的に作成するか、特定イベントの完了後に作成されます。定期通知は、毎日、毎週、または毎月送信することができます。

利用可能な通知タイプの概要は、[「電子メール通知の種類」](#) を参照してください。

デフォルトでは、ユーザーにはメールは通知されません。通知のタイプや頻度などの基準に基づいて、ユーザーが通知を受信するように管理者が設定できます。



重要

Satellite Server は、デフォルトで送信メールを有効化していないため、メール設定を確認する必要があります。詳細は、[オンラインネットワーク環境での Satellite Server のインストールの Satellite Server での送信メールの設定](#) を参照してください。

10.1. 電子メール通知の種類

Satellite は、次の電子メール通知を作成できます。

- **監査サマリー**: Satellite Server が監査した全アクティビティのサマリーです。
- **ホストの構築**: ホストが構築されるとこの通知が送信されます。
- **ホストエラータアドバイザリー**: ユーザーが管理するホストの適用およびインストール可能なエラータのサマリーです。
- **コンプライアンスポリシーサマリー**: OpenSCAP ポリシーレポートとその結果のサマリーです。
- **エラータのプロモート**: コンテンツビューのプロモーション後にのみ送信される通知です。これには、プロモートされたコンテンツビューに登録された適用およびインストール可能なエラータのサマリーが含まれます。これにより、どの更新がどのホストに適用されたかを監視できます。
- **エラータの同期**: リポジトリの同期後にのみ送信される通知です。同期によって導入された新しいエラータの概要が含まれています。

電子メール通知タイプの完全なリストを表示するには、Satellite Web UI で **Administer > Users** に移動し、必要なユーザーの **Username** をクリックして、**Email Preferences** タブを選択します。

10.2. 電子メール通知の設定

Satellite に登録されている個々のユーザーに電子メールメッセージを送信するように Satellite を設定できます。Satellite は、アカウントに追加されている電子メールアドレス (存在する場合) に電子メールを送信します。ユーザーは、Satellite web UI の右上にある自分の名前をクリックし、**My account** を選択することで、電子メールアドレスを編集できます。

Satellite Web UI でメール通知を設定します。



注記

メール通知を個人のメールアドレスではなくグループのメールアドレスに送信する場合は、グループのメールアドレスと最小の Satellite パーミッションでユーザーアカウントを作成し、そのユーザーアカウントに必要な通知タイプにサブスクライブします。

手順

1. Satellite Web UI で、**Administer** > **Users** に移動します。
2. 編集する **ユーザー名** をクリックします。
3. **ユーザー** タブで、**メール** フィールドの値を確認します。メール通知は、このフィールドのアドレスに送信されます。
4. **メール設定** タブで **メールの有効化** を選択します。
5. 通知タイプの横にあるドロップダウンメニューから、ユーザーが受信する通知を選択します。



注記

メールクエリー テキストボックスに必要なクエリーを記入すると、**監査サマリー** 通知をフィルターすることができます。

6. **Submit** をクリックします。
通知メールのユーザーへの送信が開始されます。

10.3. メールの配信テスト

メールの配信を確認するには、テストメールをユーザーに送信します。メールが配信されれば、設定が適切であることを確認できます。

手順

1. Satellite Web UI で、**Administer** > **Users** に移動します。
2. ユーザー名をクリックします。
3. **メール設定** タブで **テストメール** をクリックします。
ユーザーのメールアドレスにすぐにテストメッセージが送信されます。

メールが配信されれば、確認は完了です。配信されない場合は、以下の診断ステップを実行してください。

- a. ユーザーのメールアドレスを確認します。
- b. Satellite Server のメール設定を確認します。
- c. ファイアウォールおよびメールサーバーのログを調べます。

10.4. 電子メール通知のテスト

ユーザーが正常に E メール通知をサブスクライブしていることを確認するには、手動で通知をトリガーします。

手順

- 通知をトリガーするには、以下のコマンドを実行します。

```
# foreman-rake reports:_My_Frequency_
```

My_Frequency を次のいずれかに置き換えます。

- daily
- weekly (毎週)
- monthly (毎月)

これでサブスクライブしている全ユーザーに指定された頻度ですべての予定されている通知が配信されます。サブスクライブしているユーザーがすべて通知を受信すれば、検証に成功しています。



注記

現在、手動でトリガーした通知の個別ユーザーへの送信は、サポート対象外です。

10.5. ホストの電子メール通知設定の変更

Satellite は、ホストのイベント通知をホストの登録済み所有者に送信できます。個々のユーザーまたはユーザーグループのいずれかに電子メール通知を送信するように Satellite を設定できます。ユーザーグループに設定すると、電子メールタイプにサブスクライブしているすべてのグループメンバーがメッセージを受信します。

ホストの電子メール通知を受信することは便利ですが、たとえば、既知の問題や回避しているエラーが原因で頻繁にエラーを受信することが予想される場合は、圧倒されます。

手順

1. Satellite Web UI で、**Hosts > All Hosts** に移動し、表示するホストを見つけて、**Actions** 列の **Edit** をクリックします。
2. **Additional Information** タブに移動します。 **Include this host within Satellite reporting** チェックボックスがオンになっている場合、そのホストで E メール通知が有効になります。
3. オプション: E メール通知を有効または無効にするには、チェックボックスを切り替えます。



注記

E メール通知を受け取る場合は、ユーザー設定で電子メールアドレスが設定されていることを確認してください。

第11章 SATELLITE SERVER および CAPSULE SERVER のバックアップ

災害発生時に、Red Hat Satellite デプロイメントと関連データの継続性を確保するために、Satellite デプロイメントをバックアップすることができます。デプロイメントでカスタム設定を使用する場合は、バックアップおよび災害復旧ポリシーを策定する際にカスタム設定をどのように扱うかについて考慮する必要があります。



注記

Satellite Server の新規インスタンスを作成する場合は、バックアップを復元した後に以前のインスタンスを廃止します。クローン作成されたインスタンスは、実稼働環境で並行して実行することは想定されていません。

Satellite Server と Capsule Server のどちらか、およびすべての関連データのバックアップを作成するには、**satellite-maintain backup** コマンドを使用します。別のシステム上の別のストレージデバイスにバックアップすることを強く推奨します。

バックアップ中は Satellite サービスは利用できません。したがって、他の管理者が他のタスクをスケジュールしていないか確認する必要があります。**cron** を使用して、バックアップをスケジュールできます。詳細は、「[週次の完全バックアップ後の日次増分バックアップ例](#)」を参照してください。

オフラインバックアップまたはスナップショットバックアップ中はサービスが非アクティブになり、Satellite はメンテナンスモードに入ります。ポート 443 上での外部からのトラフィックはすべてファイアウォールで拒否され、修正がトリガーされなくなります。

バックアップには、**/root/ssl-build** ディレクトリーの機密情報が含まれます。たとえば、ホスト名、SSH キー、要求ファイル、SSL 証明書が含まれる場合があります。バックアップを暗号化するか、安全なロケーションに移動し、破損のリスクやホストへの不正アクセスを最小限に抑えます。

従来のバックアップ方法

従来のバックアップ方法を使用することもできます。詳細は、Red Hat Enterprise Linux 8 の基本的なシステム設定の設定の [システムの回復と復元](#) を参照してください。



注記

satellite-maintain backup コマンドを使用してバックアップを作成する場合は、Satellite サービスを停止しないでください。

- スナップショットまたは従来のバックアップを作成するときは、以下のようにすべてのサービスを停止する必要があります。

```
# satellite-maintain service stop
```

- スナップショットまたは従来のバックアップを作成したら、サービスを起動します。

```
# satellite-maintain service start
```

11.1. バックアップサイズの予測

完全バックアップでは、PostgreSQL および Pulp のデータベースファイルと Satellite 設定ファイルの非圧縮アーカイブを作成します。Satellite サービスが利用できない時間を短縮するため、圧縮はアーカイブの作成後に実行されます。

完全バックアップには、以下のデータを格納するための領域が必要です。

- 非圧縮の Satellite データベースおよび設定ファイル。
- 圧縮された Satellite データベースおよび設定ファイル。
- バックアップを確実にするため、予測領域全体の 20% を追加。

手順

1. **du** コマンドを入力して、Satellite データベースおよび設定ファイルを含む非圧縮ディレクトリーのサイズを予測します。

```
# du -sh /var/lib/pgsql/data /var/lib/pulp
100G  /var/lib/pgsql/data
100G /var/lib/pulp

# du -csh /var/lib/qpidd /var/lib/tftpboot /etc /root/ssl-build \
/var/www/html/pub /opt/puppetlabs
886M  /var/lib/qpidd
16M   /var/lib/tftpboot
37M   /etc
900K  /root/ssl-build
100K  /var/www/html/pub
2M    /opt/puppetlabs
942M  total
```

2. 圧縮データを保存するために必要な領域を計算します。
以下の表は、バックアップに含まれるすべてのデータ項目の圧縮率を示しています。

表11.1 バックアップデータ圧縮率

データ型	ディレクトリー	比率	圧縮結果の例
PostgreSQL データベースファイル	/var/lib/pgsql/data	80–85%	100 GB → 20 GB
Pulp RPM ファイル	/var/lib/pulp	(非圧縮)	100 GB
設定ファイル	/var/lib/qpidd /var/lib/tftpboot /etc /root/ssl-build /var/www/html/pub /opt/puppetlabs	85%	942 MB → 141 MB

この例では、圧縮されたバックアップデータは合計 120 GB を占有します。

3. バックアップの保存に必要な利用可能な領域を計算するには、圧縮および非圧縮のバックアップデータの予測値を合計し、合計値の 20% をさらに追加してバックアップの信頼性を高めます。
この例では、非圧縮および圧縮のバックアップデータに 201 GB と 120 GB の合計 321 GB が必要です。64 GB の予備領域もあわせ、385-21-2 GB をバックアップのロケーションに割り当てる必要があります。

11.2. SATELLITE SERVER または CAPSULE SERVER の完全バックアップの実行

Red Hat Satellite は、**satellite-maintain backup** コマンドを使用してバックアップを作成します。

Satellite Server のバックアップには、以下の 3 つの方法があります。

- オフラインバックアップ
 - オンラインバックアップ
 - スナップショットバックアップ
- それぞれの方法の詳細については、各バックアップ方法の使用法ステートメントを表示できます。

オフラインバックアップ

```
# satellite-maintain backup offline --help
```

オンラインバックアップ

```
# satellite-maintain backup online --help
```

スナップショットのバックアップ

```
# satellite-maintain backup snapshot --help
```

ディレクトリーの作成

satellite-maintain backup コマンドを実行すると、指定したバックアップディレクトリーにタイムスタンプの付いたサブディレクトリーが作成されます。**satellite-maintain backup** コマンドではバックアップは上書きされないの、バックアップまたは増分バックアップから復元するには、適切なディレクトリーまたはサブディレクトリーを選択する必要があります。**satellite-maintain backup** コマンドは、必要に応じてサービスを停止したり、再開したりします。

satellite-maintain backup offline コマンドを実行すると、以下のデフォルトのバックアップディレクトリーが作成されます。

- Satellite では **satellite-backup**
- Capsule では **foreman-proxy-backup**

カスタムディレクトリー名を設定する場合は、**--preserve-directory** オプションを追加して、ディレクトリー名を追加します。バックアップはその後、コマンドラインで指定したディレクトリーに保存されます。**--preserve-directory** オプションを使用する場合、バックアップが失敗してもデータは削除されません。

ローカルの PostgreSQL データベースを使用する場合、**postgres** ユーザーには、バックアップディレクトリーへの書き込みアクセス権が必要です。

リモートデータベース

satellite-maintain backup コマンドを使用して、リモートデータベースをバックアップできます。

リモートデータベースのバックアップには、オンラインとオフラインの両方の方法を使用できますが、スナップショットなどのオフライン方法を使用すると **satellite-maintain backup** コマンドはデータベースダンプを実行します。

リモート NFS 共有へのバックアップ

Satellite でバックアップを NFS 共有に保存できるようにするには、Satellite Server または Capsule Server の **root** ユーザーが NFS 共有に書き込みできることを確認します。これを防ぐには、**root_squash** や **all_squash** などの NFS エクスポートオプションを確認することができます。

詳細は、[Red Hat Enterprise Linux ネットワークファイルサービスの設定と使用](#) および [Red Hat Enterprise Linux ネットワークサービスの セキュリティー 保護](#) を参照してください。

前提条件

- バックアップロケーションには、バックアップを保存するのに十分な空きディスク領域があること。詳細は、[「バックアップサイズの予測」](#) を参照してください。



警告

Satellite Server および Capsule Server の他のユーザーにすべての変更を保存するよう指示して、バックアップ中は Satellite サービスが利用できないことを警告してください。バックアップと同じ時間に他のタスクがスケジュールされていないことを確認してください。

手順

- Satellite Server で以下のコマンドを入力します。

```
# satellite-maintain backup offline /var/satellite-backup
```

- Capsule Server で以下のコマンドを入力します。

```
# satellite-maintain backup offline /var/foreman-proxy-backup
```

11.3. PULP コンテンツなしでのバックアップの実行

Pulp ディレクトリーの内容を除外するオフラインバックアップを実行できます。Pulp コンテンツなしのバックアップはデバッグに役に立ち、Pulp データベースのバックアップなしに設定ファイルへのアクセスを提供することを目的としています。実稼働ユースケースでは、Pulp コンテンツが含まれていないディレクトリーから復元しないでください。



警告

Satellite Server および Capsule Server の他のユーザーにすべての変更を保存するよう指示して、バックアップ中は Satellite サービスが利用できないことを警告してください。バックアップと同じ時間に他のタスクがスケジュールされていないことを確認してください。

前提条件

- バックアップロケーションには、バックアップを保存するのに十分な空きディスク領域があること。詳細は、「[バックアップサイズの予測](#)」を参照してください。

手順

- Pulp コンテンツなしでオフラインバックアップを実行するには、以下のコマンドを入力します。

```
# satellite-maintain backup offline --skip-pulp-content /var/backup_directory
```

11.4. 増分バックアップの実行

この手順を使用して、前回のバックアップ以降のすべての変更のオフラインバックアップを実行します。

増分バックアップを実行するには、シーケンスの最初の増分バックアップを作成するための参照として完全バックアップを実行する必要があります。復元用として、最新の完全バックアップと、増分バックアップの完全なシーケンスを保持します。



警告

Satellite Server および Capsule Server の他のユーザーにすべての変更を保存するよう指示して、バックアップ中は Satellite サービスが利用できないことを警告してください。バックアップと同じ時間に他のタスクがスケジュールされていないことを確認してください。

前提条件

- バックアップロケーションには、バックアップを保存するのに十分な空きディスク領域があること。詳細は、「[バックアップサイズの予測](#)」を参照してください。

手順

1. 完全なオフラインバックアップを実行するには、以下のコマンドを入力します。

```
# satellite-maintain backup offline /var/backup_directory
```

2. バックアップディレクトリー内にディレクトリーを作成し、初回増分バックアップを保存するには、**--incremental** オプションを使用する **satellite-maintain backup** コマンドを入力します。

```
# satellite-maintain backup offline --incremental /var/backup_directory/full_backup
/var/backup_directory
```

3. 2 回目増分バックアップを作成するには、**--incremental** オプションを使用する **satellite-maintain backup** コマンドを入力し、次回増分の開始点を示すために、初回増分バックアップへのパスを追加します。これで 2 回目増分バックアップのディレクトリーがバックアップディレクトリー内に作成されます。

```
# satellite-maintain backup offline --incremental
/var/backup_directory/first_incremental_backup /var/backup_directory
```

4. オプション: 別のバージョンのバックアップをポイントし、開始点としてそのバックアップバージョンでの増分シリーズを作成する場合は、いつでもこれを行うことができます。たとえば、初回もしくは 2 回目増分バックアップからではなく、完全バックアップから新しい増分バックアップを作成するには、完全バックアップディレクトリーをポイントします。

```
# satellite-maintain backup offline --incremental /var/backup_directory/full_backup
/var/backup_directory
```

11.5. 週次の完全バックアップ後の日次増分バックアップ例

以下のスクリプトでは、日曜日に完全バックアップを実行した後に、増分バックアップを毎日実行します。増分バックアップが実行されるたびに、新しいサブディレクトリーが作成されます。このスクリプトでは、日次の cron ジョブが必要になります。

```
#!/bin/bash -e
PATH=/sbin:/bin:/usr/sbin:/usr/bin
DESTINATION=/var/backup_directory
if [[ $(date +%w) == 0 ]]; then
    satellite-maintain backup offline --assumeyes $DESTINATION
else
    LAST=$(ls -td -- $DESTINATION/* | head -n 1)
    satellite-maintain backup offline --assumeyes --incremental "$LAST" $DESTINATION
fi
exit 0
```

satellite-maintain backup コマンドでは、**PATH** 内に **/sbin** ディレクトリーおよび **/usr/sbin** ディレクトリーを格納する必要があり、確認プロンプトをスキップするために **--assumeyes** オプションを使用することに注意してください。

11.6. オンラインバックアップの実行

デバッグ目的でのみ、オンラインバックアップを実行します。

オンラインバックアップに関するリスク

オンラインバックアップ実行時は、Pulp データベースに影響を与える手順がある場合は、Pulp 部分のバックアップ手順は変更がなくなるまで繰り返されます。Pulp データベースのバックアップは Satellite バックアップの中で最も時間のかかる部分であるため、バックアップ中に Pulp データベースが変更される変更を加えると、バックアップ手順が繰り返されます。

実稼働環境では、スナップショット方法を使用します。詳細は、「[スナップショットバックアップの実行](#)」を参照してください。実稼働環境でオンラインバックアップ方法を使用する場合は、バックアップ中に変更がないように注意して実行してください。



警告

Satellite Server および Capsule Server の他のユーザーにすべての変更を保存するよう指示して、バックアップ中は Satellite サービスが利用できないことを警告してください。バックアップと同じ時間に他のタスクがスケジュールされていないことを確認してください。

前提条件

- バックアップロケーションには、バックアップを保存するのに十分な空きディスク領域があること。詳細は、「[バックアップサイズの予測](#)」を参照してください。

手順

- オンラインバックアップを実行するには、以下のコマンドを入力します。

```
# satellite-maintain backup online /var/backup_directory
```

11.7. スナップショットバックアップの実行

Pulp および PostgreSQL ディレクトリーの論理ボリュームマネージャー (LVM) スナップショットを使用するスナップショットバックアップを実行できます。LVM スナップショットからバックアップを作成すると、一貫性のないバックアップのリスクが軽減されます。

スナップショットバックアップ方法は、完全なオフラインバックアップよりも速いため、Satellite のダウンタイムが短縮されます。

使用方法を表示するには、以下のコマンドを入力します。

```
# satellite-maintain backup snapshot -h
```



警告

Satellite Server および Capsule Server の他のユーザーにすべての変更を保存するよう指示して、バックアップ中は Satellite サービスが利用できないことを警告してください。バックアップと同じ時間に他のタスクがスケジュールされていないことを確認してください。

前提条件

- システムは、スナップショットを作成するディレクトリー (`/var/lib/pulp/` および `/var/lib/pgsql/`) に LVM を使用します。
- 関連ボリュームグループ (VG) の空きディスク領域が、スナップショットのサイズの 3 倍あること。正確には、VG には新規スナップショットを受け入れるために十分な、メンバーの論理ボリューム (LV) に予約されていない領域が必要になります。また、LV のいずれかには、バックアップディレクトリー用の十分な空き領域が必要になります。
- ターゲットのバックアップディレクトリーが、スナップショットを作成するディレクトリー以外の LV にあること。

手順

- スナップショットバックアップを実行するには、**satellite-maintain backup snapshot** コマンドを入力します。

```
# satellite-maintain backup snapshot /var/backup_directory
```

satellite-maintain backup snapshot コマンドは、サービスがアクティブな際にスナップショットを作成し、バックアップに影響を与える可能性があるすべてのサービスを停止します。これにより、メンテナンスの時間が短縮されます。バックアップが正常に実行されると、全サービスが再起動され、LVM スナップショットが削除されます。

11.8. バックアップを実行する際のホワイトリスト化とスキップの手順

satellite-maintain backup コマンドを使用したバックアップは、ステップ順に進められます。バックアップの一部を省略するには、**--whitelist** オプションをコマンドに追加し、省略するステップのラベルを追加します。

手順

- 利用可能なステップのラベルをリスト表示するには、以下のコマンドを入力します。

```
# satellite-maintain advanced procedure run -h
```

- バックアップの手順をスキップするには、**--whitelist** オプションを指定して **satellite-maintain backup** コマンドを入力します。以下に例を示します。

```
# satellite-maintain backup online \
--whitelist backup-metadata \
/var/backup_directory
```

第12章 バックアップからの SATELLITE SERVER または CAPSULE SERVER の復元

Satellite Server または Capsule Server は、[11章 Satellite Server および Capsule Server のバックアップ](#)の一部として作成したバックアップデータから復元することができます。このプロセスでは、バックアップを生成したサーバーと同じサーバーでバックアップを復元する方法を概説します。バックアップに含まれる全データはターゲットシステムで削除されます。元のシステムが利用できない場合は、同じ設定およびホスト名でシステムをプロビジョニングしてください。

12.1. 完全バックアップからの復元

以下の手順を使用して、完全バックアップから Red Hat Satellite または Capsule Server を復元します。復元プロセスが完了するとすべてのプロセスがオンラインになり、すべてのデータベースおよびシステム設定がバックアップ時の状態に戻ります。

前提条件

- 適切なインスタンスを復元していること。Red Hat Satellite インスタンスでホスト名、設定が同一であり、マイナーバージョン (X.Y) が元のシステムと同じである必要があります。
- 既存のターゲットディレクトリーがあること。ターゲットディレクトリーは、アーカイブ内に含まれている設定ファイルから読み取られます。
- Satellite Server または Capsule Server のベースシステムにこのデータを格納するのに十分な領域と、復元後にバックアップ内に含まれる `/etc/` と `/var/` ディレクトリー内のすべてのデータを格納するのに十分な領域があること。
ディレクトリーの使用量を確認するには、以下のコマンドを入力します。

```
# du -sh /var/backup_directory
```

空き領域のサイズを確認するには、以下のコマンドを入力します。

```
# df -h /var/backup_directory
```

`--total` オプションを追加すると複数ディレクトリーの合計結果が取得できます。

- すべての SELinux コンテキストが適切であること。以下のコマンドを入力して、適切な SELinux コンテキストを復元します。

```
# restorecon -Rv /
```

手順

- Satellite または Capsule のインストールに適した方法を選択します。
 - オンライン接続されているネットワークで Satellite Server をインストールするには、[オンラインネットワーク環境での Satellite Server のインストール](#) の手順に従います。
 - オンライン接続されていないネットワークから Satellite Server をインストールするには、[オフラインネットワーク環境での Satellite Server のインストール](#) の手順に従います。
 - Capsule Server をインストールするには、[Capsule Server のインストール](#) の手順に従います。

2. バックアップデータを Satellite Server のローカルファイルシステムにコピーします。/var/ または /var/tmp/ を使用します。
3. 復元スクリプトを実行します。

```
# satellite-maintain restore /var/backup_directory
```

ここでの **backup_directory** は、バックアップされたデータを格納しているタイムスタンプ付きのディレクトリーまたはサブディレクトリーになります。

コピーするデータサイズが原因で、復元プロセスの完了に長い時間がかかることがあります。

関連情報

- トラブルシューティングを行うには、/var/log/foreman/production.log および /var/log/messages にあるファイルを参照してください。

12.2. 増分バックアップからの復元

増分バックアップから Satellite または Capsule Server を復元するには、以下の手順を実行します。複数の増分バックアップの **branch** がある場合は、完全バックアップと復元するブランチの各増分バックアップを時系列で選択します。

復元プロセスが完了するとすべてのプロセスがオンラインになり、すべてのデータベースおよびシステム設定がバックアップ時の状態に戻ります。

手順

1. 「完全バックアップからの復元」の手順を使用して、最新の完全バックアップを復元します。
2. /var/ や /var/tmp/ などの Satellite Server のローカルファイルシステムから完全バックアップデータを削除します。
3. /var/ や /var/tmp/ などの Satellite Server のローカルファイルシステムに増分バックアップデータをコピーします。
4. 増分バックアップが作成された順序で復元します。

```
# satellite-maintain restore /var/backup_directory/FIRST_INCREMENTAL  
# satellite-maintain restore /var/backup_directory/SECOND_INCREMENTAL
```

関連情報

- トラブルシューティングを行うには、/var/log/foreman/production.log および /var/log/messages にあるファイルを参照してください。

12.3. 仮想マシンのスナップショットを使用した CAPSULE SERVER のバックアップと復元

Capsule Server が仮想マシンである場合、スナップショットから復元することができます。復元元となるスナップショットは、毎週作成することが推奨されます。失敗した場合は、新規 Capsule Server をインストールまたは設定し、Satellite Server からデータベースコンテンツを同期します。

必要な場合は、新規 Capsule Server をデプロイして、ホスト名が以前のものと同一であることを確認

し、その後に Capsule 証明書をインストールします。これは、-certs.tar で終わるパッケージ名で、Satellite Server にまだ残っている可能性があります。他のオプションとして、新規に作成します。[Capsule Server のインストール](#)にある手順に従い、Satellite Web UI で Capsule Server が Satellite Server に接続されたことを確認します。この後に、「[Satellite Server から Capsule Server へのコンテンツの同期](#)」の手順で Satellite から同期します。

12.3.1. Satellite Server から Capsule Server へのコンテンツの同期

この手順を使用して、Satellite Server から Capsule Server にコンテンツを同期します。

前提条件

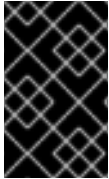
- Capsule Server の関連する組織およびロケーションのコンテキストを選択するか、任意の **Any Organization** および **Any Location** を選択します。

手順

1. Satellite Web UI で、**Infrastructure > Capsules** に移動します。
2. Capsule Server を選択します。
3. **Overview** タブで **Synchronize** をクリックします。
 - **Optimized Sync** を選択すると、Satellite Server から Capsule Server にコンテンツが同期され、不必要な手順が省略され、パフォーマンスが向上します。
 - **Complete Sync** を選択すると、Satellite Server から Capsule Server への完全同期が実行され、メタデータが変更されていない場合でもコンテンツが同期されます。

第13章 SATELLITE SERVER または CAPSULE SERVER の名前の変更

Satellite Server または Capsule Server の名前を変更するには、**satellite-change-hostname** スクリプトを使用します。



重要

Satellite Server または Capsule Server のドメイン名を変更する場合は、ネットワークの問題を回避するために、**satellite-change-hostname** を使用してホスト名を更新します。

13.1. SATELLITE SERVER の名前の変更

Satellite Server のホスト名は、Satellite Server のコンポーネント、すべての Capsule Server、および Satellite Server に登録されているホストが通信用に使用しています。この手順により、Satellite Server の名前を変更するだけでなく、新しいホスト名を指すようにすべての参照も更新されます。



警告

Satellite Server ホストの名前を変更すると、そのホスト上のすべての Satellite サービスがシャットダウンされます。名前変更が完了すると、サービスが再起動されます。

前提条件

- ホスト名を変更する前に、Satellite Server をバックアップしてください。名前の変更に失敗した場合は、バックアップから復元してください。詳細は、[11章 Satellite Server および Capsule Server のバックアップ](#)を参照してください。
- Satellite Server で **hostname** および **hostname -f** コマンドを実行します。どちらのコマンドも Satellite Server の FQDN を返さない場合、**satellite-change-hostname** スクリプトは完了できません。
hostname コマンドが FQDN ではなく Satellite Server のショートネームを返す場合は、**satellite-change-hostname** スクリプトを使用する前に、**hostnamectl set-hostname My_Old_FQDN** を使用して古い FQDN を正しく設定します。
- Satellite Server にカスタム SSL 証明書がインストールされている場合は、ホストの新しい FQDN の新しい証明書を取得します。詳細は、[オンラインネットワーク環境からの Satellite Server のインストール](#) の [カスタムの SSL 証明書を使用した Satellite Server の設定](#) を参照してください。

手順

- Satellite Server で、**satellite-change-hostname** スクリプトを実行し、新しいホスト名を指定します。以下の方法のいずれかを選択します。
 - Satellite Server をデフォルトの自己署名 SSL 証明書でインストールした場合は、以下のコマンドを入力します。

```
# satellite-change-hostname new-satellite \
--username admin \
--password password
```

- Satellite Server をカスタムの SSL 証明書を使用してインストールした場合は、以下を実行します。

```
# satellite-change-hostname new-satellite \
--username admin \
--password password \
--custom-cert "/root/ownca/test.com/test.com.crt" \
--custom-key "/root/ownca/test.com/test.com.key"
```

2. Satellite Server の新しいホスト名用にカスタム SSL 証明書を作成した場合は、Satellite インストールスクリプトを実行して証明書をインストールします。カスタム SSL 証明書のインストールに関する詳細は、[オンラインネットワーク環境での Satellite Server のインストールの Satellite Server へのカスタム SSL 証明書のデプロイ](#) を参照してください。
3. すべての Satellite ホストを再登録します。詳細は、[ホストの管理](#) の [ホストの登録](#) を参照してください。
4. すべての Capsule Server で、Satellite インストールスクリプトを実行して、新規ホスト名への参照を更新します。

```
# satellite-installer \
--foreman-proxy-foreman-base-url https://new-satellite.example.com \
--foreman-proxy-trusted-hosts new-satellite.example.com \
--puppet-server-foreman-url https://new-satellite.example.com
```

5. Satellite Server で、すべての Capsule Server をリスト表示します。

```
# hammer capsule list
```

6. Satellite Server で、コンテンツを各 Capsule Server に同期します。

```
# hammer capsule content synchronize \
--id My_capsule_ID
```

7. **virt-who** エージェントを使用する場合は、新しいホスト名で **virt-who** 設定ファイルを更新します。詳細は、Red Hat Satellite での仮想マシンサブスクリプションの設定 の **virt-who** 設定の修正 を参照してください。
8. 外部認証を使用している場合は、**satellite-change-hostname** スクリプトの実行後に、外部認証向けに Satellite Server を再設定します。詳細は、[オンラインネットワーク環境での Satellite Server のインストール](#) の 外部認証の設定 を参照してください。

13.2. CAPSULE SERVER の名前の変更

Capsule Server のホスト名は、Satellite Server のコンポーネントおよび Capsule Server に登録されている全ホストが参照しています。この手順により、Capsule Server の名前変更に加えて、新しいホスト名へのすべての参照も更新されます。



警告

Capsule Server ホストの名前を変更すると、そのホスト上のすべての Satellite サービスがシャットダウンされます。名前変更が完了すると、サービスが再起動されます。

前提条件

- 名前を変更する前に、Capsule Server をバックアップしてください。名前の変更に失敗した場合は、バックアップから復元してください。詳細は、[11章 Satellite Server および Capsule Server のバックアップ](#)を参照してください。

- Capsule Server で **hostname** および **hostname -f** コマンドを実行します。どちらのコマンドも Capsule Server の FQDN を返さない場合、**satellite-change-hostname** スクリプトは完了できません。

hostname コマンドが FQDN ではなく Capsule Server のショートネームを返す場合、**satellite-change-hostname** スクリプトを使用する前に、**hostnamectl set-hostname My_Old_FQDN** を使用して古い FQDN を正しく設定する必要があります。

手順

- Satellite Server で、Capsule Server の新しい証明書のアーカイブファイルを生成します。

- デフォルトの SSL 証明書を使用している場合は、以下のコマンドを実行します。

```
# capsule-certs-generate \
--foreman-proxy-fqdn new-capsule.example.com \
--certs-tar /root/new-capsule.example.com-certs.tar
```

.tar ファイルへの完全パスを必ず入力してください。

- カスタムの SSL 証明書を使用している場合は、Capsule Server 用の新しい SSL 証明書を作成します。詳細は、[Capsule Server のインストールの カスタム SSL 証明書を使用した Capsule Server の設定](#)を参照してください。
- Satellite Server で、証明書アーカイブファイルを Capsule Server にコピーします。たとえば、アーカイブファイルを **root** ユーザーのホームディレクトリーにコピーするには、次のようにします。

```
# scp /root/new-capsule.example.com-certs.tar root@capsule.example.com:
```

- Capsule Server で **satellite-change-hostname** スクリプトを実行し、新しいホスト名と Satellite 認証情報、および証明書アーカイブファイル名を提供します。

```
# satellite-change-hostname new-capsule.example.com --username admin \
--password password \
--certs-tar /root/new-capsule.example.com-certs.tar
```

.tar ファイルへの完全パスを必ず入力してください。

- Capsule Server のカスタム証明書を作成した場合は、前の手順で **Capsule-certs-generate** コ

マンドが返した **Satellite-installer** コマンドを入力して、Capsule Server に証明書をデプロイします。詳細は、Capsule Server のインストール の Capsule Server へのカスタム SSL 証明書のデプロイ を参照してください。

5. すべての Capsule クライアントで、以下のコマンドを入力して、ブートストラップ RPM を再インストールし、クライアントを再登録して、サブスクリプションを更新します。
この手順は、リモート実行機能を使用して実行できます。詳細は、[ホストの管理](#) の **リモートジョブの設定およびセットアップ** を参照してください。

```
# dnf remove katello-ca-consumer*

# dnf install http://new-capsule.example.com/pub/katello-ca-consumer-latest.noarch.rpm

# subscription-manager register --org="My_Organization" \
--environment="Library" \
--force

# subscription-manager refresh
```

6. Satellite Web UI で Capsule のホスト名を更新します。
 - a. Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
 - b. リストで Capsule Server を見つけて、**Edit** をクリックします。
 - c. **名前** と **URL** フィールドが Capsule Server の新規ホスト名に一致するように変更して、**送信** をクリックします。
 - d. DNS サーバーで、Capsule Server の新しいホスト名のレコードを追加し、以前のホスト名のレコードを削除します。

第14章 SATELLITE SERVER のメンテナンス

本章では、監査レコードの取り扱い、未使用タスクの消去方法、いっぱいになったディスクから Pulp を復元する方法などの Satellite Server のメンテナンス方法について説明します。

14.1. 監査レコードの手動削除

foreman-rake audits:expire コマンドを使うと、監査レコードはいつでも取り消すことができます。

手順

- **foreman-rake audits:expire** コマンドを使用して、監査レコードを削除します。

```
# foreman-rake audits:expire days=Number_Of_Days
```

このコマンドは、**Number_Of_Days** よりも古いすべての監査レコードを削除します。

14.2. 監査レコードの自動削除

Saved audits interval 設定を使用して、自動的に監査レコードを削除できます。この設定はデフォルトでは空です。つまり、Satellite は監査レコードを自動的に削除しません。

手順

1. Satellite Web UI で、**Administer > Settings** に移動します。
2. **General** タブで、**Saved audits interval** の設定を見つけます。
3. 設定の値を、Satellite が監査レコードを削除する日数に設定します。

14.3. 監査レコードの匿名化

foreman-rake audits:anonymize コマンドを使うと、データベースで監査レコードを保持しつつ、ユーザーアカウントや IP 情報を削除できます。また、cron ジョブを使用して、設定した間隔で、監査レコードの匿名化をスケジューリングすることも可能です。

デフォルトでは、**foreman-rake audits:anonymize** コマンドを使用すると 90 日以上経過した監査レコードが匿名化されます。**days** オプションに日数を追加して、監査レコードを保持する日数を指定することが可能です。

たとえば、7 日以上経過した監査レコードを匿名化する場合は、以下のコマンドを実行します。

```
# foreman-rake audits:anonymize days=7
```

14.4. レポートのレコード削除

レポートレコードは Satellite で自動作成されます。**foreman-rake reports:expire** コマンドを使用して、いつでもレポートを削除できます。また、cron ジョブを使用して、設定した間隔で、レポートレコードの削除をスケジューリングすることも可能です。

デフォルトでは、**foreman-rake reports:expire** コマンドを使用すると 90 日以上経過したレポートレコードが削除されます。**days** オプションに日数を追加して、レポートレコードを保持する日数を指定することが可能です。

たとえば、7 日以上経過したレポートレコードを削除する場合は、以下のコマンドを実行します。

```
# foreman-rake reports:expire days=7
```

14.5. 未使用タスクのクリーニング機能の設定

Satellite はクリーニングを定期的に行うことで、データベース内のディスク領域を削減し、ディスク増加率を制限します。その結果、Satellite のバックアップがより短時間で完了し、全体的なパフォーマンスも向上します。

デフォルトでは、Satellite は毎日 19 時 45 分にタスクをクリーンアップする cron ジョブを実行します。Satellite は、クリーニング中に以下のタスクを削除します。

- 正常に実行され、30 日を超過したタスク
- 1 年を超過したすべてのタスク

次のオプションを使用して、未使用のタスクのクリーニング機能を設定できます。

- Satellite が cron ジョブを実行する時刻を設定するには、**--foreman-plugin-tasks-cron-line** パラメーターを cron 形式で希望する時刻に設定します。たとえば、cron ジョブを毎日 15 時 00 分に実行するようにスケジュールするには、以下のコマンドを入力します。

```
# satellite-installer --foreman-plugin-tasks-cron-line "00 15 * * *"
```

- Satellite がタスクを削除するまでの期間を設定するには、**/etc/foreman/plugins/foreman-tasks.yaml** ファイルの **:rules:** セクションを編集します。
- Satellite で定期的なタスクのクリーンアップを無効にするには、次のコマンドを入力します。

```
# satellite-installer --foreman-plugin-tasks-automatic-cleanup false
```

- Satellite で定期的なタスクのクリーンアップを再度有効にするには、次のコマンドを入力します。

```
# satellite-installer --foreman-plugin-tasks-automatic-cleanup true
```

14.6. タスクレコードの削除

タスクレコードは Satellite で自動作成されます。**foreman-rake foreman_tasks:cleanup** コマンドを使用して、いつでもタスクを削除できます。また、cron ジョブを使用して、設定した間隔で、タスクレコードの削除をスケジュールすることも可能です。

たとえば、正常なレポジトリ同期からタスクレコードを削除する場合には、以下のコマンドを実行します。

```
# foreman-rake foreman_tasks:cleanup TASK_SEARCH='label = Actions::Katello::Repository::Sync' STATES='stopped'
```

14.7. ID によるタスクの削除

誤って機密データを送信した場合など、ID でタスクを削除できます。

手順

1. SSH を使用して Satellite Server に接続します。

```
# ssh root@satellite.example.com
```

2. オプション: タスクを表示します。

```
# hammer task info --id My_Task_ID
```

3. タスクを削除します。

```
# foreman-rake foreman_tasks:cleanup TASK_SEARCH="id=My_Task_ID"
```

4. オプション: タスクが Satellite Server から削除されていることを確認します。

```
# hammer task info --id My_Task_ID
```

タスクが削除されているため、このコマンドはゼロ以外の終了コードを返すことに注意してください。

14.8. 完全なディスクからのリカバリー

以下の手順では、Pulp データベースのある論理ボリューム (LV) に空き領域がない場合の解決方法について説明します。

手順

1. 実行中の Pulp タスクを完了させます。新たなタスクは開始しないでください。ディスクに空きスペースがないため、失敗することになります。
2. **/var/lib/pulp** ディレクトリーのある LV に十分な空き領域があることを確認します。以下のような方法があります。

- a. 孤立したコンテンツを削除します:

```
# foreman-rake katello:delete_orphaned_content RAILS_ENV=production
```

これは1週間ごとに実行されるので、多くの領域が解放されるわけではありません。

- b. できるだけ多くのレポートのダウンロードポリシーを **即時** から **オンデマンド** に変更し、ダウンロード済みパッケージを削除します。手順については、Red Hat カスタマーポータル [のナレッジベースのソリューション How to change syncing policy for Repositories on Satellite from "Immediate" to "On-Demand"](#) を参照してください。
- c. **/var/lib/pulp** ディレクトリーのある LV 上のファイルシステムを拡張します。詳細は、Red Hat Enterprise Linux 8 [の論理ボリュームの設定および管理](#) の [論理ボリュームおよびファイルシステムの拡張](#) を参照してください。



注記

(ext3、ext4、または xfs などの) 通常外のファイルシステムを使用している場合は、そのファイルシステムをアンマウントして使用されていない状態にする必要があります。その場合は、以下を実行します。

1. Satellite サービスを停止します。

```
# satellite-maintain service stop
```

2. LV 上のファイルシステムを拡張します。

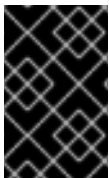
3. Satellite サービスを起動します。

```
# satellite-maintain service start
```

3. ディスクに空きスペースがないために Pulp タスクが失敗していた場合は、それらのタスクを再実行します。

14.9. SATELLITE SERVER または CAPSULESERVER のベースオペレーティングシステムでのパッケージの管理

Satellite Server または Capsule Server のベースオペレーティングシステムでパッケージをインストールして更新するには、**satellite-maintain packages** コマンドを入力する必要があります。Satellite では、**yum** を使用したパッケージのインストールと更新はできません。理由は、**yum** が Satellite Server または Capsule Server 関連のパッケージも更新し、その結果システムの不整合が発生する可能性があるためです。



重要

パッケージのインストール後に **satellite-installer** コマンドが実行されるため、**satellite-maintain packages** コマンドは、このコマンドを実行するオペレーティングシステムのサービスの一部を再起動します。

以下のように **satellite-maintain packages** コマンドを使用してパッケージを管理できます。

- Satellite Server または Capsule Server にパッケージをインストールするには、以下を実行します。

```
# satellite-maintain packages install package_1 package_2
```

- Satellite Server または Capsule Server で利用可能なパッケージの更新を確認するには、以下を実行します。

```
# satellite-maintain packages check-update
```

- Satellite Server または Capsule Server ですべてのパッケージを更新するには、以下を実行します。

```
# satellite-maintain packages update
```

- Satellite Server または Capsule Server で特定のパッケージを更新するには、以下を実行します。

```
# satellite-maintain packages update package_1 package_2
```

パッケージを個別に更新すると、Satellite Server または Capsule Server でパッケージの不整合が発生する可能性があります。Satellite Server のパッケージ更新に関する詳細は、[Red Hat Satellite の更新の Satellite Server の次のマイナーバージョンへの更新](#) を参照してください。

14.10. POSTGRESQL 領域の確保

PostgreSQL データベースは、特に負荷の高いデプロイメントにおいて、大容量のディスク領域を使用できます。Satellite でこのディスク領域の一部を回収するには、この手順を使用します。

手順

1. **postgresql** サービス以外の全サービスを停止します。

```
# satellite-maintain service stop --exclude postgresql
```

2. **postgres** ユーザーに切り替えて、データベースの領域を回収します。

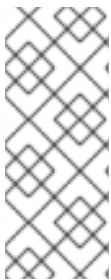
```
# su - postgres -c 'vacuumdb --full --all'
```

3. Vacuum が完了したら、他のサービスを開始します。

```
# satellite-maintain service start
```

14.11. オンデマンドリポジトリからのスペースの再利用

download policy をオンデマンドに設定すると、Satellite はクライアントがパッケージを要求した場合にのみパッケージをダウンロードします。これらのパッケージをクリーンアップして、領域を回収できます。



注記

スペースの再利用には既存のリポジトリが必要です。削除されたりリポジトリの場合は、次にスケジュールされている孤立したクリーンアップを待つか、孤立したコンテンツを手動で削除します。

```
# foreman-rake katello:delete_orphaned_content
```

単一のリポジトリの場合

- Satellite Web UI で、**Content > Products** に移動します。
- 製品を選択します。
- **Repositories** タブで、リポジトリ名をクリックします。
- **Select Actions** リストから、**Reclaim Space** を選択します。

複数のリポジトリの場合

- Satellite Web UI で、**Content** > **Products** に移動します。
- 製品名を選択します。
- **Repositories** タブで、リポジトリのチェックボックスを選択します。
- 右上隅にある **Reclaim Space** をクリックします。

Capsules の場合

- Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
- Capsule Server を選択します。
- **Reclaim space** をクリックします。

第15章 カスタム SSL 証明書の更新

この章では、Satellite Server および Capsule Server でカスタム SSL 証明書を更新する方法について説明します。

15.1. SATELLITE SERVER でのカスタム SSL 証明書の更新

この手順を使用して、Satellite Server のカスタム SSL 証明書を更新します。

前提条件

- 新しい証明書署名要求 (CSR) を作成し、それを認証局に送信して証明書に署名する必要があります。サーバー証明書には、必要な値を含む X.509 v3 **Key Usage** および **Extended Key Usage** 拡張が必要です。そのため、新しい CSR を作成する前に、[カスタムの SSL 証明書を使用した Satellite Server の設定](#) ガイドを参照してください。送信すると、Satellite Server 証明書と CA バンドルが返却されます。

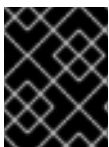
手順

- 更新されたカスタム証明書を Satellite Server にデプロイする前に、カスタムの SSL 入力ファイルを検証します。**katello-certs-check** コマンドが正しく実行されるには、証明書の共通ネーム (CN) が Satellite Server の FQDN と一致する必要があることに注意してください。

```
# katello-certs-check -t satellite \
-b /root/satellite_cert/ca_cert_bundle.pem \
-c /root/satellite_cert/satellite_cert.pem \
-k /root/satellite_cert/satellite_cert_key.pem
```

コマンドが正常に実行されると、以下の **satellite-installer** コマンドが返されます。このコマンドを使用して、更新された CA 証明書を Satellite Server にデプロイできます。

```
# satellite-installer --scenario satellite \
--certs-server-cert "/root/satellite_cert/satellite_cert.pem" \
--certs-server-key "/root/satellite_cert/satellite_cert_key.pem" \
--certs-server-ca-cert "/root/satellite_cert/ca_cert_bundle.pem" \
--certs-update-server \
--certs-update-server-ca
```



重要

証明書のデプロイ後に証明書ファイルを削除しないでください。これらは Satellite Server のアップグレード時に必要です。



注記

別の証明書署名局により、新しいコンシューマーパッケージ **katello-ca-consumer-latest.noarch.rpm** が生成される場合は、Satellite Server に登録されたすべてのクライアントを更新する必要があります。

検証

- ローカルマシンから Satellite Web UI にアクセスします。たとえば、<https://satellite.example.com> からアクセスします。

2. ブラウザーで、証明書の詳細を表示して、デプロイした証明書を確認します。

15.2. CAPSULE SERVER でのカスタム SSL 証明書の更新

この手順を使用して、Capsule Server のカスタム SSL 証明書を更新します。**capsule-certs-generate** コマンドにより返される、**satellite-installer** コマンドは、Capsule Server ごとに一意となっています。複数の Capsule Server で同じコマンドを使用することはできません。

前提条件

- 新しい証明書署名要求を作成し、認証局に送信して証明書に署名する必要があります。
Satellite Server 証明書には、必要な値を含む X.509 v3 **Key Usage** および **Extended Key Usage** 拡張が必要です。そのため、新しい CSR を作成する前に、[カスタムの SSL 証明書を使用した Satellite Server の設定](#) ガイドを参照してください。代わりに、Capsule Server 証明書と CA バンドルを受け取ります。

手順

1. Satellite Server で、カスタムの SSL 証明書の入力ファイルを検証します。

```
# katello-certs-check -t capsule \
-b /root/capsule_cert/ca_cert_bundle.pem \
-c /root/capsule_cert/capsule_cert.pem \
-k /root/capsule_cert/capsule_cert_key.pem
```

2. Satellite Server で、Capsule Server の証明書アーカイブファイルを生成します。

```
# capsule-certs-generate \
--certs-tar "/root/My_Certificates/capsule.example.com-certs.tar" \
--certs-update-server \
--foreman-proxy-fqdn "capsule.example.com" \
--server-ca-cert "/root/My_Certificates/ca_cert_bundle.pem" \
--server-cert "/root/My_Certificates/capsule_cert.pem" \
--server-key "/root/My_Certificates/capsule_cert_key.pem"
```

3. Satellite Server から、証明書アーカイブファイルを Capsule Server にコピーします。

```
# scp /root/My_Certificates/capsule.example.com-certs.tar
user@capsule.example.com:
```

必要に応じて、コピーしたファイルを該当するパスに移動できます。

4. **capsule-certs-generate** コマンドが返す **satellite-installer** コマンドのコピーをメモし、Capsule Server に証明書をデプロイします。
5. **capsule-certs-generate** コマンドで返された **satellite-installer** コマンドを使用して、Capsule Server に証明書をデプロイします。

```
# satellite-installer --scenario capsule \
--certs-tar-file "/root/My_Certificates/capsule.example.com-certs.tar" \
--certs-update-server \
--foreman-proxy-foreman-base-url "https://satellite.example.com" \
--foreman-proxy-register-in-foreman "true"
```



重要

証明書のデプロイ後に、Capsule Server で証明書アーカイブファイルを削除しないでください。これらは、Capsule Server のアップグレード時に必要です。



注記

別の証明書署名局により、新しいコンシューマーパッケージ **katello-ca-consumer-latest.noarch.rpm** が生成される場合は、Capsule Server に登録されたすべてのクライアントを更新する必要があります。

第16章 問題のログとレポート

次のセクションでは、Satellite で使用できる最も一般的なログファイルとデバッグツールについて説明します。

16.1. デバッグに役立つログレベルの引き上げ

Satellite のトラブルシューティングを行うために、Satellite コンポーネントのログレベルを上げることができます。ログレベルを **debug** に上げると、最も詳細な情報を取得できます。デフォルトのログレベルは **info** です。

16.1.1. Foreman アプリケーションのログレベルを上げる

Foreman アプリケーションのロギングレベルを上げるには、以下を実行します。

1. ログレベルを **debug** に設定します。

```
# satellite-installer --foreman-logging-level debug
```

2. デバッグが完了したら、ログレベルをデフォルト値にリセットします。

```
# satellite-installer --reset-foreman-logging-level
```

Satellite のロギング設定の詳細は、**satellite-installer** を使用し、**--full-help** オプションを指定します。

```
# satellite-installer --full-help | grep logging
```

16.1.2. Hammer のログレベルを上げる

ログは `~/.hammer/log/hammer.log` にあります。

Hammer のロギングレベルを上げるには、以下を実行します。

- `/etc/hammer/cli_config.yml` で、`:log_level:` オプションを **debug** に設定します。

```
:log_level: 'debug'
```

16.1.3. Capsule のログレベルを上げる

ログは `/var/log/foreman-proxy/proxy.log` にあります。

Capsule のロギングレベルを上げるには、以下を実行します。

1. ログレベルを **debug** に設定します。

```
# satellite-installer --foreman-proxy-log-level DEBUG
```

2. デバッグが完了したら、ログレベルをデフォルト値にリセットします。

```
# satellite-installer --reset-foreman-proxy-log-level
```

**注記**

satellite-installer を実行すると、設定がデフォルトに戻ります。

16.1.4. Candlepin のログレベルを上げる

ログは `/var/log/candlepin/candlepin.log` および `/var/log/candlepin/error.log` にあります。

Candlepin のロギングレベルを上げるには、以下を実行します。

1. `/etc/candlepin/candlepin.conf` で、ログレベルを **DEBUG** に設定します。

```
log4j.logger.org.candlepin=DEBUG
```

2. **tomcat** サービスを起動します。

```
# systemctl restart tomcat
```

candlepin ログファイルが冗長すぎる場合は、デフォルトのデバッグレベルを下げることができます。

```
log4j.logger.org.candlepin.resource.ConsumerResource=WARN
log4j.logger.org.candlepin.resource.HypervisorResource=WARN
```

16.1.5. Qpid Dispatch Router のログレベルを上げる

Qpid は **syslog** にログを記録します。ログは `/var/log/messages` に表示されます。または **journalctl** を使用して表示できます。

Qpid Dispatch Router のログレベルを上げるには、以下を実行します。

1. `/etc/qpid-dispatch/qdrouterd.conf` で、ログレベルを **debug** に設定します。

```
enable: debug+
```

2. Qpid Dispatch Router を再起動します。

```
# systemctl restart qdrouterd
```

**注記**

satellite-installer を実行すると、設定がデフォルトに戻ります。

16.1.6. Qpid Broker のログレベルの引き上げ

Qpid は **syslog** にログを記録します。ログは `/var/log/messages` に表示されます。または **journalctl** を使用して表示できます。

Qpid Broker のログレベルを上げるには、以下を実行します。

1. `/etc/qpid/qpid.conf` で、ログレベルを **debug** に設定します。

```
log-enable=debug+
```


2. Qpid Broker を再起動します。

```
# systemctl restart qpidd
```



注記

satellite-installer を実行すると、設定がデフォルトに戻ります。

16.1.7. Redis のログレベルの引き上げ

Redis のログは **/var/log/redis/redis.log** にあります。

Redis のログレベルを上げるには、以下を実行します。

1. **/etc/redis.conf** で、ログレベルを **debug** に設定します。

```
loglevel debug
```

2. Redis サービスを再起動します。

```
# systemctl restart redis
```



注記

satellite-installer を実行すると、設定がデフォルトに戻ります。

16.1.8. Satellite インストーラーのログレベルの増加

ログファイルは **/var/log/foreman-installer/** にあります。

Satellite インストーラーのログレベルを上げるには、インストール時に **--verbose-log-level debug** オプションを追加します。

```
# satellite-installer --verbose-log-level debug
```

16.1.9. Pulp のログレベルの引き上げ

デフォルトでは、Pulp は **syslog** にログを記録します。ログは **/var/log/messages** に表示されます。または **journalctl** を使用して表示できます。

Pulp のログレベルを上げるには、以下を実行します。

1. **/etc/pulp/settings.py** で、ログレベルを **DEBUG** に設定します。

```
LOGGING = {"dynaconf_merge": True, "loggers": {"": {"handlers": ['console'], 'level': 'DEBUG'}}
```

2. Pulp サービスを再起動します。

```
# systemctl restart \
pulpcore-api \
pulpcore-content \
pulpcore-resource-manager \
```

```
pulpcore-worker@1 \
pulpcore-worker@2 \
redis
```

16.1.10. Puppet Agent のログレベルの引き上げ

ログは `/var/log/puppetlabs/puppet/` にあります。

Puppet エージェントのログレベルを上げるには、以下を実行します。

1. Satellite で Puppet が有効になっていることを確認します。詳細は、[Red Hat Satellite で Puppet インテグレーションを使用して設定を管理する](#) の [Satellite で Puppet インテグレーションを有効にする](#) を参照してください。
2. ログレベルを **debug** に設定します。

```
# satellite-installer --puppet-agent-additional-settings log_level:debug
```

16.1.11. Puppet サーバーのログレベルの引き上げ

ログは `/var/log/puppetlabs/puppetserver/` にあります。

Puppet サーバーのログレベルを上げるには、以下を実行します。

1. Satellite で Puppet が有効になっていることを確認します。詳細は、[Red Hat Satellite で Puppet インテグレーションを使用して設定を管理する](#) の [Satellite で Puppet インテグレーションを有効にする](#) を参照してください。
2. ログレベルを **debug** に設定します。

```
# satellite-installer --puppet-server-additional-settings log_level:debug
```

3. Puppet サーバーを再起動します。

```
# satellite-maintain service restart --only puppetserver
```

16.2. HAMMER CLI を使用してログレベルを設定する

Hammer を設定して、さまざまな Satellite コンポーネントのデバッグ情報をログに記録できます。デフォルトのログレベルは **INFO** です。

- **hammer** 使用してログレベルを変更できるコンポーネントをリストするには、以下を実行します。

```
# hammer admin logging --list
```

- すべてのコンポーネントのログレベルを **DEBUG** に設定するには、以下を実行します。

```
# hammer admin logging --all --level-debug
# satellite-maintain service restart
```

- すべてのコンポーネントのログレベルを **PRODUCTION** に設定するには、以下を実行します。

```
# hammer admin logging --all --level-production
# satellite-maintain service restart
```

- 特定のコンポーネントのログレベルを **DEBUG** に設定するには、以下を実行します。

```
# hammer admin logging --components My_Component --level-debug
# satellite-maintain service restart
```

- 利用可能なロギングオプションをリスト表示するには、以下のコマンドを使用します。

```
# hammer admin logging --help
```

関連情報

- [Hammer CLI ガイド](#)

16.3. ジャーナルまたはファイルベースのログへのログの設定

Satellite はデフォルトでファイルベースのログを使用します。ログ記録を再設定するには、**satellite-installer** コマンドを使用できます。

Journal を使用したロギングの設定手順

1. 以下の **satellite-installer** コマンドを入力して、ロギングを **journald** サービスに設定します。

```
# satellite-installer \
--foreman-logging-layout pattern \
--foreman-logging-type journald \
--foreman-proxy-log JOURNAL
```

2. オプション: ログメッセージを検査するには、**journalctl** ユーティリティを使用します。以下に例を示します。
 - **journalctl --unit foreman** および **journalctl --unit foreman-proxy** は、**foreman** および **foreman-proxy** ユニットのメッセージを表示します。
 - **journalctl REQUEST=request_ID** は、指定されたリクエストのメッセージを表示します。

ファイルベースのロギングを設定する手順

1. 次の **satellite-installer** コマンドを入力して、ファイルベースのログを設定します。

```
# satellite-installer \
--reset-foreman-logging-layout \
--reset-foreman-logging-type \
--reset-foreman-proxy-log
```

2. オプション: ログメッセージを検査するには、次のファイルを表示します。
 - **/var/log/foreman/production.log**
 - **/var/log/foreman-proxy.log**

関連情報

ジャーナルの詳細は、[Red Hat Enterprise Linux 8 Configuring Basic System Settings Guideの Viewing logs using the command line](#) を参照してください。

16.4. 個別のロガーを使用した選択的デバッグ

ロガーを個別に有効にして、選択的にデバッグを実行できます。

手順

1. 必要なロガーを個別に有効にします。たとえば、**sql** と **ldap** のロガーを有効にするには、以下のコマンドを入力します。

```
# satellite-installer \
--foreman-loggers ldap:true \
--foreman-loggers sql:true
```

2. オプション: すべてのロガーをデフォルト値にリセットします。

```
# satellite-installer --reset-foreman-loggers
```

16.5. 個別のロガーの概要

次のリストは、選択したロガーの概要を示しています。すべてのロガーを含むリストと、そのデフォルト値は、**Foreman::Logging.add_loggers** の `/usr/share/foreman/config/application.rb` で表示されます。

app

Web 要求とアプリケーションの一般的なメッセージをすべてロギングします。
デフォルト値: true

audit

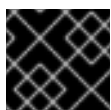
追加のファクト統計、追加、更新、削除されたファクトの数をロギングします。
デフォルト値: true

background

バックグラウンド処理コンポーネントからの情報をログに記録します。

blob

監査目的で、レンダリングされたテンプレートの内容をログに記録します。



重要

blob ロガーには機密データが含まれている可能性があります。

dynflow

Dynflow プロセスからの情報をログに記録します。

ldap

ハイレベルの LDAP クエリーと LDAP オペレーションをロギングします。

デフォルト値: false

notifications

通知コンポーネントからの情報をログに記録します。

permissions

ページを読み込む時にユーザーのロール、フィルター、パーミッションへのクエリーをロギングします。

デフォルト値: false

sql

Rails ActiveRecord を使用した SQL クエリーをロギングします。

デフォルト値: false

telemetry

Telemetry からのデバッグ情報をログに記録します。

templates

テンプレートレンダラーコンポーネントからの情報をログに記録します。

16.6. サービスのステータスの取得

Satellite は、一連のバックエンドサービスを使用します。トラブルシューティングを行うときに、Satellite サービスのステータスを確認できます。

手順

- Satellite Web UI で、**Administer > About** に移動します。
 - **Smart Proxies** タブで、すべての Capsule のステータスを表示します。
 - **Compute Resources** タブで、接続されているコンピュータリソースプロバイダーのステータスを表示します。
 - **Backend System Status** テーブルで、すべてのバックエンドサービスのステータスを表示します。

CLI 手順

- データベースと Satellite サービスから情報を取得します。

```
# hammer ping
```

- systemd で実行されているサービスのステータスを確認します。

```
# satellite-maintain service status
```

詳細は、**satellite-maintain service --help** を実行して確認してください。

- ヘルスチェックを実行します。

```
$ satellite-maintain health check
```

詳細は、**satellite-maintain health --help** を実行して確認してください。

16.7. サービスの再起動:

Satellite は、一連のバックエンドサービスを使用します。トラブルシューティングを行う場合、必要に応じてサービスを再起動できます。

手順

- Satellite サービスを再起動します。

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

16.8. ログ情報を処理するためのユーティリティー

ログファイルから情報を収集して、Satellite のトラブルシューティングを行うことができます。

sosreport

sosreport コマンドは、実行中のカーネルバージョン、ロードされたモジュール、実行中のサービス、システムおよびサービス設定ファイルなど、Linux システムから設定および診断情報を収集します。この出力は、`/var/tmp/sosreport-XXX-20171002230919.tar.xz` にある tar ファイルに保存されます。

詳細については、**sosreport --help** を実行するか、[What is an sosreport and how to create one in Red Hat Enterprise Linux?](#) を参照してください。



重要

sosreport コマンドは、情報を収集する際にパスワード、トークン、キーなどのセキュリティ情報を削除します。ただし、tar ファイルには依然として Satellite Server についての機密情報が含まれる可能性があります。tar ファイルは、公開ターゲットではなく、対象とする受信者に直接送信してください。

foreman-tail

foreman-tail コマンドは、Satellite のログをリアルタイムで表示します。

詳細は、**foreman-tail(8)** の man ページを参照してください。

16.9. ログファイルディレクトリー

次の表は、Satellite Server 上の選択されたログディレクトリーの概要を示しています。

ログファイルのディレクトリー	ログファイルの内容の説明
<code>/var/log/candlepin/</code>	サブスクリプションの管理
<code>/var/log/foreman-installer/</code>	インストーラー
<code>/var/log/foreman-maintain/</code>	Foreman の維持

ログファイルのディレクトリー	ログファイルの内容の説明
<code>/var/log/foreman-proxy/</code>	Foreman プロキシ
<code>/var/log/foreman/</code>	Foreman
<code>/var/log/httpd/</code>	Apache HTTP サーバー
<code>/var/log/messages/</code>	その他の各種ログメッセージ
<code>/var/log/puppetlabs/puppet/</code>	Puppet
<code>/var/log/tomcat/</code>	Candlepin Web サービスのログ

16.10. システムジャーナルメタデータ

次の表に、Satellite で **journal** サービスが使用するメタデータを示します。このメタデータを使用して、クエリーをフィルタリングできます。

AUDIT_ACTION

実行された監査アクション
例: 作成、更新、または削除

AUDIT_TYPE

監査リソースの種類
例: ホスト、サブネット、または ContentView

AUDIT_ID

数字としての監査リソースデータベース ID

AUDIT_ATTRIBUTE

監査リソースフィールドまたは更新されたデータベース列

AUDIT_FIELD_OLD

更新アクションの古い監査値

AUDIT_FIELD_NEW

更新アクションの新しい監査値

AUDIT_ID

監査対象のレコードデータベース ID

AUDIT_ATTRIBUTE

アクションが実行された属性名または列
例: 名前または説明

EXCEPTION_MESSAGE

エラーがログに記録される場合の例外メッセージ

EXCEPTION_CLASS

エラーがログに記録される場合の例外 Ruby クラス

EXCEPTION_BACKTRACE

エラーがログに記録される場合の、複数行の文字列としての例外バックトレース

LOC_ID

ロケーションデータベース ID

LOC_NAME

ロケーション名

LOC_LABEL

ロケーションラベル

LOGGER

ロガー名

デフォルトで有効になっているロガーの現在のリストを表示するには、次のコマンドを入力します。

```
# awk '/add_loggers/,/^$/' /usr/share/foreman/config/application.rb
```

ORG_ID

組織データベース ID

ORG_NAME

組織名

ORG_LABEL

組織ラベル

REMOTE_IP

クライアントのリモート IP アドレス

REQUEST

Action Dispatch モジュールによって生成されたリクエスト ID

SESSION

セッションごと、またはセッションレスリクエストのリクエストごとに生成されるランダム ID

TEMPLATE_NAME

テンプレート名

TEMPLATE_DIGEST

レンダリングされたテンプレートコンテンツのダイジェスト (SHA256)

TEMPLATE_HOST_NAME

レンダリングされたテンプレートのホスト名 (存在する場合)

TEMPLATE_HOST_ID

レンダリングされたテンプレートのホストデータベース ID (存在する場合)

USER_LOGIN

ユーザーのログイン名

第17章 リソースの監視

この章では、管理対象システムの監視とレポートを設定する方法について詳しく説明します。これには、ホストの設定やコンテンツビュー、コンプライアンス、サブスクリプションと現在登録されているホスト、プロモーションおよび同期が含まれます。

17.1. 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には、ホストの設定の概要やコンテンツビュー、コンプライアンスレポート、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、さらに最新の通知リストなどを提供する各種ウィジェットが含まれています。

Satellite web UI で、**Monitor > Dashboard** 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。ダッシュボードは、各ウィジェットをクリックして別の位置にドラッグすることで、配置を変更することができます。以下のウィジェットが利用できます。

ホスト設定の状態

最後のレポートング期間におけるホストの設定状態およびそれに該当するホスト数。以下の表では、各設定状態を説明しています。

表17.1 ホスト設定の状態

アイコン	状態	説明
	変更をエラーなく実行したホスト	最後のレポートング期間に変更が正常に実行されたホスト。
	エラー状態のホスト	最後のレポートング期間にエラーが検出されたホスト。
	直近 35 分間での良好なホストレポート	直近の 35 分間で変更を行わず、エラーがないホスト。
	保留中の変更があるホスト	いくつかのリソースが適用されているものの、Puppet が noop モードで実行されるように設定されたホスト。
	同期していないホスト	同期がされておらず、最後のレポートング期間にレポートが受信されていないホスト。
	レポートのないホスト	最後のレポートング期間にレポートが受信されていないホスト。
	警告が無効にされているホスト	監視対象外のホスト。

設定状態のいずれかをクリックすると、該当するホストが表示されます。

ホスト設定チャート

ホスト状態の割合と該当するホストのパーセンテージを示す円グラフです。

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リストです。

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監視します。

実行分布 (直近 30 分)

デフォルトでは 30 分となっている直近の Puppet 間隔中の実行中 Puppet エージェントの分布状況を示すグラフです。このケースでは、各コラムで 3 分間にクライアントから受け取ったレポート数を示しています。

新規ホスト

最近作成されたホストリストです。ホストをクリックすると、詳細が表示されます。

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サマリーです。タスク番号をクリックすると、対応するタスクのリストが表示されます。

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リストです。タスクをクリックして詳細を確認してください。

検出されたホスト

検出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リストです。

最新のエラータ

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リストです。

コンテンツビュー

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リストです。


同期の概要

Satellite で有効にされているすべての製品またはリポジトリおよびそれらの同期の状態の概要です。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リスト表示されます。

ホスト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17.2 ホストのサブスクリプションの状態

アイコン	状態	説明
	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。

アイコン	状態	説明
	部分使用	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、これらのホストを監視する必要があります。
	有効	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ホストを表示します。

サブスクリプションのステータス

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。

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リストで、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。

Virt-who 設定の状態

環境内のホスト上で稼働している **virt-who** デーモンから受け取ったレポートの状態。以下の状態があります。

表17.3 Virt-who 設定状態

状態	説明
レポートなし	virt-who 設定デプロイメント中にエラーが発生したか、設定がデプロイされていないか、もしくは予定された期間に virt-who が Satellite に接続できないか、いずれかのためにレポートが受信されていません。
変更なし	ハイパーバイザーが仮想マシン上で変更を検出していない、または virt-who が予定された期間中にレポートのアップロードに失敗したために、レポートが受信されていません。仮想マシンを追加したものの、設定が 変更なし 状態にある場合は、その virt-who が実行中か確認してください。
OK	予定期間中にエラーなしでレポートが受信されました。
設定合計数	virt-who 設定の合計数。

各状態の設定を表示するには、その設定状態をクリックします。

このウィジェットでは、**変更のない最新の設定**にある **変更なし** で最新の 3 つの設定もリスト表示されます。

最新のコンプライアンスレポート

最新のコンプライアンスレポートリスト。各コンプライアンスレポートでは、パス (P)、不合格 (F)、その他 (O) のルール数が表示されます。ホストをクリックすると、コンプライアンスレポートの詳細が表示されます。ポリシーをクリックすると、その詳細が表示されます。

コンプライアンスレポートの内訳

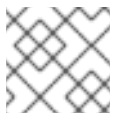
コンプライアンスレポートの状態の分布を示す円グラフです。

Red Hat Insights アクション

Red Hat Insights は Satellite に組み込まれたツールで、環境をチェックし、実行可能なアクションを提案します。アクションは、可用性、安定性、パフォーマンス、セキュリティの 4 つに分けられます。

Red Hat Insights リスクサマリー

リスクレベルに応じたアクションの分布を示す表です。リスクレベルは、アクションの重要性和問題を発生させる可能性を示しています。リスクレベルには、低、中、高、重大があります。



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。

17.1.1. タスクの管理

Red Hat Satellite は、同期されたリポジトリ、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、**監視 > タスク** に移動します。

タスクウィンドウでは、特定のタスクを検索し、そのステータス、詳細、およびタスクが開始してから経過時間を表示できます。1 つ以上のタスクをキャンセルして再開することもできます。

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。

タイムアウト設定を調整するには、以下を実行します。

1. Satellite Web UI で、**Administer > Settings** に移動します。
2. 検索ボックスに `%_timeout` と入力し、**Search** をクリックします。検索では、説明を含む 4 つの設定が返されます。
3. **Value** 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. 希望の値を秒単位で入力し、**Save** 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` 値の編集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チェック機能で問題が発生する場合は、以下の方法で無効にできます。

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. Satellite Web UI で、**Administer > Settings** に移動します。
2. 検索ボックスに `check_services_before_actions` と入力し、**Search** をクリックします。
3. **Value** 列でアイコンをクリックして値を編集します。
4. ドロップダウンメニューから **false** を選択します。
5. **Save** をクリックします。

17.2. RSS 通知の設定

Satellite のイベント通知アラートを表示するには、画面右上の **通知** アイコンをクリックします。

デフォルトでは、通知エリアには [Red Hat Satellite Blog](#) で公開された RSS フィードイベントが表示されます。

フィードは12時間ごとにリフレッシュされ、新規イベントが利用可能となるたびに通知エリアがリフレッシュされます。

URL フィードを変更することで RSS フィード通知を設定できます。サポートされるフィード形式は、RSS 2.0 と Atom です。RSS 2.0 フィード設定の例については、[Red Hat Satellite Blog feed](#) を参照してください。Atom フィード設定の例については、[Foreman blog feed](#) を参照してください。

RSS フィード通知の設定方法

1. Satellite web UI で、**Administer > Settings** に移動し、**Notifications** タブを選択します。
2. RSS URL の行で、**Value** 列の編集アイコンをクリックし、必要な URL を入力します。
3. RSS 有効化の行で、**Value** 列の編集アイコンをクリックし、この機能を有効または無効にします。

17.3. SATELLITE SERVER の監視

監査レコードには、Satellite ですべてのユーザーが行った変更が一覧表示されます。この情報は、メンテナンスとトラブルシューティングに使用できます。

手順

1. Satellite Web UI で、**Monitor > Audits** に移動して、監査レコードを表示します。
2. すべての監査属性のリストを取得するには、次のコマンドを使用します。

```
# foreman-rake audits:list_attributes
```

17.4. CAPSULE SERVER の監視

以下の項では、Satellite Web UI を使用して、保守とトラブルシューティングに役に立つ Capsule 情報を見つける方法について説明します。

17.4.1. 一般的な Capsule 情報の表示

Satellite web UI で、**Infrastructure > Capsules** に移動して、Satellite Server に登録されている Capsule サーバーのテーブルを表示します。表に含まれる情報には以下の質問に対する回答が含まれます。

Capsule Server は稼働していますか？

これは、**ステータス** コラムで緑色のアイコンにより示されます。赤色のアイコンは、非アクティブな Capsule を示します。その Capsule をアクティベートするには、Capsule Server で **service foreman-proxy restart** コマンドを使用します。

Capsule Server で有効なサービスはどれですか？

機能 コラムで、Capsule がたとえば DHCP サービスを提供するかどうか、また Pulp ミラーとして動作するかどうかを確認できます。Capsule の機能はインストール中に有効にしたり、後で設定したりできます。詳細は、Capsule Server のインストールを参照してください。

Capsule Server はどの組織およびロケーションに割り当てられていますか？

Capsule Server は複数の組織およびロケーションに割り当てることができますが、現在選択された組織に属する Capsule のみが表示されます。すべての Capsule をリストするには、左上隅にあるコンテキストメニューから **任意の組織** を選択します。

Capsule 設定の変更後に、**アクション** コラムのドロップダウンメニューから **リフレッシュ** を選択して Capsule の表を最新状態にしてください。

詳細情報を表示するには Capsule 名をクリックします。**概要** タブでは、Capsule の表にある情報と同じものを見つけることができます。さらに、以下の質問に回答することができます。

Capsule Server が管理するホストはどれですか？

関連するホストの数は **マネージドホスト** ラベルの横に表示されます。関連するホストの詳細を表示するには、その数をクリックします。

Capsule Server で利用可能なストレージ容量はどれくらいですか？

/var/lib/pulp で Pulp コンテンツが使用しているストレージ容量が表示されます。また、Capsule で利用可能な残りのストレージ容量を確認できます。

17.4.2. サービスの監視

Satellite web UI で、**Infrastructure > Capsules** に移動し、選択した Capsule の名前をクリックします。**サービス** タブでは、DNS ドメインのリストや Pulp ワーカーの数などの、Capsule サービスに関する基本的な情報を見つけることができます。ページの外観は、Capsule Server で有効なサービスによって異なります。より詳細なステータス情報を提供するサービスには Capsule ページで専用のタブが用意されることがあります。詳細は、[「Puppet の監視」](#)を参照してください。

17.4.3. Puppet の監視

Satellite web UI で、**Infrastructure > Capsules** に移動し、選択した Capsule の名前をクリックします。**Puppet** タブでは、以下を確認できます。

- **全般** サブタブで、Puppet イベントのサマリー、最新の Puppet 実行の概要、関連するホストの同期ステータス。
- **環境** サブタブで、Puppet 環境のリスト。

Puppet CA タブでは、以下の情報を確認できます。

- **全般** サブタブで、証明書ステータスの概要と自動署名エントリーの数。

- **証明書** サブタブで、Capsule に関連する CA 証明書の表。ここでは、証明書失効データを調べたり、**取り消し** をクリックして証明書をキャンセルしたりすることができます。
- **エントリーの自動署名** サブタブで、自動署名エントリーのリスト。ここでは、**新規** をクリックしてエントリーを作成したり、**削除** をクリックしてエントリーを削除したりできます。



注記

Puppet タブと **Puppet CA** タブは、Satellite で Puppet を有効にしている場合にのみ使用できます。詳細は、Red Hat Satellite で Puppet インテグレーションを使用して設定を管理する の Satellite で Puppet インテグレーションを有効にする を参照してください。

第18章 WEBHOOK の使用

Webhook は、Web ページまたは Web アプリケーションが他のアプリケーションにリアルタイムで情報を提供するための方法です。Webhook は、イベントが発生した後にのみトリガーされます。リクエストには通常、イベントの詳細が含まれています。イベントは、ホストがプロビジョニングされたことを確認する電子メールの送信などの呼出しをトリガーします。Webhook を使用すると、fire-and-forget メッセージ交換パターンを使用して、Satellite 内部イベントに基づいて外部 API への呼び出しを定義できます。リクエストを送信するアプリケーションは、レスポンスを待たないか、無視します。

Webhook のペイロードは、Webhook テンプレートから作成されます。Webhook テンプレートは、プロビジョニングテンプレートと同じ ERB 構文を使用します。使用可能な変数:

- **@event_name**: イベントの名前。
- **@webhook_id**: 一意のイベント ID。
- **@payload**: ペイロードデータ。イベントタイプごとに異なります。個々のフィールドにアクセスするには、**@payload[:key_name]** Ruby ハッシュ構文を使用します。
- **@payload[:object]**: データベースアクション (作成、更新、削除) によってトリガーされるイベントのデータベースオブジェクト。カスタムイベントには使用できません。
- **@payload[:context]**: リクエストやセッションの UUID、リモート IP アドレス、ユーザー、組織、ロケーションなどのハッシュなどの追加情報。

Webhook は HTTP を使用するため、既存の Web サービスに新しいインフラストラクチャーを追加する必要はありません。

Satellite での Webhook の一般的な使用例は、ホストが作成または削除されたときにモニタリングシステムを呼び出すことです。

Webhook は、外部システムで実行するアクションを API を介して実行できる場合に役立ちます。追加のコマンドを実行したりファイルを編集したりする必要がある場合は、Capsules 用の shellhooks プラグインを利用できます。shellhooks プラグインを使用すると、API を介して実行できる Capsule 上のシェルスクリプトを定義できます。

shellhooks プラグインをインストールしなくても、webhook を正常に使用できます。

利用可能なイベントのリストについては、[利用可能な Webhook イベント](#) を参照してください。

18.1. WEBHOOK への移行

従来の **foreman_hooks** プラグインは、webhooks プラグインが意図的に提供していないモデルオブジェクトへのフルアクセスを提供していました。

利用可能な範囲はセーフモードによって制限され、すべてのオブジェクトとマクロは両方とも API の安定性の約束の対象であり、完全に文書化されています。

webhook によってトリガーされるイベントの数は、**foreman_hooks** の場合よりも大幅に少なくなります。

Webhook は非同期で処理されるため、システムの内部が改ざんされるリスクは最小限に抑えられます。個々の Webhook スクリプトごとにペイロードを作成せずに、**foreman_hooks** から移行することはできません。ただし、webhook プラグインにはいくつかのサンプルペイロードテンプレートが付属しています。シェルフックでサンプルペイロードを使用して、移行を簡素化することもできます。

同様の結果を得るには、スクリプトテンプレートとペイロードテンプレートの両方をカスタマイズする必要があります。

18.2. WEBHOOK のインストール

次の手順を使用して、Webhook をインストールします。Webhook をインストールした後、Webhook リクエストを送信するように Satellite Server を設定できます。

手順

- 次のコマンドを使用して Webhook をインストールします。

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

- オプション: Webhook CLI プラグインをインストールします。

```
# satellite-installer --enable-foreman-cli-webhooks
```

18.3. WEBHOOK テンプレートの作成

Webhook テンプレートは、Webhook のトリガー時に、設定されたターゲットに対して HTTP リクエストの本文を生成するために使用されます。次の手順を使用して、Satellite web UI で Webhook テンプレートを作成します。

手順

1. Satellite Web UI で、**Administer > Webhook Templates** に移動します。
2. **Clone an existing template** または **Create Template** をクリックします。
3. テンプレートの名前を入力します。
4. エディターを使用して、テンプレートペイロードに変更を加えます。
Webhook HTTP ペイロードは、Satellite テンプレート構文を使用して作成する必要があります。Webhook テンプレートは、イベントのメインオブジェクトを表すことができる **@object** と呼ばれる特別な変数を使用できます。特定のイベントの場合、**@object** が欠落する可能性があります。**@payload** 変数を使用して、実際にどのデータが利用可能であるかを判断できます。

詳細は、**ホストの管理** の [テンプレート作成リファレンス](#) を参照してください。使用可能なテンプレートマクロとメソッドについては、Satellite Server の [/templates_doc](#) を参照してください。
5. オプション: 説明と監査コメントを入力します。
6. 組織とロケーションを割り当てます。
7. **Submit** をクリックします。

例

Webhook テンプレートを作成するときは、テンプレートの対象となるターゲットアプリケーションの形式に従う必要があります。たとえば、アプリケーションでは、Webhook メッセージを含む **テキスト** フィールドが想定できます。Webhook テンプレートの形式の詳細は、対象アプリケーションのドキュメントを参照してください。

リモート実行ジョブの実行

この Webhook テンプレートは、リモート実行ジョブの ID と結果を含むメッセージを定義します。このテンプレートを使用する Webhook は、**Actions Remote Execution Run Host Job Succeeded** または **Actions Remote Execution Run Host Job Failed** などのイベントにサブスクライブできます。

```
{
  "text": "job invocation <%= @object.job_invocation_id %> finished with result <%= @object.task.result %>"
}
```

ユーザーの作成

この Webhook テンプレートは、作成されたユーザーのログインとメールを含むメッセージを定義します。このテンプレートを使用する Webhook は、**User Created** イベントにサブスクライブする必要があります。

```
{
  "text": "user with login <%= @object.login %> and email <%= @object.mail %> created"
}
```

18.4. WEBHOOK の作成

Satellite web UI を使用して、イベント、ペイロード、HTTP 認証、コンテンツタイプ、およびヘッダーをカスタマイズできます。

次の手順を使用して、SatelliteWebUI で Webhook を作成します。

手順

1. Satellite Web UI で、**Administer > Webhooks** に移動します。
2. **Create new** をクリックします。
3. **Subscribe to** リストからイベントを選択します。
4. Webhook の **Name** を入力します。
5. **Target URL** を入力します。Webhook は、事前設定された URL に対して HTTP リクエストを行います。ターゲット URL は動的 URL にすることができます。
6. テンプレートをクリックして **Template** を選択します。Webhook テンプレートは、Webhook のトリガー時に Satellite Server に対して HTTP リクエストの本文を生成するために使用されます。
7. HTTP メソッドを入力します。
8. オプション: Webhook の作成時にアクティブ化しない場合は、**Enabled** フラグのチェックを外します。
9. **Credentials** タブをクリックします。
10. オプション: HTTP 認証が必要な場合は、**User** および **Password** を入力します。

11. オプション: システム証明書ストアまたは Satellite CA に対してサーバー証明書を検証しない場合は、**Verify SSL** のチェックを外します。
12. **Additional** タブで、**HTTP Content Type** を入力します。たとえば、定義したペイロードの **application/json**、**application/xml**、**text/plain** などです。アプリケーションは、指定されたコンテンツタイプに一致するようにコンテンツを変換しようとはしません。
13. オプション: HTTP ヘッダーを JSON として提供します。ERB も許可されます。

非標準の HTTP または HTTPS ポートを持つエンドポイントで Webhook を設定する場合、SELinux ポートを割り当てる必要があります。[オンラインネットワーク環境での Satellite Server のインストールの カスタムポートで Satellite にアクセスできるように SELinux を設定する](#) を参照してください。

18.5. 利用可能な WEBHOOK イベント

次の表には、Satellite web UI から利用できる Webhook イベントのリストが含まれています。**Action** イベントは **success** した場合にのみ Webhook をトリガーするため、アクションが失敗した場合、Webhook はトリガーされません。

ペイロードの詳細については、**Administer > About > Support > Templates DSL** アクセスしてください。使用可能なタイプのリストを次の表に示します。一部のイベントは **custom** としてマークされます。その場合、ペイロードはオブジェクトオブジェクトですが、Ruby ハッシュ (Key-Value データ構造) であるため、構文が異なります。

イベント名	説明	ペイロード
Actions Katello Content View Promote Succeeded	コンテンツビューが正常にプロモートされました。	Actions::Katello::ContentView::Promote
Actions Katello Content View Publish Succeeded	リポジトリは正常に同期されました。	Actions::Katello::ContentView::Publish
Actions Remote Execution Run Host Job Succeeded	ホストの汎用リモート実行ジョブが成功しました。このイベントは、完了すると、すべてのリモート実行ジョブに対して発行されます。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Errata Install Succeeded	Katello インターフェイスを使用してエラータをインストールします。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Group Install Succeeded	Katello インターフェイスを使用してパッケージグループをインストールします。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Install Succeeded	Katello インターフェイスを使用してパッケージをインストールします。	Actions::RemoteExecution::RunHostJob

イベント名	説明	ペイロード
Actions Remote Execution Run Host Job Katello Group Remove	Katello インターフェイスを使用してパッケージグループを削除します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Remove Succeeded	Katello インターフェイスを使用してパッケージを削除します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Service Restart Succeeded	Katello インターフェイスを使用してサービスを再起動します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Group Update Succeeded	Katello インターフェイスを使用してパッケージグループを更新します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Katello Package Update Succeeded	Katello インターフェイスを使用してパッケージを更新します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Foreman OpenSCAP Run Scans Succeeded	OpenSCAP スキャンを実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Host Succeeded	ホストに定義されたすべてのロールを含む Ansible Playbook を実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Capsule Upgrade Succeeded	特定の Capsule Server ホストで Capsule をアップグレードします。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Configure Cloud Connector Succeeded	特定のホストで Cloud Connector を設定します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Insights Plan Succeeded	ID を指定して Red Hat から指定されたメンテナンスプランを実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Run Playbook Succeeded	特定のホストに対して Ansible Playbook を実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Ansible Enable Web Console Succeeded	Ansible Playbook を実行して、特定のホストで Web コンソールを有効にします。	Actions::RemoteExecution::RunHostJob

イベント名	説明	ペイロード
Actions Remote Execution Run Host Job Puppet Run Host Succeeded	Puppet を 1 回実行します。	Actions::RemoteExecution::RunHostJob
アクションリモート実行ホストジョブの実行 Katello モジュールストリームアクションを成功させる	Katello インターフェイスを使用してモジュールストリームアクションを実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Pre-upgrade Succeeded	RHEL7 ホストのアップグレード可能性チェック。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Remediation Plan Succeeded	Leapp で修復計画を実行します。	Actions::RemoteExecution::RunHostJob
Actions Remote Execution Run Host Job Leapp Upgrade Succeeded	RHEL7 ホストの Leapp アップグレードジョブを実行します。	Actions::RemoteExecution::RunHostJob
Build Entered	ホストがビルドモードに入りました。	カスタムイベント: @payload[:id] (ホスト ID)、 @payload[:hostname] (ホスト名)。
Build Exited	ホストビルドモードがキャンセルされました。正常にプロビジョニングされたか、ユーザーが手動でビルドをキャンセルしました。	カスタムイベント: @payload[:id] (ホスト ID)、 @payload[:hostname] (ホスト名)。
Content View Created/Updated/Destroyed	コンテンツビューでの一般的なデータベース操作。	Katello::ContentView
Domain Created/Updated/Destroyed	ドメインでの一般的なデータベース操作。	ドメイン
Host Created/Updated/Destroyed	ホストでの一般的なデータベース操作。	ホスト
Hostgroup Created/Updated/Destroyed	ホストグループでの一般的なデータベース操作。	Hostgroup
Model Created/Updated/Destroyed	モデルに対する一般的なデータベース操作。	モデル

イベント名	説明	ペイロード
Status Changed	ホストのグローバルホストステータスが変更されました。	カスタムイベント: @payload[:id] (ホスト ID)、 @payload[:hostname] 、 @payload[:global_status] (ハッシュ)
Subnet Created/Updated/Destroyed	サブネットでの一般的なデータベース操作。	サブネット
Template Render Performed	レポートテンプレートがレンダリングされました。	テンプレート
User Created/Updated/Destroyed	ユーザーに対する一般的なデータベース操作。	User

18.6. シェルフック

Webhook では、1つの Satellite イベントを1つの API 呼び出しにのみマッピングできます。単一のシェルスクリプトに複数のコマンドを含めることができる高度な統合の場合、REST HTTP API を使用して実行可能ファイルを公開する Capsule shellhooks プラグインをインストールできます。

その後、Capsule API に接続して事前定義されたシェルフックを実行するように Webhook を設定できます。シェルフックは、実行可能であればどの言語でも記述できる実行可能なスクリプトです。たとえば、シェルフックにはコマンドや編集ファイルを含めることができます。

実行可能スクリプトは、名前に英数字とアンダースコアのみを使用して **/var/lib/foreman-proxy/shellhooks** に配置する必要があります。

Webhook ペイロードを通じてシェルフックスクリプトに入力を渡すことができます。この入力はシェルフックスクリプトの標準入力にリダイレクトされます。**X-Shellhook-Arg-1** ~ **X-Shellhook-Arg-99** 形式の HTTP ヘッダーを使用して、シェルフックスクリプトに引数を渡すことができます。シェルフックスクリプトに引数を渡す方法は、次を参照してください。

- [「Webhook を使用したシェルフックスクリプトへの引数の指定」](#)
- [「Curl を使用したシェルフックスクリプトへの引数の指定」](#)

HTTP メソッドは POST である必要があります。URL の例は、**https://capsule.example.com:9090/shellhook/My_Script** のようになります。



注記

shellhooks ディレクトリーとは異なり、URL を有効にするには、URL に単数形の **/shellhook/** が含まれている必要があります。

呼び出しを承認できるようにするには、シェルフックに接続されている Webhook ごとに **Capsule Authorization** を有効にする必要があります。

標準出力と標準エラー出力は、それぞれデバッグレベルまたは警告レベルのメッセージとして Capsule ログにリダイレクトされます。

シェルフックの HTTPS 呼び出しは値を返しません。

シェルフックスクリプトの作成例については、「[引数を出力するためのシェルフックの作成](#)」を参照してください。

18.7. SHELLHOOKS プラグインのインストール

オプションで、次のコマンドを使用して、シェルフックに使用される各 Capsule にシェルフックプラグインをインストールして有効にすることができます。

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

18.8. WEBHOOK を使用したシェルフックスクリプトへの引数の指定

Webhook を使用してシェルフックスクリプトに引数を渡すには、以下の手順を使用します。

手順

- Webhook を作成するときは、**Additional** タブで、次の形式の HTTP ヘッダーを作成します。

```
{
  "X-Shellhook-Arg-1": "VALUE",
  "X-Shellhook-Arg-2": "VALUE"
}
```

ヘッダーが有効な JSON または ERB 形式であることを確認してください。データベース ID、名前、ラベルなど、改行や引用符を含まない安全なフィールドのみを渡します。

詳細は、「[Webhook の作成](#)」を参照してください。

例

```
{
  "X-Shellhook-Arg-1": "<%= @object.content_view_version_id %>",
  "X-Shellhook-Arg-2": "<%= @object.content_view_name %>"
}
```

18.9. CURL を使用したシェルフックスクリプトへの引数の指定

curl でシェルフックスクリプト引数を渡すには、以下の手順を使用します。

手順

- curl** を使用してシェルフックスクリプトを実行する場合は、次の形式で HTTP ヘッダーを作成します。

```
"X-Shellhook-Arg-1: VALUE"
"X-Shellhook-Arg-2: VALUE"
```

例

```
# curl -sX POST -H 'Content-Type: text/plain' \
```

```
-H "X-Shellhook-Arg-1: Version 1.0" \
-H "X-Shellhook-Arg-2: My Content View" \
--data "" https://capsule.example.com:9090/shellhook/My_Script
```

18.10. 引数を入力するためのシェルフックの作成

リモート実行ジョブの実行時に、Hello World! を出力する単純なシェルフックスクリプトを作成します。

前提条件

- **Webhook** と **Shellhook** プラグインがインストールされている。詳細は以下を参照してください。
 - [「Webhook のインストール」](#)
 - [「Shellhooks プラグインのインストール」](#)

手順

1. `/var/lib/foreman-proxy/shellhooks/print_args` スクリプトを変更して、引数を標準エラーに出力し、Capsule ログで確認できるようにします。

```
#!/bin/sh
#
# Prints all arguments to stderr
#
echo "$@" >&2
```

2. Satellite Web UI で、**Administer** > **Webhooks** に移動します。
3. **Create new** をクリックします。
4. **Subscribe to** リストから、**Actions Remote Execution Run Host Job Succeeded** を選択します。
5. Webhook の **Name** を入力します。
6. **Target URL** フィールドに、Capsule Server の URL に続けて **:9090/shellhook/print_args** と入力します。

```
https://capsule.example.com:9090/shellhook/print_args
```

Shellhooks ディレクトリーとは異なり、URL 内の **Shellhook** は単数形であることに注意してください。

7. **Template** リストから **Empty Payload** を選択します。
8. **Credentials** タブで、**Capsule Authorization** をチェックします。
9. **Additional** タブの **Optional HTTP headers** フィールドに次のテキストを入力します。

```
{
  "X-Shellhook-Arg-1": "Hello",
  "X-Shellhook-Arg-2": "World!"
}
```



```
    }
```

10. **Submit** をクリックします。これで、Hello World! を出力するシェルフックが正常に作成され、リモート実行ジョブが成功するたびに、Capsule ログに記録されます。

検証

1. 任意のホストでリモート実行ジョブを実行します。**time** をコマンドとして使用できます。詳細は、[ホストの管理](#) の リモートジョブの実行 を参照してください。
2. シェルフックスクリプトがトリガーされ、Hello World! が Capsule Server ログへ出力されたことを確認します。

```
# tail /var/log/foreman-proxy/proxy.log
```

ログの最後に次の行が表示されているはずです。

```
[I] Started POST /shellhook/print_args
[I] Finished POST /shellhook/print_args with 200 (0.33 ms)
[I] [3520] Started task /var/lib/foreman-proxy/shellhooks/print_args\ Hello\ World!
[W] [3520] Hello World!
```

第19章 検索およびブックマーク機能

Satellite は、Satellite web UI のほとんどのページで強力な検索機能を備えています。この機能によって Satellite 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。

19.1. 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リストが表示されます。一覧からオプションを選択するか、補完機能を使用してクエリーを構築するか、入力を継続するか、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。

19.1.1. 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。

すべての演算子のリストについては、[Operators](#) を参照してください。値の形式についての説明は、[Values](#) を参照してください。

19.1.2. クエリー Operator

パラメーターと値の間で利用できるすべての演算子を次の表に示します。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19.1 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) 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。	hostgroup = RHEL7
!=	NOT EQUALS		
~	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 * 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 検索結果では、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が返されます。
<	LESS THAN		registered_at <= Yesterday 検索結果では、昨日以前に登録されたホストが返されます。
^	IN	SQL の場合と同様に、値のリスト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 or set?		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 または set? hostgroup Puppet クラスページでは、検索結果は 1 つ以上のホストグループに割り当てられたクラスを返します。
NOT HAS or null?			not has hostgroup または null? hostgroup ホストの概要を表示するダッシュボードで、検索結果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19.2 検索で利用できる論理演算子

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

19.1.3. クエリー値

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。囲まないと、空白文字は 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とも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。

時間の値

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。

19.2. 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ーは引用符で囲む必要があります。囲まないと、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ーを使用することが推奨されます。

19.3. ブックマークの管理

検索クエリーをブックマークとして保存し、再利用することもできます。ブックマークは削除したり、変更したりすることもできます。

ブックマークは、作成されたページでのみ表示されます。一部のページには、すべての **active** または **disabled** ホストなど、一般的な検索で使用可能なデフォルトのブックマークがあります。

19.3.1. ブックマークの作成

本セクションでは、検索クエリーをブックマークとして保存する方法について説明します。関連ページ用の検索クエリーのブックマークは、作成したそのページで保存する必要があります。たとえば、ホスト関連の検索クエリーは、ホストページで保存します。

手順

1. Satellite web UI で、ブックマークを作成するページに移動します。
2. **検索** フィールドに保存する検索クエリーを入力します。
3. **検索** ボタンの右側にある矢印を選択し、**この検索をブックマーク** を選択します。
4. **名前** フィールドに 新規ブックマークの名前を入力します。
5. **検索クエリー** フィールドに正しい検索クエリーがあることを確認します。
6. **公開** チェックボックスの設定を確認します。
 - **公開** にチェックを入れると、ブックマークが公開されて全ユーザーに見えるようになります。
 - **公開** のチェックを外すと、ブックマークが非公開となり、作成したユーザーのみに見えるようになります。
7. **Submit** をクリックします。

作成されたことを確認するには、**検索** ボタンの右側にある矢印を選択してブックマークリストを表示するか、**管理 > ブックマーク** に移動してから、ブックマークの名前を **ブックマーク** リストで確認します。

19.3.2. ブックマークの削除

ブックマークは、ブックマークページで削除できます。

手順

1. Satellite web UI で、**Administer > Bookmarks** に移動します。
2. ブックマークページで、削除するブックマークの **削除** をクリックします。
3. 確認ウィンドウが表示されたら、**OK** をクリックして削除を確認します。

削除されたことを確認するには、ブックマークの名前がないことを **ブックマーク** リストで確認します。

付録A 管理設定

このセクションには、Satellite Web UI で **Administer > Settings** に移動して編集できる設定に関する情報が含まれています。

A.1. 一般設定

設定	デフォルト値	説明
管理者のメールアドレス		デフォルトの管理者の電子メールアドレス
Satellite URL		Satellite インスタンスに到達可能な URL。 Provisioning > Unattended URL も参照してください。
1ページあたりのエントリー	20	Satellite のページごとに表示されるレコードの数
DB キャッシュの修正	No	Satellite は、パーミッションとロールのキャッシュを保持します。これを Yes に設定すると、Satellite は、次回再起動時にこのキャッシュを再作成します。
DB 保留シード	No	foreman-rake db:seed は、インストーラーモジュールの次回の実行時に実行する必要がありますか？
Capsule リクエストのタイムアウト	60	Satellite から Capsule への HTTP リクエストのオープンおよび読み取りタイムアウト (秒単位)。
ログインページのフッターテキスト		ログインページのフッターに表示されるテキスト。
HTTP(S) プロキシ		Satellite 製品からの発信 HTTP(S) 接続のプロキシを設定します。システム全体のプロキシは、オペレーティングシステムレベルで設定する必要があります。
ホストを除く HTTP(S) プロキシ	[]	リクエストのプロキシとならないホスト名を設定します。ローカルホストへのリクエストはデフォルトで除外されます。
Experimental Labs を表示する	No	Experimental Labs の機能にアクセスするためのメニューを表示するかどうか (ページのリロードが必要)。

設定	デフォルト値	説明
ホストにドメイン名を追加する	あり	Yes に設定すると、Satellite は新しいホストがプロビジョニングされるときにドメイン名を追加します。
同期間隔外	30	マネージドホストは定期的にレポートを作成し、レポート間の時間がこの期間 (分単位) を超えると、ホストは同期していないと見なされます。ホストごとに、 Hosts > All hosts > \$host > Edit > Parameters > Add Parameter の追加で outofsync_interval パラメーターを追加することにより、ホストでこれをオーバーライドできます。
Satellite UUID		Satellite インスタンス ID。Satellite インスタンスを一意に識別します。
デフォルトの言語		新規ユーザーの UI はこの言語を使用します。
デフォルトのタイムゾーン		新規ユーザーに使用するタイムゾーン。
インスタンスのタイトル		インスタンスのタイトルは上部のナビゲーションバーに表示されます (ページの再読み込みが必要です)。
保存された監査間隔		監査データを保持するための日数。監査のクリーンアップを無効にするには、空のままにします。
新しいホストの詳細 UI	Yes	Satellite は、ホストの詳細について新しい UI をロードします。

A.2. SATELLITE タスク設定

設定	デフォルト値	説明
同期タスクのタイムアウト	120	例外が発生する前に同期タスクが終了するのを待機する秒数。
dynflow コンソールを有効にする	Yes	デバッグ用に dynflow コンソール (/foreman_tasks/dynflow) を有効にします。

設定	デフォルト値	説明
dynflow コンソールに認証が必要	Yes	ユーザーは、dynflow コンソールにアクセスする前に、管理者権限を持っていることを認証する必要があります。
Capsule アクションの再試行回数	4	失敗する前に Capsule でタスクを開始するために許可された試行回数。
Capsule アクションの再試行間隔	15	再試行間の秒単位の時間。
Capsule バッチタスクを許可する	Yes	Capsule でタスクのバッチトリガーを有効にします。
Capsule タスクのバッチサイズ	100	foreman_tasks_proxy_batch_trigger が有効になっている場合に、Capsule への1つのリクエストに含まれるタスクの数。
タスクのトラブルシューティング URL		タスクのトラブルシューティングドキュメントを指す URL。正規化されたタスクラベル (英数字のみに制限) に置き換えられる %{label} プレースホルダーが含まれている必要があります。 %{version} プレースホルダーも利用できます。
ポーリング間隔乗数	1	デフォルトのポーリング間隔を乗算するために使用されるポーリング乗数。これを使用して、長時間実行されるタスクで頻繁にポーリングするのを防ぐことができます。

A.3. テンプレート同期設定

設定	デフォルト値	説明
関連付け	New	テンプレートを OS、組織、ロケーションに関連付けます。
ブランチ		Git リポジトリのデフォルトのブランチ。
メッセージのコミット	Satellite ユーザーが作成したテンプレートのエクスポート	エクスポートされたテンプレートのカスタムコミットメッセージ。

設定	デフォルト値	説明
ディレクトリー名	/	テンプレートを含む Git リポジトリー内のディレクトリー。
Filter		この正規表現に一致する名前のインポートまたはエクスポート。大文字小文字を区別しません。スニペットはフィルタリングされません。
強制インポート	No	Yes に設定すると、ロックされたテンプレートはインポート中に上書きされます。
テンプレートのロック	保持し、新しいものをロックしないでください	インポートされたテンプレートのロックを処理する方法。
メタデータエクスポートモード	更新	<p>デフォルトのメタデータエクスポートモード。</p> <p>可能なオプション:</p> <p>refresh により、メタデータが再レンダリングされます。</p> <p>keep は既存のメタデータを保持します。</p> <p>remove は、メタデータなしでテンプレートをエクスポートします。</p>
否定	No	インポートまたはエクスポート用にフィルターを無効にします。
接頭辞		インポートされたテンプレートに接頭辞として追加された文字列。

設定	デフォルト値	説明
リポジトリ		<p>テンプレートをインポートまたはエクスポートする場所からのターゲットパス。たとえば、さまざまなプロトコルを使用できます。</p> <p><code>/tmp/dir</code></p> <p><code>git://example.com</code></p> <p><code>https://example.com</code></p> <p><code>ssh://example.com</code></p> <p><code>/tmp</code> にエクスポートする場合、本番環境のデプロイメントは private tmp を使用するように設定されている可能性があることに注意してください。</p>
詳細度	No	テンプレートをインポートする Rake タスクの冗長性を選択します。

A.4. 検出された設定

設定	デフォルト値	説明
ディスカバリーロケーション		検出されたホストを配置するデフォルトのロケーションを示します。
ディスカバリー組織		検出されたホストが追加されるデフォルトの組織を示します。
インターフェイスのファクト	discovery_bootif	プライマリーインターフェイスの検出に使用するファクト名。
ボンドインターフェイスを作成する	No	LLDP を使用して同じ VLAN で別のインターフェイスが検出された場合、ボンドインターフェイスを自動的に作成します。
すべてのファクトをクリーンアップする	No	プロビジョニング中に、報告されたすべてのファクト (検出ファクトを除く) をクリーンアップします。
ホスト名のファクト	discovery_bootif	ホスト名に使用するファクトのリスト (コンマ区切り、最初の勝利)。

設定	デフォルト値	説明
自動プロビジョニング	No	プロビジョニングルールを使用して、新しく検出されたホストを自動的にプロビジョニングします。
再起動	Yes	プロビジョニング中に自動的に再起動するか、kexec が検出したホスト。
ホスト名接頭辞	mac	ホスト名に使用するデフォルトの接頭辞。文字で始める必要があります。
ファクト列		ホストリストに表示する追加のファクト列 (コンマ区切り)。
ハイライトされたファクト		ハイライトセクションのファクトを整理するための正規表現 - 例: <code>^(abc cde)\$</code> 。
ストレージのファクト		ストレージセクションのファクトを整理するための正規表現。
ソフトウェアのファクト		ソフトウェアセクションのファクトを整理するための正規表現。
ハードウェアのファクト		ハードウェアセクションのファクトを整理するための正規表現。
ネットワークのファクト		ネットワークセクションのファクトを整理するための正規表現。
IPMI のファクト		インテリジェントプラットフォーム管理インターフェイス (IPMI) セクションのファクトを整理するための正規表現。
PXE をロックする	No	Preboot Execution Environment (PXE) 設定を自動的に生成して、新しく検出されたホストを検出に固定します。
ロックされた PXELinux テンプレート名	pxelinux_discovery	ホストを検出に固定するときに使用される PXELinux テンプレート。
ロックされた PXEGrub テンプレート名	pxegrub_discovery	ホストを検出に固定するときに使用される PXEGrub テンプレート。
ロックされた PXEGrub2 テンプレート名	pxegrub2_discovery	ホストを検出に固定するときに使用される PXEGrub2 テンプレート。
DNS を強制する	Yes	検出されたホストをプロビジョニングするときに、DNS エントリーの作成を強制します。

設定	デフォルト値	説明
既存の NIC で発生したエラー	いいえ	プロビジョニング ネットワーク インターフェイスカード (NIC) の MAC に一致する既存のマネージドホストの検出を許可しないでください (早期にエラーが発生します)。
名前ジェネレーターのタイプ	ファクト + 接頭辞	検出ホスト名の命名パターン。
Prefer IPv6	No	検出されたノードを呼び出すときは、IPv4 よりも IPv6 を優先します。

A.5. ブートディスク設定

設定	デフォルト値	説明
iPXE ディレクトリー	<code>/usr/share/ipxe</code>	iPXE イメージを含むディレクトリーへのパス。
ISOLINUX ディレクトリー	<code>/usr/share/syslinux</code>	ISOLINUX イメージを含むディレクトリーへのパス。
SYSLINUX ディレクトリー	<code>/usr/share/syslinux</code>	SYSLINUX イメージを含むディレクトリーへのパス。
Grub2 ディレクトリー	<code>/var/lib/tftpboot/grub2</code>	grubx64.efi および shimx64.efi を含むディレクトリーへのパス。
ホストイメージテンプレート	ブートディスク iPXE- ホスト	ホスト固有のブートディスクに使用する iPXE テンプレート。
一般的なイメージテンプレート	ブートディスク iPXE- 汎用ホスト	汎用ホストブートディスクに使用する iPXE テンプレート。
汎用 Grub2 EFI イメージテンプレート	ブートディスク Grub2 EFI- 汎用ホスト	汎用 Extensible Firmware Interface (EFI) ホストブートディスクに使用する Grub2 テンプレート。
ISO 生成コマンド	<code>genisoimage</code>	ISO イメージを生成するコマンド genisoimage または mkisofs を使用します。

設定	デフォルト値	説明
インストールメディアのキャッシュ	Yes	インストールメディアファイルは、完全なホストイメージ用にキャッシュされます。
許可されるブートディスクの種類	[generic, host, full_host, subnet]	許可されているブートディスクタイプのリスト。無効にするには空白のままにします。

A.6. RED HAT クラウド設定

設定	デフォルト値	説明
自動インベントリアップロード	Yes	ホストインベントリの Red Hat クラウドへの自動アップロードを有効にします。
推奨事項を自動的に同期する	No	RedHat クラウドからの Insights 推奨事項の自動同期を有効にします。
ホスト名の難読化	No	RedHat クラウドに送信されたホスト名を難読化します。
ホストの ipv4 アドレスの難読化	No	RedHat クラウドに送信された IPv4 アドレスを難読化します。
RHC デーモンの ID	*****	RHC デーモン ID。

A.7. コンテンツ設定

設定	デフォルト値	説明
デフォルトの HTTP プロキシ		コンテンツを同期するためのデフォルトの HTTP プロキシ。
CDN SSL バージョン		CDN との通信に使用される SSL バージョン。
デフォルトの同期 OS プロビジョニングテンプレート	Kickstart default	同期されたコンテンツから作成されたオペレーティングシステムのデフォルトのプロビジョニングテンプレート。

設定	デフォルト値	説明
デフォルトの同期 OS 終了テンプレート	キックスタートデフォルトフィニッシュ	同期されたコンテンツから作成された新しいオペレーティングシステムのデフォルトの仕上げテンプレート。
デフォルトの同期された OS ユーザーデータ	デフォルトのユーザーデータをキックスタート	同期されたコンテンツから作成された新しいオペレーティングシステムのデフォルトのユーザーデータ。
デフォルトの同期 OS PXELinux テンプレート	デフォルトの PXELinux をキックスタート	同期されたコンテンツから作成された新しいオペレーティングシステム用のデフォルトの PXELinux テンプレート。
デフォルトの同期 OS PXEGrub テンプレート	デフォルトの PXEGrub をキックスタートします	同期されたコンテンツから作成された新しいオペレーティングシステム用のデフォルトの PXEGrub テンプレート。
デフォルトの同期 OS PXEGrub2 テンプレート	デフォルトの PXEGrub2 をキックスタートします	同期されたコンテンツから作成された新しいオペレーティングシステム用のデフォルトの PXEGrub2 テンプレート。
デフォルトの同期された OS iPXE テンプレート	キックスタートのデフォルトの iPXE	同期されたコンテンツから作成された新しいオペレーティングシステム用のデフォルトの iPXE テンプレート。
デフォルトの同期 OS パーティションテーブル	Kickstart default	同期されたコンテンツから作成された新しいオペレーティングシステムのデフォルトのパーティションテーブル。
デフォルトの同期 OS kexec テンプレート	Discovery Red Hat kexec	同期されたコンテンツから作成された新しいオペレーティングシステムのデフォルトの kexec テンプレート。
デフォルトの同期された OS アトミックテンプレート	アトミックキックスタートのデフォルト	同期されたコンテンツから作成された新しいアトミックオペレーティングシステムのデフォルトのプロビジョニングテンプレート。
マニフェストリフレッシュタイムアウト	1200	マニフェストを更新するときのタイムアウト (秒単位)。
アクションタイムアウトを受け入れる	20	ホストがリモートアクションを取得するのを待機する秒単位の時間。
アクションタイムアウトの終了	3600	ホストがリモートアクションを終了するのを待つ秒単位の時間。

設定	デフォルト値	説明
サブスクリプション接続が有効	Yes	サブスクリプションのために Red Hat ポータルと通信できます。
コンテンツビューからインストール可能なエラータ	いいえ	ホストのコンテンツビューおよびライフサイクル環境のエラータのみに基づいてエラータホストのステータスを計算します。
複合コンテンツビューのプロモーションを制限する	No	これが有効になっている場合は、複合コンテンツビューに含まれるコンテンツビューのバージョンがターゲット環境に存在しない限り、複合コンテンツビューを公開またはプロモートすることはできません。
アクションの前にサービスを確認してください	Yes	アクションを実行する前に、pulp や candlepin などのバックエンドサービスのステータスを確認しますか？
リポジトリを同期するためのバッチサイズ	100	Capsule で同時に同期する必要があるリポジトリの数。数値が小さいほど、同期時間が長くなる可能性があります。数値を大きくすると、dynflow の負荷が増加します。
コンテンツビュープロモーション後に Capsules を同期する	あり	コンテンツビューのプロモーション後に Capsules を自動同期するかどうか。
デフォルトのカスタムリポジトリダウンロードポリシー	immediate	カスタムリポジトリのデフォルトのダウンロードポリシー。 immediate または on_demand のいずれか。
デフォルトの Red Hat リポジトリダウンロードポリシー	on_demand	有効な Red Hat リポジトリのデフォルトのダウンロードポリシー。 immediate または on_demand のいずれか。
デフォルトの Capsule ダウンロードポリシー	on_demand	Capsule 同期のデフォルトのダウンロードポリシー。 inherit 、 immediate 、または on_demand のいずれか。
Pulp のエクスポート先ファイルパス	/var/lib/pulp/katello-export	エクスポートされたリポジトリのディスク上のロケーション。
Pulp 3 のエクスポート先ファイルパス	/var/lib/pulp/exports	Pulp 3 のエクスポートされたリポジトリのディスク上のロケーション。

設定	デフォルト値	説明
Pulp クライアントキー	<code>/etc/pki/katello/private/pulp-client.key</code>	Pulp サーバー認証に使用される SSL キーのパス。
Pulp クライアント証明書	<code>/etc/pki/katello/certs/pulp-client.crt</code>	Pulp サーバー認証に使用される SSL 証明書のパス。
同期接続タイムアウト	300	同期時の接続の合計タイムアウト (秒単位)。
デフォルトでリモート実行を使用	いいえ	有効にすると、リモートアクションに katello-agent の代わりにリモート実行が使用されます。
登録解除時にホストを削除	No	subscription-manager を使用してホストの登録を解除する場合は、ホストレコードも削除してください。仮想マシンや DNS レコードなど、ホストにリンクされているマネージドリソースも削除される可能性があります。
サブスクリプションマネージャー名登録ファクト		subscription-manager を使用してホストを登録する場合は、ホスト名に指定されたファクトを強制的に使用します (fact.fact の形式で)。
サブスクリプションマネージャー名登録ファクト strict マッチ	No	これが有効で、 register_hostname_fact が設定および提供されている場合、登録はそのファクトのみを使用して名前で新しいホストを検索し、一致するすべてのホスト名をスキップします。
デフォルトのロケーションサブスクライブされたホスト	デフォルトのロケーション	登録後に新しいサブスクライブされたホストが保存されるデフォルトのロケーション。
すぐに期限切れ	120	サブスクリプションの更新について通知されるまでのサブスクリプションの残り日数。
コンテンツビューの依存関係を解決するデフォルト	No	新しいコンテンツビューのデフォルトの依存関係解決値。

設定	デフォルト値	説明
重複する DMIUUID をホストする	[]	デスクトップ管理インターフェイス (DMI)UUID が重複しているためにホストが登録に失敗した場合は、ここにコンマ区切りの値を追加します。後続の登録により、影響を受けるホストに対して一意の DMIUUID が生成されます。
ホストプロファイルを想定	Yes	登録 DMIUUID が別のホストによって使用されていない限り、新しいホスト登録を有効にして、ホスト名が一致する登録済みプロファイルを想定します。
ビルドでホストプロファイルを変更できる	No	ホストがビルドモードである限り、ホストプロファイルをバイパスするようにホスト登録を有効にします。
ホストはビルドでのみ再登録できます	No	ホストがビルドモードの場合にのみ、ホストが再登録できるようにします。
ホストタスクワーカーのプールサイズ	5	ホスト関連タスクの実行を処理するためのプール内のワーカーの数。0 に設定すると、デフォルトのキューが使用されます。dynflowd/foreman-tasks サービスの再起動が必要です。
適用性バッチサイズ	50	タスクごとに処理するホスト適用性計算の数。
Autosearch	Yes	それをサポートするページの場合、検索入力を入力しながら自動的に検索を実行します。
自動検索の遅延	500	自動検索が有効になっている場合は、入力中に検索を実行する前にミリ秒単位で遅延します。
Pulp バルクロードサイズ	2000	1 ページの PulpAPI 呼び出しからフェッチされたアイテムの数。
Dynflow なしでプロファイルをアップロードする	Yes	Katello を有効にして、ホストがインストールしたパッケージ、有効なリポジトリ、およびモジュールインベントリを、Dynflow タスクでラップするのではなく、直接更新できるようにします (Puma プロセスが大量のメモリーを使用している場合はオフにしてみてください)。

設定	デフォルト値	説明
孤立したコンテンツの保護時間	1440	孤立したコンテンツを孤立していると思われる時間 (分単位)。
リモート実行のために Capsule を介して登録することを推奨します	No	リモート実行を使用する場合は、ホストが登録されているプロキシを使用することを推奨します。
公開されたコンテンツビューのリポジトリの削除を許可する	あり	ユーザーが以前に 1 つ以上のコンテンツビューバージョンで公開したリポジトリの削除を有効にします。

A.8. 認証設定

設定	デフォルト値	説明
OAuth がアクティブ	Yes	Satellite は API 認証に OAuth を使用します。
OAuth コンシューマーキー	*****	OAuth コンシューマーキー。
OAuth コンシューマーシークレット	*****	OAuth コンシューマーシークレット。
OAuth マップユーザー	No	Satellite は、リクエストヘッダーのユーザー名でユーザーをマッピングします。これが無効になっている場合、OAuth リクエストには管理者権限があります。
ログイン試行の失敗制限	30	Satellite は、指定された回数のログイン試行の失敗後 5 分間、受信 IP アドレスからのユーザーログインをブロックします。ブルートフォース保護を無効にするには、0 に設定します。
登録された Capsules を制限する	Yes	既知の Capsules のみが、Capsules 認証を使用する機能にアクセスできます。
Capsules には SSL が必要	Yes	クライアント SSL 証明書は、Capsules を識別するために使用されます (:require_ssl も有効にする必要があります)。
信頼できるホスト	[]	ファクト/レポートインポーターおよび ENC 出力にアクセスするための Capsules に加えて、信頼されるホスト名、IPv4、IPv6 アドレス、またはサブネットのリスト。

設定	デフォルト値	説明
SSL 証明書	<code>/etc/foreman/client_cert.pem</code>	Satellite がプロキシとの通信に使用する SSL 証明書パス。
SSL CA ファイル	<code>/etc/foreman/proxy_ca.pem</code>	Satellite がプロキシとの通信に使用する SSLCA ファイルパス。
SSL 秘密鍵	<code>/etc/foreman/client_key.pem</code>	Satellite がプロキシとの通信に使用する SSL 秘密鍵パス。
SSL クライアント DN 環境	<code>HTTP_SSL_CLIENT_S_DN</code>	クライアント SSL 証明書からのサブジェクト DN を含む環境変数。
SSL クライアント検証環境	<code>HTTP_SSL_CLIENT_VERIFY</code>	クライアント SSL 証明書の検証ステータスを含む環境変数。
SSL クライアント証明書環境	<code>HTTP_SSL_CLIENT_CERT</code>	クライアントの SSL 証明書を含む環境変数。
サーバー CA ファイル		Satellite への接続を確認するためにテンプレートで使用する SSLCA ファイルパス。
Websocket SSL キー	<code>etc/pki/katello/private/katello-apache.key</code>	Satellite が WebSocket を暗号化するために使用する秘密鍵ファイルのパス。
Websocket SSL 証明書	<code>/etc/pki/katello/certs/katello-apache.crt</code>	Satellite が WebSocket を暗号化するために使用する証明書パス。
WebSocket の暗号化	Yes	VNC/SPICE WebSocket プロキシコンソールアクセス暗号化 (websockets_ssl_key/cert 設定が必要)。
ログイン委譲ログアウト URL		ログアウト時にユーザーをこの URL にリダイレクトします。 ログイン委譲の承認 も有効にします。
ログイン委譲認証ソースユーザーの自動作成を承認します	External	不明な外部認証ユーザー (ログイン委譲の承認 を参照) が作成される外部認証ソースの名前。空は自動作成がないことを意味します。
ログイン委譲を承認する	No	REMOTE_USER HTTP ヘッダーを使用してログイン委譲を承認します。
ログイン委譲 API を承認する	No	API 呼び出しにも REMOTE_USERHTTP ヘッダーを使用してログイン委譲を承認します。

設定	デフォルト値	説明
アイドルタイムアウト	60	指定した分数が経過したら、アイドル状態のユーザーをログアウトします。
BCrypt パスワードのコスト	9	内部認証ソースの bcrypt パスワードハッシュ関数のコスト値 (4-30)。値が大きいほど安全ですが、特にステートレス API 呼び出しと UI ログインの場合、検証は遅くなります。既存のパスワードに影響を与えるには、パスワードの変更が必要です。
BMC クレデンシャルアクセス	Yes	ENCYAML 出力およびテンプレートでの BMC インターフェイスパスワードへのアクセスを許可します。
OIDC JWKs URL		OpenID Connect JSON Web Key Set (JWKS) の URL。Keycloak を OpenID プロバイダーとして使用する場合、通常は <code>https://keycloak.example.com/auth/realms/<realm name>/protocol/openid-connect/certs</code> 。
OIDC Audience	[]	認証に使用されている OpenIDConnect Audience の名前。Keycloak の場合、これはクライアント ID です。
OIDC Issuer		issuer claim は、 JSON Web トークン (JWT) を発行したプリンシパルを識別します。これは、ほとんどの OpenID プロバイダーの場合は <code>/.well-known/openid-configuration</code> に存在します。
OIDC Algorithm		OpenID プロバイダーで JWT をエンコードするために使用されるアルゴリズム。

A.9. メール設定

設定	デフォルト値	説明
メール返信アドレス		Satellite が送信している電子メールの電子メール返信アドレス。
メールの件名の接頭辞		すべての送信メールに追加する接頭辞。

設定	デフォルト値	説明
ウェルカムメールを送信する	No	新しいユーザーにユーザー名と URL を含むウェルカムメールを送信します。
配信方法	Sendmail	電子メールの配信に使用される方法。
SMTP enable StartTLS auto	Yes	SMTP は自動的に StartTLS を有効にします。
SMTP OpenSSL 検証モード	デフォルトの検証モード	TLS を使用する場合、OpenSSL が証明書をチェックする方法を設定できます。
SMTP アドレス		接続する SMTP アドレス。
SMTP ポート	25	接続する SMTP ポート。
SMTP HELO/EHLO ドメイン		HELO/EHLO ドメイン。
SMTP ユーザー名		必要に応じて、認証に使用するユーザー名。
SMTP パスワード	*****	必要に応じて、認証に使用するパスワード。
SMTP 認証	none	必要に応じて、認証タイプを指定します。
Sendmail の引数	-i	sendmail に追加のオプションを指定します。配信方法が sendmail に設定されている場合にのみ使用されます。
Sendmail のロケーション	/usr/sbin/sendmail	sendmail 実行可能ファイルのロケーション。配信方法が sendmail に設定されている場合にのみ使用されます。

A.10. 通知設定

設定	デフォルト値	説明
RSS を有効にする	Yes	RSS 通知をプルします。
RSS URL	https://www.redhat.com/en/rss/blog/channel/red-hat-satellite	RSS 通知を取得する URL。

A.11. プロビジョニング設定

設定	デフォルト値	説明
ホストオーナー		空の Satellite が現在のユーザーを使用する場合、プロビジョニングされたホストのデフォルトの所有者。
root パスワード	*****	プロビジョニングされたホストのデフォルトの暗号化されたルートパスワード。
無人 URL		ホストがビルド中にテンプレートを取得する URL。https、無人、または userdata で始まる場合、HTTP を使用してコントローラーにアクセスすることはできません。
Safemode レンダリング	Yes	<p>プロビジョニングテンプレートのセーフモードレンダリングを有効にします。デフォルトの推奨オプションである Yes は、Satellite にリストされていない変数およびオブジェクトへのアクセスを拒否します。</p> <p>No に設定すると、テンプレート機能を使用するパーミッションがあるユーザーは、テンプレートやパラメーター、スマート変数のいずれかを編集することで、いかなるオブジェクトにもアクセスすることが可能になります。こうなると、ユーザーは Satellite Server で完全なリモートコード実行が可能になり、すべての認証が無効になります。これは、特に大企業では安全なオプションではありません。</p>
ビルドせずに無人でアクセス	No	ビルドモードを使用せずに無人 URL へのアクセスを有効にします。
ローカルネームサーバーのクエリー	No	Satellite は、SOA/NS 権限ではなく、ローカルに設定されたリゾルバーにクエリーを実行します。
インストールトークンの有効期間	360	インストールトークンが有効である必要がある時間(分単位)。トークンを無効にするには、0 に設定します。
SSH タイムアウト	120	SSH プロビジョニングがタイムアウトするまでの秒単位の時間。
Libvirt のデフォルトのコンソールアドレス	0.0.0.0	libvirt を使用して新しい仮想マシンをプロビジョニングするときに、コンソールのリッスンアドレスに使用する必要がある IP アドレス。

設定	デフォルト値	説明
ビルドされたリクエストから IP を更新	No	Satellite は、ビルド要求を行った IP でホスト IP を更新します。
VM には短い名前を使用します	No	Satellite は、新しい仮想マシンを作成するために FQDN の代わりに短いホスト名を使用します。
DNS タイムアウト	[5, 10, 15, 20]	dns_lookup マクロや DNS レコードの競合検証などの DNS ルックアップ試行のタイムアウト (秒単位) のリスト。
失敗したデプロイメントのクリーンアップ	Yes	プロビジョニングスクリプトがゼロ以外の終了コードで終了した場合、Satellite は仮想マシンを削除します。
名前ジェネレーターのタイプ	ランダムベース	<p>新規ホスト作成時のホスト名の生成方法を指定します。</p> <p>デフォルトの Random-based オプションでは、使用可能ではあるものの必須ではない、一意のランダムなホスト名を生成します。多くのホストを作成し、命名方法がわからないユーザーには便利です。</p> <p>MAC-based オプションは、ベアメタルのホストのみになります。ホストを削除してから、後で作成すると、MAC アドレスをベースにした同じホスト名が付けられます。サーバーを再利用し、常に同じホスト名にする場合に便利です。</p> <p>Off オプションでは、名前生成関数が無効になり、ホスト名フィールドは空白になります。</p>
デフォルトの PXE グローバルテンプレートエントリ		グローバルテンプレートのデフォルトの PXE メニュー項目 - local 、 discovery 、またはカスタム。テンプレートのデフォルトには空白を使用します。
デフォルトの PXE ローカルテンプレートエントリ		ローカルテンプレートのデフォルトの PXE メニュー項目 - local 、 local_chain_hd0 、またはカスタム、テンプレートのデフォルトには空白を使用します。
iPXE 中間スクリプト	iPXE 中間スクリプト	無人インストール用の中間 iPXE スクリプト。

設定	デフォルト値	説明
ホストの削除で関連付けられた VM を破棄します	No	ホストの削除時に関連する VM を破棄します。有効にすると、ホストにリンクされている VM がコンピューティングリソースから削除されます。無効にすると、ホストが削除されたときに VM のリンクが解除されます。つまり、VM はコンピュートリソースに残り、再関連付けしたり、Satellite に再度インポートしたりできます。これにより、VM の電源が自動的にオフになることはありません
最大の構造化されたファクト	100	構造化サブツリー内のキーの最大数、 satellite::dropped_subtree_facts に保存されている統計。
デフォルトのグローバル登録テンプレート	グローバル登録	グローバル登録テンプレート。
デフォルトのホスト初期設定テンプレート	Linuxhost_init_config のデフォルト	デフォルトのホスト初期設定テンプレート。新しいオペレーティングシステムが作成されると自動的に割り当てられます。
グローバルデフォルト PXEGrub2 テンプレート	PXEGrub2 グローバルデフォルト	グローバルデフォルト PXEGrub2 テンプレート。このテンプレートは、設定されているすべての TFTP サーバーにデプロイメントされます。アップグレードの影響を受けません。
グローバルデフォルト PXELinux テンプレート	PXELinux グローバルデフォルト	グローバルデフォルト PXELinux テンプレート。このテンプレートは、設定されているすべての TFTP サーバーにデプロイメントされます。アップグレードの影響を受けません。
グローバルデフォルト PXEGrub テンプレート	PXEGrub グローバルデフォルト	グローバルデフォルト PXEGrub テンプレート。このテンプレートは、設定されているすべての TFTP サーバーにデプロイメントされます。アップグレードの影響を受けません。
グローバルデフォルト iPXE テンプレート	iPXE グローバルデフォルト	グローバルなデフォルトの iPXE テンプレート。このテンプレートは、設定されているすべての TFTP サーバーにデプロイメントされます。アップグレードの影響を受けません。

設定	デフォルト値	説明
ローカルブート PXEGrub2 テンプレート	PXEGrub2 のデフォルトのローカルブート	ローカルブートのデフォルトの PXEGrub2 として選択されているテンプレート。
ローカルブート PXELinux テンプレート	PXELinux のデフォルトのローカルブート	ローカルブートの PXELinux デフォルトとして選択されているテンプレート。
ローカルブート PXEGrub テンプレート	PXEGrub のデフォルトのローカルブート	ローカルブートの PXEGrub デフォルトとして選択されているテンプレート。
ローカルブート iPXE テンプレート	iPXE のデフォルトのローカルブート	ローカルブートの iPXE デフォルトとして選択されているテンプレート。
PuppetCA を管理する	Yes	Satellite は、新しいホストの提供時に証明書の署名を自動化します。
証明書に UUID を使用する	No	Satellite は、ホスト名の代わりにランダムな UUID を証明書の署名に使用します。
サポートされていないプロビジョニングテンプレートを表示する	No	サポートされていないプロビジョニングテンプレートを表示します。有効にすると、使用可能なすべてのテンプレートが表示されます。無効にすると、RedHat でサポートされているテンプレートのみが表示されます。

A.12. ファクト設定

設定	デフォルト値	説明
ファクトがアップロードされたときに新しいホストを作成する	Yes	新しいファクトが受信されると、Satellite がホストを作成します。
ロケーションのファクト	satellite_location	Puppet の実行後に作成されたホストは、このファクトで指定されたロケーションに配置されます。
組織のファクト	satellite_organization	Puppet の実行後に作成されたホストは、このファクトで指定された組織に配置されます。このファクトの内容は、組織の完全なラベルである必要があります。
デフォルトのロケーション	デフォルトのロケーション	ロケーションファクトを送信しなかった Puppet の実行後に作成されたホストは、このロケーションに配置されます。

設定	デフォルト値	説明
デフォルトの組織	デフォルトの組織	組織ファクトを送信しなかった Puppet の実行後に作成されたホストは、この組織に配置されます。
ファクトからホストグループを更新する	Yes	Satellite は、ホストのホストグループをそのファクトから更新します。
オペレーティングシステムのファクトを無視する	No	ファクトからオペレーティングシステムを更新するのをやめます。
ドメインのファクトを無視する	No	ファクトからのドメイン値の更新を停止します。
ファクトからサブネットを更新する	None	Satellite は、ホストのサブネットをそのファクトから更新します。
プロビジョニングのためにインターフェイスファクトを無視する	No	ファクトからの IP および MAC アドレス値の更新を停止します (すべてのインターフェイスに影響します)。
一致する識別子を持つインターフェイスを無視	[lo、en*v*、usb*、vnet*、macvtap*、;vdsmdummy*、veth*、tap*、qbr*、qvb*、qvo*、qr-*、qg-*、vlinuxbr*、vovsbr*、br-int]	受信ファクトからのこれらの値に一致する識別子を持つホストネットワークインターフェイスオブジェクトの作成または更新をスキップします。*ワイルドカードを使用して、識別子をインデックスと照合できます (例: macvtap*)。無視されたインターフェイスの Raw ファクトは引き続きデータベースに保存されます。詳細は、 パターンの除外 設定を参照してください。
Satellite に保存されているファクトの除外パターン	[lo、en*v*、usb*、vnet*、macvtap*、;vdsmdummy*、veth*、tap*、qbr*、qvb*、qvo*、qr-*、qg-*、vlinuxbr*、vovsbr*、br-int、load_averages::*、memory::swap::available*、memory::swap::capacity、memory::swap::used*、memory::system::available*、memory::system::capacity、memory::system::used*、memoryfree、memoryfree_mb、swapfree、swapfree_mb、uptime_hours、uptime_days]	インポートされたすべてのタイプのファクト (Puppet、Ansible、rhsm) のパターンを除外します。これらのファクトは Satellite データベースに保存されません。*ワイルドカードを使用して、名前をインデックスと一致させることができます。たとえば、 ignore* は、ignore、ignore123、および a::ignore、さらには a::ignore123::b を除外します。

A.13. 管理設定

設定	デフォルト値	説明
レポートのアップロード時に新しいホストを作成する	Yes	レポートを受信すると、Satellite がホストを作成します。
Matchers の継承	Yes	Satellite マッチャーは、ホストグループ、組織、およびロケーションのスマートクラスパラメーターを評価するときに子に継承されます。
デフォルトパラメータールックアップパス	fqdn、hostgroup、os、do main	Satellite は、デフォルトでホストスマートクラスパラメーターをこの順序で評価します。
パラメーター内の ERB を補間します	Yes	Satellite は、ENC 出力のパラメーター値で ERB を解析します。
常に設定ステータスを表示する	No	Puppet Capsule が割り当てられていない場合でも、すべてのホストに設定ステータスが表示されます。

A.14. リモート実行設定

設定	デフォルト値	説明
Capsule へのフォールバック	No	リモート実行を使用して、ホストでプロキシを検索します。これは、ホストにサブネットがない場合、またはサブネットに実行プロキシがない場合に役立ちます。
グローバル Capsule の有効化	Yes	ホストに割り当てられたプロキシの外部でリモート実行プロキシを検索します。検索は、ホストの組織とロケーションに限定されます。
SSH ユーザー	root	SSH に使用するデフォルトのユーザー。 remote_execution_ssh_user パラメーターを設定することにより、ホストごとにオーバーライドできます。
実効ユーザー	root	スクリプトの実行に使用するデフォルトのユーザー。ユーザーが SSH ユーザーと異なる場合は、su または sudo を使用してユーザーを切り替えます。

設定	デフォルト値	説明
実効ユーザーメソッド	sudo	有効なユーザーに切り替えるために使用されるコマンド。 sudo 、 dzdo 、 su の1つ
有効なユーザーパスワード	*****	有効なユーザーパスワード。効果的なユーザーを参照してください。
ジョブテンプレートの同期	Yes	db:seed の実行時にディスクからテンプレートを同期するかどうか。
SSH ポート	22	SSH 通信に使用するポート。デフォルトのポート 22 remote_execution_ssh_port パラメーターを設定することにより、ホストごとにオーバーライドできます。
IP での接続	No	ホストインターフェイスの IP アドレスが FQDN よりも優先されるかどうか。DNS が FQDN を適切に解決していない場合に役立ちます。 remote_execution_connect_by_ip パラメーターを設定することにより、ホストごとにこれをオーバーライドできます。デュアルスタックホストの場合は、 remote_execution_connect_by_ip_prefer_ipv6 設定を検討してください。
IPv4 よりも IPv6 を優先する	No	IP アドレスを使用して接続する場合、IPv6 アドレスが優先されますか？IPv6 アドレスが設定されていない場合は、自動的に IPv4 にフォールバックします。 remote_execution_connect_by_ip_prefer_ipv6 パラメーターを設定することにより、ホストごとにこれをオーバーライドできます。デフォルトおよび互換性のために、IPv6 よりも IPv4 が優先されます。
デフォルトの SSH パスワード	*****	SSH に使用するデフォルトのパスワード。 remote_execution_ssh_password パラメーターを設定することにより、ホストごとにオーバーライドできます。

設定	デフォルト値	説明
デフォルトの SSH 鍵のパスフレーズ	*****	SSH に使用するデフォルトのキーパスフレーズ。 remote_execution_ssh_key_passphrase パラメーターを設定することにより、ホストごとにオーバーライドできます。
ワーカーのプールサイズ	5	リモート実行ジョブの実行を処理するためのプール内のワーカーの数。 dynflowd/satellite-tasks サービスの再起動が必要です。
作業ディレクトリーの消去	Yes	タスクの完了後に作業ディレクトリーを削除するかどうか。 remote_execution_cleanup_working_dirs パラメーターを設定することにより、ホストごとにこれをオーバーライドできます。
Cockpit の URL		Web コンソールボタンの Cockpit インスタンスの場所。デフォルトでは、ボタンは表示されません。
フォームジョブテンプレート	コマンドの実行 -SSH のデフォルト	ジョブ呼び出しフォームで事前に選択されているジョブテンプレートを選択します。
ジョブ呼び出しレポートテンプレート	ジョブ - 呼び出しレポートテンプレート	特定のリモート実行ジョブのレポートを生成するために使用されるレポートテンプレートを選択します。
取得時間	86400	ホストがジョブを取得しなければならない時間 (秒単位)。この制限時間内にジョブが取得されない場合、ジョブはキャンセルされます。pull-mqtt ベースのジョブにのみ適用されます。デフォルトは1日です。

A.15. ANSIBLE 設定

設定	デフォルト値	説明
秘密鍵パス		これを使用して、パスワードの代わりに Ansible が使用する SSH 秘密鍵へのパスを指定します。 ansible_ssh_private_key_file ホストパラメーターでオーバーライドします。

設定	デフォルト値	説明
connection.type	ssh	Ansible Playbook を実行するときは、デフォルトでこの接続タイプを使用します。 ansible_connection パラメーターを追加することで、ホストでこれをオーバーライドできます。
WinRM 証明書の検証	validate	Ansible Playbook を実行するときに、WinRM サーバー証明書の検証を有効または無効にします。 ansible_winrm_server_cert_validation パラメーターを追加することで、ホストでこれをオーバーライドできます。
デフォルトの詳細レベル	Disabled	Satellite は、Ansible Playbook を実行するときに、追加のデバッグ出力のためにこのレベルの冗長性を追加します。
プロビジョニング後のタイムアウト	360	ホストが完全にプロビジョニングされた後、Satellite が Ansible ロールタスク Playbook をトリガーするタイミングを設定するタイムアウト (秒単位)。これを、ホストが再起動後に準備ができるまでにかかると予想される最大時間に設定します。
Ansible レポートのタイムアウト	30	ホストがレポートする必要があるタイムアウト (分単位)。
Ansible の同期が無効	No	設定された間隔内にレポートが到着しない場合、Ansible の同期がとれていないホスト設定ステータスを無効にします。
デフォルトの Ansible インベントリレポートテンプレート	Ansible-Ansible インベントリ	Satellite はこのテンプレートを使用して、Ansible インベントリでレポートをスケジュールします。
無視する Ansible のロール	[]	Capsule からロールをインポートするときに除外するロール。予想される入力ではコンマ区切り値であり、*ワイルドカードメタ文字を使用できます。例: foo* 、 *b* 、 *bar 。
Ansible の Capsules タスクのバッチサイズ		satellite_tasks_proxy_batch_trigger が有効になっている場合に、1 回のリクエストで Capsules に送信する必要があるタスクの数。設定されている場合、Ansible ジョブの satellite_tasks_proxy_batch_size 設定をオーバーライドします。

