



Red Hat Satellite 6.11

Red Hat Satellite の管理

Red Hat Satellite の管理ガイド

Red Hat Satellite 6.11 Red Hat Satellite の管理

Red Hat Satellite の管理ガイド

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 証明書のインストール	6
1.2. SATELLITE へのログイン	6
1.3. SATELLITE WEB UI のナビゲーションタブ	7
1.4. パスワードの変更	7
1.5. 管理ユーザーパスワードのリセット	8
1.6. ログインページでのカスタムメッセージの設定	9
第2章 RED HAT SATELLITE の起動および停止	10
第3章 内部 SATELLITE データベースから外部データベースへの移行	11
3.1. 外部データベースとして POSTGRESQL を使用する際の注意点	11
3.2. 外部データベース用のホストの準備	12
3.3. POSTGRESQL のインストール	13
3.4. 外部データベースへの移行	16
第4章 ANSIBLE COLLECTIONS を使用した SATELLITE の管理	17
4.1. SATELLITE ANSIBLE モジュールのインストール	17
4.2. SATELLITE ANSIBLE モジュールの表示	17
第5章 ユーザーとロールの管理	18
5.1. ユーザー管理	18
5.2. SSH キー管理	21
5.3. PERSONAL ACCESS TOKEN の管理	22
5.4. ユーザーグループの作成と管理	23
5.5. ロールの作成および管理	24
5.6. SATELLITE で利用可能な事前定義済みロール	27
5.7. 詳細なパーミッションフィルタリング	29
第6章 メール通知	33
6.1. メール通知の設定	33
6.2. メールの配信テスト	34
6.3. 電子メール通知のテスト	34
6.4. 通知タイプ	35
6.5. ホストの電子メール通知設定の変更	35
第7章 セキュリティコンプライアンスの管理	36
7.1. SECURITY CONTENT AUTOMATION PROTOCOL	36
7.2. SATELLITE 内の SCAP コンテンツ	37
7.3. コンプライアンスポリシーのデプロイメント方法	37
7.4. コンプライアンスポリシーのデプロイメント方法の設定	37
7.5. 利用可能な SCAP コンテンツのリスト	38
7.6. SCAP コンテンツの設定	38
7.7. コンプライアンスポリシーの管理	41
7.8. コンプライアンスポリシーのデプロイ	44
7.9. オンデマンドでのセキュリティコンプライアンススキャンの実行	47
7.10. コンプライアンスの監視	47
第8章 SATELLITE SERVER および CAPSULE SERVER のバックアップ	52
8.1. バックアップサイズの予測	52
8.2. SATELLITE SERVER または CAPSULE SERVER の完全バックアップの実行	54

8.3. PULP コンテンツなしでのバックアップの実行	56
8.4. 増分バックアップの実行	57
8.5. 週次の完全バックアップ後の日次増分バックアップ例	58
8.6. オンラインバックアップの実行	58
8.7. スナップショットバックアップの実行	59
8.8. 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	60
第9章 バックアップからの SATELLITE SERVER または CAPSULE SERVER の復元	61
9.1. 完全バックアップからの復元	61
9.2. 増分バックアップからの復元	62
9.3. 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	62
第10章 SATELLITE SERVER または CAPSULE SERVER の名前の変更	64
10.1. SATELLITE SERVER の名前の変更	64
10.2. CAPSULE SERVER の名前の変更	65
第11章 SATELLITE SERVER のメンテナンス	68
11.1. 監査レコードの削除	68
11.2. 監査レコードの匿名化	68
11.3. レポートのレコード削除	68
11.4. 未使用タスクのクリーニング機能の設定	68
11.5. タスクレコードの削除	69
11.6. ID によるタスクの削除	69
11.7. 完全なディスクからのリカバリー	70
11.8. 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	71
11.9. POSTGRESQL 領域の確保	72
11.10. オンデマンドリポジトリからのスペースの再利用	72
第12章 カスタム SSL 証明書の更新	74
12.1. SATELLITE SERVER でのカスタム SSL 証明書の更新	74
12.2. CAPSULE SERVER でのカスタム SSL 証明書の更新	75
第13章 問題のログとレポート	77
13.1. デバッグロギングの有効化	77
13.2. デバッグに役立つログレベルの増加	77
13.3. サービスのステータスの取得	82
13.4. サービスの再起動:	82
13.5. 個別のロガーの有効化	83
13.6. 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	84
13.7. SATELLITE が提供するログファイルディレクトリー	84
13.8. ログ情報の収集ユーティリティー	85
13.9. システムジャーナルメタデータ	85
第14章 外部認証の設定	88
14.1. LDAP の使用	89
14.2. RED HAT IDENTITY MANAGEMENT の使用	94
14.3. ACTIVE DIRECTORY の使用	97
14.4. 外部ユーザーグループの設定	101
14.5. LDAP の外部ユーザーグループのリフレッシュ	102
14.6. RED HAT IDENTITY MANAGEMENT または AD の外部ユーザーグループの更新	103
14.7. プロビジョンされたホストの外部認証	103
14.8. RED HAT SINGLE SIGN ON 認証を使用した SATELLITE の設定	106
14.9. TOTP での RED HAT SINGLE SIGN ON 認証の設定	112
14.10. RED HAT SINGLE SIGN ON 認証の無効化	119

第15章 リソースの監視	120
15.1. RED HAT SATELLITE コンテンツダッシュボードの使用	120
15.2. RSS 通知の設定	124
15.3. SATELLITE SERVER の監視	124
15.4. CAPSULE SERVER の監視	125
第16章 WEBHOOK の使用	127
16.1. WEBHOOK への移行	127
16.2. WEBHOOK のインストール	128
16.3. WEBHOOK テンプレートの作成	128
16.4. WEBHOOK の作成	128
16.5. 利用可能な WEBHOOK イベント	129
16.6. シェルフック	132
16.7. SHELLHOOKS プラグインのインストール	133
16.8. WEBHOOK を使用したシェルフックスクリプトへの引数の指定	133
16.9. CURL を使用したシェルフックスクリプトへの引数の指定	134
16.10. 引数を入力するためのシェルフックの作成	134
第17章 検索およびブックマーク機能	136
17.1. 検索クエリーの構築	136
17.2. フリーテキスト検索の使用	138
17.3. ブックマークの管理	138
付録A 管理設定	140
A.1. 一般設定	140
A.2. SATELLITE タスク設定	141
A.3. テンプレート同期設定	142
A.4. 検出された設定	144
A.5. ブートディスク設定	146
A.6. RED HAT クラウド設定	147
A.7. コンテンツ設定	147
A.8. 認証設定	152
A.9. メール設定	154
A.10. 通知設定	155
A.11. プロビジョニング設定	155
A.12. ファクト設定	159
A.13. 管理設定	161
A.14. リモート実行設定	161
A.15. ANSIBLE 設定	163

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. [Bugzilla](#) のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** 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 証明書のインストール

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 証明書がブラウザにインストールされるまでこのブラウザを Satellite に接続できない可能性があります。以下の手順を実行して、Satellite 上でルート CA 証明書を特定し、ブラウザにインストールします。

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

- Satellite Server の完全修飾ドメイン名を特定します。

```
# hostname -f
```

- 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
```

- 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 をセキュリティー例外として追加し、設定を上書きします。この手順は、使用しているブラウザによって異なる場合があります。セキュリティー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。
- katello-server-ca.crt** を選択します。
- 証明書を認証局としてブラウザにインポートして信頼し、Web サイトを特定します。

Katello ルート CA 証明書の手動インポート

- 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
```

- ブラウザで、**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 証明書のインストール」](#) 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. 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。
コンテンツ	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。
ホスト	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。
インフラストラクチャー	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。
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。

1.4. 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. 右上にある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. **送信** ボタンをクリックして、新しいパスワードを保存します。

1.5. 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。

1.6. 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、[8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 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 Red Hat Enterprise Linux 8 サーバーまたは Red Hat Enterprise Linux 7 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。

3.1. 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となります。

- 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。

3.2. 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を最新の Red Hat Enterprise Linux 8 サーバーまたは Red Hat Enterprise Linux 7 サーバーにインストールし、外部データベースをホストします。

Red Hat Software Collections および 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

- 準備されたホストは、Satellite の [Storage Requirements](#) を満たしている必要があります。

手順

1. [Attaching the Satellite Infrastructure Subscription](#) の手順に従い、サーバーに Satellite サブスクリプションをアタッチします。
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。

- Red Hat Enterprise Linux 7 の場合:

```
# subscription-manager repos --disable '*'
# subscription-manager repos --enable=rhel-server-rhsc1-7-rpms \
--enable=rhel-7-server-rpms --enable=rhel-7-server-satellite-6.11-rpms
```

- Red Hat Enterprise Linux 8 の場合:

```
# subscription-manager repos --disable '*'
# subscription-manager repos \
--enable=satellite-6.11-for-rhel-8-x86_64-rpms \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms
```

3. Red Hat Enterprise Linux 8 で、以下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については、[Red Hat Enterprise Linux Application Streams Life Cycle](#) を参照してください。

3.3. 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。PostgreSQL は、Red Hat Enterprise Linux 8 リポジトリまたは Red Hat Enterprise Linux Server 7 リポジトリを使用してインストールできます。Satellite は PostgreSQL バージョン 12 をサポートします。

- [Red Hat Enterprise Linux 8 への PostgreSQL のインストール](#)
- [Red Hat Enterprise Linux 7 への PostgreSQL のインストール](#)

3.3.1. Red Hat Enterprise Linux 8 への PostgreSQL のインストール

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
```

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
```

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/24 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 start postgresql
# systemctl enable postgresql
```

- 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
```

- `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
```

- 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;
```

- `postgres` ユーザーをログアウトします。

```
# \q
```

- 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"
```

3.3.2. Red Hat Enterprise Linux 7 への PostgreSQL のインストール

手順

- 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。

```
# yum install rh-postgresql12-postgresql-server \
rh-postgresql12-syspaths \
rh-postgresql12-postgresql-evr
```

- 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
```

- `/var/opt/rh/rh-postgresql12/lib/pgsql/data/postgresql.conf` ファイルを編集します。

```
# vi /var/opt/rh/rh-postgresql12/lib/pgsql/data/postgresql.conf
```

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/opt/rh/rh-postgresql12/lib/pgsql/data/pg_hba.conf ファイルを編集します。

```
# vi /var/opt/rh/rh-postgresql12/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/24 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 start postgresql
# systemctl enable 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. postgres ユーザーをログアウトします。

```
# \q
```

12. 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"
```

3.4. 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 --skip-pulp-content --preserve-directory -y  
/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-host postgres.example.com \  
  --foreman-db-password Foreman_Password \  
  --foreman-db-database foreman \  
  --foreman-db-manage false \  
  --foreman-db-username foreman \  
  --katello-candlepin-db-host postgres.example.com \  
  --katello-candlepin-db-name candlepin \  
  --katello-candlepin-db-password Candlepin_Password \  
  --katello-candlepin-manage-db false \  
  --katello-candlepin-db-user candlepin \  
  --foreman-proxy-content-pulpcore-manage-postgresql false \  
  --foreman-proxy-content-pulpcore-postgresql-host postgres.example.com \  
  --foreman-proxy-content-pulpcore-postgresql-db-name pulpcore \  
  --foreman-proxy-content-pulpcore-postgresql-password Pulpcore_Password \  
  --foreman-proxy-content-pulpcore-postgresql-user pulp
```

第4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。

4.1. 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。

前提条件

- Ansible 2.9 以降のリポジトリが有効になっていて、Ansible パッケージが更新されていること。

```
# subscription-manager repos --enable rhel-7-server-ansible-2.9-rpms
# satellite-maintain packages update ansible
```

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
```

4.2. SATELLITE ANSIBLE モジュールの表示

次のディレクトリーのコンテンツをリスト表示することで、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。

```
# ls /usr/share/ansible/collections/ansible_collections/redhat/satellite/plugins/modules/
```



注記

本ガイド作成時では、**ansible-doc -l** コマンドはまだコレクションをリスト表示していません。

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5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。ユーザーの使用の詳細は、「[ユーザー管理](#)」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。

5.1. ユーザー管理

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。

5.1.1. 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**: [14章 外部認証の設定](#)の説明にある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。

5.1.2. 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
```

5.1.3. 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。

5.1.4. 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

1. 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、[ユーザーの作成](#) と [ユーザーへのロールの割り当て](#) 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。

5.2. SSH キー管理

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、[Provisioning ガイドの Deploying SSH Keys during Provisioning](#) を参照してください。

SSH キーおよびその作成方法についての詳細は、[Red Hat Enterprise Linux 7 システム管理者のガイドの SSH ベースの認証の使用](#) を参照してください。

5.2.1. 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

1. Satellite Web UI で、**Administer > Users** に移動します。
2. **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。
3. **SSH キー** タブをクリックします。
 - SSH キーの追加
 - i. 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。
 - ii. **SSH キーの追加** をクリックします。
 - iii. **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。
 - iv. **名前** フィールドに SSH キーの名前を入力します。
 - v. **Submit** をクリックします。
 - SSH キーの削除
 - i. 削除する SSH キーの列にある **削除** をクリックします。
 - ii. 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
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBHHS2KmNylYa2
7Qaa7EHp+2I99ucGStx4P77e03ZvE3yVRJEFikpoP3MJtYYfle8k
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
```

5.3. 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。

5.3.1. 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 サーバーに 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":"6.11.0","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。

5.3.2. 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 サーバーに 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"}  
}
```

5.4. ユーザーグループの作成と管理

5.4.1. 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、「[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対してマップできます。

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。

5.4.2. 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
```

5.4.3. ユーザーグループの削除

Satellite Web UI を使用して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. 警告ボックスで、**OK** をクリックしてユーザーグループを削除します。

5.5. 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ル一式が提供されます (「[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。

5.5.1. 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. **ロールの作成** 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。

5.5.2. 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。

5.5.3. 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。

5.5.4. 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。

5.5.5. 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
```

- 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。

- 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>
```

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

5.5.6. ロールの削除

Satellite Web UI を使用してロールを削除します。

手順

- Satellite Web UI で、**Administer > Roles** に移動します。
- 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。
- 警告ボックスで、**OK** をクリックしてロールを削除します。

5.6. 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が付与するパーミッションのセットを正確に表示するには、Satellite Web UI でロールを特権ユーザーとして表示します。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。

表5.1 ロールが提供するパーミッション

ロール	ロールが提供するパーミッション
Access Insights Admin	Insights のルールを追加して編集します。
Access Insights Viewer	Insight レポートを表示します。
Ansible Roles Manager	ホストおよびホストグループでのロールのプレイ。Ansible ロールを表示、破棄、およびインポートします。Ansible 変数を表示、編集、作成、破棄、およびインポートします。
Ansible Tower Inventory Reader	ファクト、ホスト、およびホストグループを表示します。
Bookmarks manager	ブックマークを作成、編集、削除します。
Boot disk access	起動ディスクをダウンロードします。
Compliance manager	SCAP コンテンツファイル、コンプライアンスポリシー、テーラリングファイルの表示、作成、編集、破棄を行います。コンプライアンスレポートを表示します。
Compliance viewer	コンプライアンスレポートを表示します。
Create ARF report	コンプライアンスレポートを作成します。
Default role	他のロールに関係なく、各ユーザーに与えられる一連のパーミッション。
Discovery Manager	検出されたホストを表示、プロビジョニング、編集、および破棄し、検出ルールを管理します。
Discovery Reader	ホストと検出ルールを表示します。
Edit hosts	ホストを表示、作成、編集、破棄、および構築します。
Edit partition tables	パーティションテーブルを表示、作成、編集、および破棄します。
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。 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。 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。
Red Hat Access Logs	ログビューアーとログを表示します。

ロール	ロールが提供するパーミッション
Remote Execution Manager	インフラストラクチャジョブを実行するパーミッションを持つロールを制御します。
Remote Execution User	ホストに対してリモート実行ジョブを実行します。
Site manager	Manager ロールの制限バージョン。
System admin	<ul style="list-style-type: none"> ● 管理 > 設定 でグローバル設定を編集します。 ● ユーザー、ユーザーグループ、ロールを参照、作成、編集、破棄します。 ● 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。そのため、このロールは信頼できるユーザーにのみ付与してください。</p>
Tasks manager	Satellite タスクを表示および編集します。
Tasks reader	Satellite タスクの表示のみが可能なロール。
Viewer	Satellite 構造、ログ、レポートおよび統計の各要素の設定を表示できる機能を提供する受動的なロール。
View hosts	ホストの表示のみが可能なロール。
Virt-who Manager	完全な virt-who パーミッションのあるロール。
Virt-who Reporter	virt-who が生成したレポートを Satellite にアップロードできます。virt-who を手動で設定して、限定的な virt-who パーミッションを持つユーザーロールが必要な場合に使用できます。
Virt-who Viewer	virt-who 設定の表示ができます。このロールでは、既存の virt-who 設定を使用した virt-who インスタンスのデプロイができます。

5.7. 詳細なパーミッションフィルタリング

5.7.1. 詳細なパーミッションフィルター

「[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。

5.7.2. 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。

5.7.3. 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。

5.7.3.1. 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
```

以下のクエリーは、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。

5.7.3.2. 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。

5.7.4. 詳細な検索に対してサポートされる演算子

表5.2 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5.3 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6章 メール通知

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。

通知をトリガーするイベントは以下のとおりです。

- ホストのビルド
- コンテンツビューのプロモーション
- ホストが報告するエラー
- リポジトリの同期

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。



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を必要な通知タイプにサブスクライブします。



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、[Installing Satellite Server from a Connected Network](#) の [Configuring Satellite Server for Outgoing Emails](#) を参照してください。

6.1. 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。

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。

6.2. 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。

6.3. 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。

6.4. 通知タイプ

Satellite では以下の通知が作成されます。

- **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。
- **ホストエラーアドバイザー**: 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のサマリーです。
- **OpenSCAP 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。
- **Puppet エラー状態**: ホストが Puppet に関連するエラーを報告した後に送信される通知です。
- **Puppet サマリー**: Puppet レポートのサマリーです。
- **エラーの同期**: リポジトリの同期後にのみ送信される通知です。これには、同期で導入された新しいエラーのサマリーが含まれます。

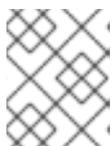
6.5. ホストの電子メール通知設定の変更

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7章 セキュリティーコンプライアンスの管理

セキュリティーコンプライアンス管理は、セキュリティーポリシーを定義し、それらのポリシーに準拠しているかどうかシステムを監査し、非準拠のインスタンスを解決する継続的なプロセスです。コンプライアンス違反は、組織の設定管理ポリシーに基づいて管理されます。セキュリティーポリシーは、ホスト固有のものから業界共通のものまでに及ぶため、ポリシー定義には柔軟性が必要になります。

Satellite を使用すると、登録されているすべてのホストに対するコンプライアンスの監査とレポートをスケジュールできます。

7.1. SECURITY CONTENT AUTOMATION PROTOCOL

Satellite は、Security Content Automation Protocol (SCAP) 標準を使用してセキュリティーポリシーを定義します。

SCAP は、XML ベースのいくつかの仕様 (Extensible Checklist Configuration description Format (XCCDF) で説明されているチェックリストや、Open Vulnerability and Assessment Language (OVAL) で説明されている脆弱性など) のフレームワークです。これらの仕様は、**データストリーム** ファイルとしてカプセル化されます。

XCCDF のチェックリスト項目 (**ルール** と呼ばれます) は、システム項目の必要な設定を表します。たとえば、ルールによって、どのユーザーも **root** ユーザーアカウントを使用して SSH 経由でホストにログインできないように指定できます。ルールは1つ以上の **XCCDF プロファイル** にグループ化できます。これにより、複数のプロファイルでルールを共有できます。

OpenSCAP スキャナーツールは、ホスト上のシステム項目をルールに照らして評価し、Asset Reporting Format (ARF) でレポートを生成します。このレポートは、モニタリングと分析のために Satellite に返されます。

表7.1 OpenSCAP スキャナーでサポートされる SCAP フレームワーク 1.3 の仕様

タイトル	説明	バージョン
SCAP	Security Content Automation Protocol	1.3
XCCDF	Extensible Configuration Checklist Description Format	1.2
OVAL	Open Vulnerability and Assessment Language	5.11
-	Asset Identification	1.1
ARF	Asset Reporting Format	1.1
CCE	Common Configuration Enumeration	5.0
CPE	Common Platform Enumeration	2.3
CVE	Common Vulnerabilities and Exposures	2.0
CVSS	Common Vulnerability Scoring System	2.0

関連情報

- SCAP の詳細は、[OpenSCAP プロジェクト](#) を参照してください。

7.2. SATELLITE 内の SCAP コンテンツ

SCAP コンテンツは、コンプライアンス、設定、またはセキュリティーベースラインの実装を含む SCAP データストリームファイルです。通常、単一のデータストリームには複数の XCCDF プロファイルが含まれます。XCCDF プロファイルは、General Purpose Operating Systems (OSPP)、Health Insurance Portability and Accountability Act (HIPAA)、PCI-DSS v3.2.1 Control Baseline for Red Hat Enterprise Linux 9 など、Satellite のホスト設定のコンプライアンスを評価するための業界標準またはカスタムのセキュリティー標準を定義したものです。テラリングファイルを使用すると、要件に応じて既存の XCCDF プロファイルを調整できます。

Satellite では、SCAP コンテンツからの XCCDF プロファイルを使用し、最終的にはテラリングファイルを使用して、コンプライアンスポリシーを定義します。Satellite には、[OpenSCAP プロジェクト](#) によって提供される SCAP セキュリティーガイドのデフォルトの SCAP コンテンツが含まれています。

独自のコンテンツをダウンロード、デプロイ、変更、作成する方法の詳細は、以下を参照してください。

- [Red Hat Enterprise Linux 9 セキュリティーの強化](#)
- [Red Hat Enterprise Linux 8 セキュリティーの強化](#)
- [Red Hat Enterprise Linux 7 セキュリティーガイド](#)
- [Red Hat Enterprise Linux 6 セキュリティーガイド](#)

7.2.1. サポートされている SCAP バージョン

Satellite は、SCAP バージョン 1.2 および 1.3 のコンテンツをサポートします。

7.3. コンプライアンスポリシーのデプロイメント方法

次のいずれかの方法を使用して、コンプライアンスポリシーをデプロイできます。

Ansible デプロイメント

Ansible ロールを使用して、コンプライアンススキャン用にホストを設定します。

Puppet デプロイメント

Puppet クラスと Puppet エージェントを使用して、コンプライアンススキャン用にホストを設定します。

手動デプロイメント

コンプライアンススキャン用にホストを手動で設定します。

7.4. コンプライアンスポリシーのデプロイメント方法の設定

次の手順のいずれかを使用して、コンプライアンスポリシーをデプロイするために選択した方法に合わせて Satellite を設定します。後で [コンプライアンスポリシーを作成](#) するときに、これらの方法のいずれかを選択します。

Ansible デプロイメントの手順

1. **theforeman.foreman_scap_client** Ansible ロールをインポートします。
詳細は、[Red Hat Satellite で Ansible を使用する設定](#)を参照してください。
2. 作成したポリシーと **theforeman.foreman_scap_client** Ansible ロールをホストまたはホストグループに割り当てます。
3. デプロイメントをトリガーするには、ホストまたはホストグループで Ansible ロールを手動で実行するか、定期的なポリシー更新のリモート実行を使用して定期的なジョブを設定します。
詳細は、[ホストの管理のリモートジョブの設定とセットアップ](#)を参照してください。

Puppet デプロイメントの手順

1. Puppet が有効になっていることを確認します。
2. Puppet エージェントがホストにインストールされていることを確認します。
3. **foreman_scap_client** Puppet モジュールを含む Puppet 環境をインポートします。
詳細は、[Red Hat Satellite で Puppet インテグレーションを使用した設定の管理](#)を参照してください。
4. 作成したポリシーと **foreman_scap_client** Puppet クラスをホストまたはホストグループに割り当てます。
Puppet は次回の定期実行時にデプロイメントをトリガーします。または、Puppet を手動で実行することもできます。Puppet はデフォルトで 30 分ごとに実行します。

手動デプロイメントの手順

- 手動デプロイメント方法の場合、追加の Satellite 設定は必要ありません。
手動デプロイメントの詳細は、[Red Hat ナレッジベースの How to set up OpenSCAP Policies using Manual Deployment option](#) を参照してください。

7.5. 利用可能な SCAP コンテンツのリスト

Satellite にすでにロードされている SCAP コンテンツを表示するには、この手順を使用してください。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#)を参照してください。

前提条件

- ユーザーアカウントに **view_scap_contents** 権限がある。

手順

- Satellite Web UI で、**Hosts > Compliance - SCAP contents** に移動します。

CLI 手順

- Satellite Server で次の Hammer コマンドを実行します。

```
# hammer scap-content list \
--location "My_Location" \
--organization "My_Organization"
```

7.6. SCAP コンテンツの設定

SCAP データストリームとテラリングファイルをアップロードして、コンプライアンスポリシーを定義できます。

7.6.1. デフォルト SCAP コンテンツのロード

デフォルト SCAP コンテンツを Satellite Server にロードすると、SCAP セキュリティーガイド (SSG) からのデータストリームがロードされ、すべての組織と場所に割り当てられるようになります。

SSG は、Satellite Server のオペレーティングシステムによって提供され、`/usr/share/xml/scap/ssg/content/` にインストールされます。利用可能なデータストリームは、Satellite が実行されているオペレーティングシステムのバージョンによって異なることに注意してください。この SCAP コンテンツは、Satellite Server と同じ RHEL マイナーバージョン持つホストをスキャンする場合にのみ使用できます。詳細は、「[RHEL でサポートされている SCAP コンテンツの取得](#)」を参照してください。

前提条件

- ユーザーアカウントに `create_scap_contents` 権限を持つロールが割り当てられている。

手順

- Satellite Server で次の Hammer コマンドを使用します。

```
# hammer scap-content bulk-upload --type default
```

7.6.2. RHEL でサポートされている SCAP コンテンツの取得

Red Hat Enterprise Linux の最新の SCAP セキュリティーガイド (SSG) は Red Hat カスタマーポータルで入手できます。ホストの RHEL マイナーバージョン用に指定された SSG バージョンを取得する必要があります。

手順

1. [パッケージブラウザーの SCAP セキュリティーガイド](#) にアクセスします。
2. [バージョンメニュー](#) から、ホストが実行している RHEL マイナーバージョン向けの最新の SSG バージョンを選択します。たとえば、RHEL 8.6 の場合は、`*.el8_6` という名前のバージョンを選択します。
3. パッケージ RPM をダウンロードします。
4. RPM からデータストリームファイル (`*-ds.xml`) を展開します。以下に例を示します。

```
$ rpm2cpio scap-security-guide-0.1.69-3.el8_6.noarch.rpm \  
| cpio -iv --to-stdout ./usr/share/xml/scap/ssg/content/ssg-rhel8-ds.xml \  
> ssg-rhel-8.6-ds.xml
```

5. データストリームを Satellite にアップロードします。詳細は、「[追加の SCAP コンテンツのアップロード](#)」を参照してください。

関連情報

- Red Hat ナレッジベースの [Supported versions of the SCAP Security Guide in RHEL](#)

- [Red Hat Enterprise Linux 9 セキュリティーの強化の RHEL 9 で対応する SCAP セキュリティーガイドプロファイル](#)
- [Red Hat Enterprise Linux 8 セキュリティーの強化の RHEL 8 で対応している SCAP セキュリティーガイドプロファイル](#)
- [Red Hat Enterprise Linux 7 セキュリティーガイドの RHEL 7 で対応する SCAP セキュリティーガイドプロファイル](#)

7.6.3. 追加の SCAP コンテンツのアップロード

追加の SCAP コンテンツ (自分で作成したコンテンツ、または他の場所で取得したコンテンツ) を Satellite Server にアップロードできます。Red Hat は、Red Hat から取得した SCAP コンテンツのサポートのみを提供することに注意してください。Satellite Web UI の代わりに CLI を使用する場合は、[CLI 手順](#) を参照してください。

前提条件

- ユーザーアカウントに `create_scap_contents` 権限がある。
- SCAP データストリームファイルを取得している。

手順

1. Satellite Web UI で、**Hosts > Compliance > SCAP contents** に移動します。
2. **Upload New SCAP Content** をクリックします。
3. **Title** テキストボックスにタイトルを入力します (**My SCAP Content** など)。
4. **Scap File** で、**Choose file** をクリックし、SCAP データストリームファイルが含まれる場所に移動して、**Open** をクリックします。
5. **Locations** タブで場所を選択します。
6. **Organizations** タブで組織を選択します。
7. **Submit** をクリックします。

SCAP コンテンツファイルが正常にロードされると、**Successfully created My SCAP Content** のようなメッセージが表示されます。

CLI 手順

1. SCAP データストリームファイルを Satellite Server 上のディレクトリー (`/usr/share/xml/scap/my_content/` など) に配置します。
2. Satellite Server で次の Hammer コマンドを実行します。

```
# hammer scap-content bulk-upload --type directory \  
--directory /usr/share/xml/scap/my_content/ \  
--location "My_Location" \  
--organization "My_Organization"
```

検証

- [利用可能な SCAP コンテンツをリスト表示](#) します。SCAP コンテンツのリストに新しいタイトルが含まれています。

7.6.4. XCCDF プロファイルの調整

元の SCAP コンテンツを編集せずに、[テーラリングファイル](#) を使用して既存の XCCDF プロファイルをカスタマイズできます。1つのテーラリングファイルに複数の XCCDF プロファイルのカスタマイズを含めることができます。

テーラリングファイルは [SCAP Workbench](#) ツールを使用して作成できます。SCAP ワークベンチツールの使用方法の詳細は、[Customizing SCAP Security Guide for your use case](#) を参照してください。

作成したら、テーラリングファイルをコンプライアンスポリシーに割り当てて、ポリシー内の XCCDF プロファイルをカスタマイズできます。

7.6.5. テーラリングファイルのアップロード

テーラリングファイルをアップロードすると、それをコンプライアンスポリシーに適用して、XCCDF プロファイルをカスタマイズできます。

前提条件

- ユーザーアカウントに `create_tailoring_files` 権限がある。

手順

1. Satellite の Web UI で **Hosts > Compliance - Tailoring Files** に移動し、**New Tailoring File** をクリックします。
2. **Name** テキストボックスに、名前を入力します。
3. **Choose File** をクリックし、テーラリングファイルが含まれる場所に移動して、**Open** を選択します。
4. **Submit** をクリックして、選択したテーラリングファイルをアップロードします。

7.7. コンプライアンスポリシーの管理

コンプライアンスポリシー は、SCAP コンテンツの特定の XCCDF プロファイルに対して、指定されたホストのコンプライアンスをチェックする定期的な監査です。

Satellite Server 上でスキャンのスケジュールを指定すると、ホスト上でスキャンが実行されます。スキャンが完了すると、ARF 形式のレポートが生成され、Satellite Server にアップロードされます。コンプライアンスポリシーがスキャンされたホストに変更を加えることはありません。

コンプライアンスポリシーは、SCAP クライアント設定と cron スケジュールを定義します。ポリシーは、ポリシーが割り当てられているホスト上に SCAP クライアントとともにデプロイされます。

7.7.1. コンプライアンスポリシー

コンプライアンスポリシー と呼ばれる定期監査は、XCCDF プロファイルに対して指定したホストのコンプライアンスをチェックするスケジュールタスクです。スキャンのスケジュールは Satellite Server で指定され、スキャンはホストで実行されます。スキャンが完了すると、**Asset Reporting File (ARF)** が XML 形式で生成され、Satellite Server にアップロードされます。スキャンの結果はコンプライアンス

スポリシーダッシュボードで確認できます。コンプライアンスポリシーでは、スキャンされるホストに変更はなされません。SCAP コンテンツには、関連付けられたルールのあるいくつかのプロファイルが含まれますが、デフォルトではポリシーは含まれません。

7.7.2. コンプライアンスポリシーの作成

コンプライアンスポリシーを作成することで、セキュリティーコンプライアンス要件を定義および計画し、ホストがセキュリティーポリシーに準拠した状態を維持できます。

前提条件

- 選択した [コンプライアンスポリシーのデプロイメント方法](#) に合わせて Satellite を設定している。
- Satellite で、SCAP コンテンツ、最終的にはテーラリングファイルが利用可能である。
 - 利用可能な SCAP コンテンツを確認するには、「[利用可能な SCAP コンテンツのリスト](#)」を参照してください。
 - SCAP コンテンツとテーラリングファイルをアップロードするには、「[SCAP コンテンツの設定](#)」を参照してください。
- ユーザーアカウントに `view_policies` 権限と `create_policies` 権限がある。

手順

1. Satellite Web UI で、**Hosts > Compliance - Policies** に移動します。
2. **New Policy** または **New Compliance Policy** をクリックします。
3. デプロイメント方法を **Ansible**、**Puppet**、または **Manual** から選択します。 **Next** をクリックします。
4. ポリシーの名前、説明 (省略可能) を入力してから **次へ** をクリックします。
5. 適用する **SCAP Content** および **XCCDF Profile** を選択し、 **Next** をクリックします。
Satellite は、選択された XCCDF プロファイルにルールが含まれているかどうかを検出しないことに注意してください。 **Default XCCDF Profile** などの空の XCCDF プロファイルは、空のレポートを返します。
6. オプション: XCCDF プロファイルをカスタマイズするには、 **Tailoring File** と **XCCDF Profile in Tailoring File** を選択し、 **Next** をクリックします。
7. ポリシーを適用するスケジュール時刻を指定します。 **Period** のリストから、 **Weekly**、 **Monthly**、または **Custom** を選択します。 **Custom** オプションを使用すると、ポリシーのスケジュールをより柔軟に行うことができます。
 - **Weekly** を選択したら **Weekday** リストから曜日を選択します。
 - **Monthly** を選択したら **Day of month** フィールドで日付を指定します。
 - **Custom** を選択したら **Cron line** フィールドに有効な Cron 式を入力します。
8. ポリシーを適用する場所を選択し、 **Next** をクリックします。
9. ポリシーを適用する組織を選択し、 **Next** をクリックします。

10. オプション: ポリシーを割り当てるホストグループを選択します。

11. **Submit** をクリックします。

7.7.3. コンプライアンスポリシーの表示

特定の OpenSCAP コンテンツおよびプロファイルの組み合わせ別に適用されるルールをプレビューできます。これは、ポリシーを計画するときに役立ちます。

前提条件

- ユーザーアカウントに **view_policies** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Compliance - Policies** に移動します。
2. 必要なポリシーの **Actions** 列で、**Show Guide** をクリックするか、リストから Show Guide を選択します。

7.7.4. コンプライアンスポリシーの編集

Satellite Web UI では、コンプライアンスポリシーを編集できます。

Puppet エージェントが、次回の実行時に、編集されたポリシーをホストに適用します。これはデフォルトで 30 分ごとに実行されます。Ansible を使用する場合は、Ansible ロールを手動で再度実行するか、ホスト上で Ansible ロールを実行する定期的なリモート実行ジョブを設定する必要があります。

前提条件

- ユーザーアカウントに **view_policies** 権限と **edit_policies** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Compliance - Policies** に移動します。
2. 必要なポリシーの名前をクリックします。
3. 必要な属性を編集します。
4. **Submit** をクリックします。

7.7.5. コンプライアンスポリシーの削除

Satellite Web UI では、既存のコンプライアンスポリシーを削除できます。

前提条件

- ユーザーアカウントに **view_policies** 権限と **destroy_policies** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Compliance - Policies** に移動します。

2. 必要なポリシーの **Actions** 列で、リストから **Delete** を選択します。
3. 確認メッセージで **OK** をクリックします。

7.8. コンプライアンスポリシーのデプロイ

コンプライアンスポリシーをデプロイするには、SCAP クライアントをインストールし、cron スケジュールファイルを更新して、ポリシーで選択されている SCAP コンテンツをホストにアップロードする必要があります。

7.8.1. Ansible を使用したホストグループへのポリシーのデプロイ

Ansible を使用してホストグループにコンプライアンスポリシーをデプロイすると、Ansible ロールによって SCAP クライアントがインストールされ、選択したコンプライアンスポリシーに従ってホスト上で OpenSCAP スキャンが設定されます。

前提条件

- Capsule で OpenSCAP を有効にしている。詳細は、[Capsule Server のインストールの Capsule Server での OpenSCAP の有効化](#) を参照してください。
- Satellite Client 6 リポジトリを有効にして Satellite に同期し、ホスト上でそれを有効にした。
- Ansible デプロイメント方法を使用して [コンプライアンスポリシーを作成](#) し、ホストグループを割り当てている。

手順

1. Satellite Web UI で、**Configure > Host Groups** に移動します。
2. OpenSCAP レポート用に設定するホストグループをクリックします。
3. **OpenSCAP Capsule** リストから、使用する OpenSCAP が有効な Capsule を選択します。
4. **Ansible Roles** タブで、**foreman.foreman_scap_client** Ansible ロールを割り当てます。
5. オプション: **Parameters** タブで、ロールの Ansible 変数を設定します。
6. **Submit** をクリックして変更を保存します。
7. 必要なホストグループの行で、**Actions** 列に移動し、**Run all Ansible roles** を選択します。

7.8.2. Ansible を使用したホストへのポリシーのデプロイ

Ansible を使用してホストにコンプライアンスポリシーをデプロイすると、Ansible ロールによって SCAP クライアントがインストールされ、選択したコンプライアンスポリシーに従ってホスト上で OpenSCAP スキャンが設定されます。

前提条件

- Capsule で OpenSCAP を有効にしている。詳細は、[Capsule Server のインストールの Capsule Server での OpenSCAP の有効化](#) を参照してください。

- Satellite Client 6 リポジトリを有効にして Satellite に同期し、ホスト上でそれを有効にした。
- Ansible デプロイメント方法を使用して [コンプライアンスポリシーを作成](#) している。

手順

1. Satellite Web UI で **Hosts > All Hosts** に移動して、OpenSCAP レポートを設定するホストで、**Edit** をクリックします。
2. **OpenSCAP Capsule** リストから、使用する OpenSCAP が有効な Capsule を選択します。
3. **Ansible Roles** タブで、**theforeman.foreman_scap_client** Ansible ロールを追加します。
4. オプション: **Parameters** タブで、ロールの Ansible 変数を設定します。
5. **Submit** をクリックして変更を保存します。
6. **Hosts** ブレッドクラムリンクをクリックして、ホストのインデックスページに戻ります。
7. ポリシーを追加するホストを選択します。
8. **Select Action** をクリックします。
9. リストから **Assign Compliance Policy** を選択します。
10. **Assign Compliance Policy** ウィンドウで、**Remember hosts selection for the next bulk action** を選択します。
11. 使用可能なポリシーのリストから必要なポリシーを選択し、**Submit** をクリックします。
12. **Select Action** をクリックします。
13. リストから **Run all Ansible roles** を選択します。

7.8.3. Puppet を使用したホストグループへのポリシーのデプロイ

Puppet を使用してホストグループにコンプライアンスポリシーをデプロイすると、Puppet エージェントによって SCAP クライアントがインストールされ、選択したコンプライアンスポリシーに従って次の Puppet 実行時にホスト上で OpenSCAP スキャンが設定されます。

前提条件

- Capsule で OpenSCAP を有効にしている。詳細は、**Capsule Server のインストールの [Capsule Server での OpenSCAP の有効化](#)** を参照してください。
- Satellite Client 6 リポジトリを有効にして Satellite に同期し、ホスト上でそれを有効にした。
- Puppet デプロイメント方法を使用して [コンプライアンスポリシーを作成](#) し、ホストグループを割り当てている。

手順

1. Satellite Web UI で、**Configure > Host Groups** に移動します。

2. OpenSCAP レポート用に設定するホストグループをクリックします。
3. **Environment** リストで、**foreman_scap_client*** Puppet クラスを含む Puppet 環境を選択します。
4. **OpenSCAP Capsule** リストで、使用する OpenSCAP が有効になっている Capsule を選択します。
5. **Puppet ENC** タブで、**foreman_scap_client** Puppet クラスを追加します。
6. オプション: **Puppet Class Parameters** を設定します。
7. **Submit** をクリックして変更を保存します。

7.8.4. Puppet を使用したホストへのポリシーのデプロイ

Puppet を使用してホストにコンプライアンスポリシーをデプロイすると、Puppet エージェントによって SCAP クライアントがインストールされ、選択したコンプライアンスポリシーに従って次の Puppet 実行時にホスト上で OpenSCAP スキャンが設定されます。

前提条件

- Capsule で OpenSCAP を有効にしている。詳細は、**Capsule Server のインストールの Capsule Server での OpenSCAP の有効化** を参照してください。
- Satellite Client 6 リポジトリを有効にして Satellite に同期し、ホスト上でそれを有効にした。
- Puppet デプロイメント方法を使用して **コンプライアンスポリシーを作成** している。

手順

1. Satellite Web UI で **Hosts > All Hosts** に移動して、OpenSCAP レポートを設定するホストで、**Edit** をクリックします。
2. **Environment** リストから、**foreman_scap_client** および **foreman_scap_client::params** が含まれる Puppet 環境を選択します。
3. **OpenSCAP Capsule** リストから、使用する OpenSCAP が有効な Capsule を選択します。
4. **Puppet ENC** タブで、**foreman_scap_client** Puppet クラスを追加します。
5. オプション: **Puppet Class Parameters** を設定します。
6. **Hosts** ブレッドクラムリンクをクリックして、ホストのインデックスページに戻ります。
7. ポリシーを追加するホストを選択します。
8. **Select Action** をクリックします。
9. リストから **Assign Compliance Policy** を選択します。
10. **Assign Compliance Policy** ウィンドウで、**Remember hosts selection for the next bulk action** を選択します。
11. 使用可能なポリシーのリストから必要なポリシーを選択し、**Submit** をクリックします。

7.9. オンデマンドでのセキュリティーコンプライアンススキャンの実行

ホストは、ホストに割り当てられたコンプライアンスポリシーで定義された CRON スケジュールに従って、OpenSCAP スキャンを定期的に行います。ただし、ホスト上で、設定されているすべてのコンプライアンスポリシーのスキャンを随時手動で実行することもできます。

前提条件

- ユーザーアカウントに、**view_hosts**、**create_job_invocations**、および **view_job_invocations** 権限がある。
- コンプライアンスポリシーを作成し、ホストにデプロイした。
 - ポリシーの管理の詳細は、「[コンプライアンスポリシーの管理](#)」を参照してください。
 - ポリシーのデプロイの詳細は、「[コンプライアンスポリシーのデプロイ](#)」を参照してください。

手順

1. **Hosts > All Hosts** に移動します。
2. 必要なホストのホスト名をクリックします。
3. ホストの詳細ページで、**Schedule a job** ドロップダウンメニューを展開します。
4. **Run OpenSCAP scan** を選択します。

検証

1. ホストの詳細の概要で、**Recent jobs** カードを見つけます。
2. **Running** タブを選択します。ジョブがすでに終了していない限り、表に **Run scan for all OpenSCAP policies** というジョブが表示されます。
3. **Recent jobs** カードで、**Finished** タブを選択します。
4. ジョブが正常に終了すると、ジョブの行に **succeeded** ステータスが表示されます。
5. オプション: ジョブ名をクリックして呼び出しの詳細を確認します。

7.10. コンプライアンスの監視

Satellite を使用すると、コンプライアンスの監視と管理を一元化できます。コンプライアンスダッシュボードでは、ホストのコンプライアンスの概要が表示され、そのポリシーの範囲内にある各ホストの詳細を表示できます。コンプライアンスレポートでは、適用可能なポリシーを使用して、各ホストのコンプライアンスの詳細を分析します。この情報を使用して、各ホストが提示するリスクを評価し、ホストがコンプライアンスを満たすために必要なリソースを管理できます。SCAP を使用してコンプライアンスを監視することで、ポリシーのコンプライアンスを確認し、コンプライアンスの変化を検出できます。

7.10.1. コンプライアンスレポートの検索

コンプライアンスレポート検索フィールドを使用して、ホストのサブセットで使用可能なレポートのリストをフィルタリングします。

手順

1. Satellite Web UI で、**Hosts > Reports** に移動します。
2. オプション: 空の **Search** フィールドをクリックすると、利用可能な検索パラメーターがリスト表示されます。
3. **Search** フィールドに検索クエリーを入力し、**Search** をクリックします。検索クエリーでは大文字と小文字は区別されません。

検索クエリーの例

5つを超えるルールが失敗したすべてのコンプライアンスレポートを検索する

```
failed > 5
```

2023年1月1日以降に作成された、ホスト名に **prod-** が含まれるホストのすべてのコンプライアンスレポートを検索します。

```
host ~ prod- AND date > "Jan 1, 2023"
```

1時間前に **rhel7_audit** コンプライアンスポリシーによって生成されたすべてのレポートを検索する

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

XCCDF ルールに合格するレポートを検索する

```
xccdf_rule_passed = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

XCCDF ルールに不合格となるレポートを検索する

```
xccdf_rule_failed = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

結果が XCCDF ルールに合格または不合格以外のレポートを検索する

```
xccdf_rule_othered = xccdf_org.ssgproject.content_rule_firefox_preferences-auto-download_actions
```

追加情報

- 論理演算子 **and**、**not**、および **has** を使用すると、複雑なクエリーを作成できます。論理演算子の詳細は、[Red Hat Satellite の管理の 詳細な検索に対してサポートされる演算子](#) を参照してください。
- 正規表現は、検索クエリーで使用できません。ただし、1つの検索式に複数のフィールドを使用できます。使用可能なすべての検索演算子の詳細は、[Red Hat Satellite の管理の 詳細な検索に対してサポートされる演算子](#) を参照してください。
- 検索をブックマークすると、同じ検索クエリーを再利用できます。詳細は、[Red Hat Satellite の管理の ブックマークの作成](#) を参照してください。

7.10.2. コンプライアンスのメール通知

Satellite Server は、**コンプライアンスポリシーサマリー** のメール通知をサブスクライブしているすべてのユーザーに、OpenSCAP サマリーメールを送信します。通知メールをサブスクライブする方法の詳細は、「[メール通知の設定](#)」を参照してください。

ポリシーが実行されるたびに、Satellite は直前の実行との比較で結果をチェックし、変更がないかどうかを確認します。メールは各サブスクライバーがリクエストする頻度で送信され、各ポリシーのサマリーと直近の結果を提供します。

7.10.3. コンプライアンスポリシー統計情報の表示

コンプライアンスポリシーダッシュボードを表示して、特定のポリシーのコンプライアンスレポートを確認できます。コンプライアンスポリシーダッシュボードでは、ホストのコンプライアンスの統計的なサマリーが表示され、そのポリシーの範囲内にある各ホストのレポートの詳細を表示できます。

コンプライアンスレポートを表示するときは、次のホストを優先することを検討してください。

- **Failed** と評価されたホスト
- ステータスが不明なため **Never audited** とラベル付けされたホスト

前提条件

- ユーザーアカウントに **view_policies** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Policies** に移動します。
2. 必要なポリシーの行で、**Actions** 列に移動し、**Dashboard** をクリックします。

7.10.4. ルールのコンプライアンス結果に基づくホストの調査

簡略化されたレポートを調べ、ポリシールールを使用して、特定のルールの不合格など、特定のコンプライアンス結果を持つホストをリスト表示できます。

前提条件

- ユーザーアカウントに **view_arf_reports** 権限と **view_hosts** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Reports** に移動します。
2. **Reported At** 列で、必要なホストとコンプライアンスポリシーのレポートに移動し、時刻のリンクをクリックします。
3. Satellite で、ポリシールールの簡略化されたリストとスキャンの結果が表示されます。
4. オプション: チェック結果によってルールをフィルターします。**Show log messages** ドロップダウンリストから、次のフィルターのいずれかを選択します。
 - **Failed and othered** - 不合格となったルール、またはスキャン中にチェックされなかったルールを表示します。
 - **Failed only** - 不合格となったルールのみを表示します。

5. オプション: ルールの詳細を調べます。 **Message** 列で、ルール名の横にあるアイコンをクリックします。
6. 必要なルールの行で、 **Actions** 列に移動し、 **Hosts failing this rule** をクリックします。

7.10.5. ホストのコンプライアンス違反の調査

完全なコンプライアンスレポートを調べて、ホストがルールに準拠できなかった理由を特定できます。場合によっては、非準拠状態を修復する方法を確認できます。



警告

推奨される修復アクションやスクリプトは、先に実稼働以外の環境でテストしてから実装してください。修復によりシステムが機能しなくなる可能性があります。

コンプライアンスレポートは次のエリアで構成されています。

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

前提条件

- ユーザーアカウントに **view_arf_reports** 権限と **view_hosts** 権限がある。

手順

1. Satellite Web UI で、 **Hosts > Reports** に移動して、すべてのコンプライアンスレポートをリスト表示します。
2. 必要なホストの行で、 **Actions** 列に移動し、 **Full Report** をクリックして、評価レポートの完全な詳細を表示します。
3. **Evaluation Characteristics** エリアに移動して、特定のプロファイルに対するホストの評価に関する基本情報を確認します。
4. **Compliance and Scoring** エリアに移動して、評価の統計情報とホストのコンプライアンススコアを確認します。
5. **Rule Overview** に移動してルールを調べます。
6. オプション: **pass**、 **notapplicable**、または **fix** など、非表示にするチェックステータスの選択を解除します。
7. オプション: **Group rule by** ドロップダウンメニューから、 **Severity** などのルールのグループ化の基準を選択します。

8. オプション: 検索フィールドに検索文字列を入力して、ルールをタイトルでフィルターします。検索は大文字と小文字を区別せず、入力時に動的に適用されます。
9. ルールのタイトルをクリックして、結果の詳細を調べます。
 - ルールの説明と、ホストのコンプライアンスを満たすための指示 (利用できる場合)。
 - ルールの根拠。
 - 場合により、修復スクリプト。

7.10.6. コンプライアンスレポートの削除

Satellite のコンプライアンスレポートを削除できます。

前提条件

- ユーザーアカウントに **view_arf_reports** 権限と **destroy_arf_reports** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Reports** に移動します。
2. コンプライアンスレポートウィンドウで、削除するポリシーを特定し、ポリシーの名前の右側にある **Delete** を選択します。
3. **OK** をクリックします。

7.10.7. 複数のコンプライアンスレポートの削除

複数のコンプライアンスポリシーを同時に削除できます。ただし、Satellite Web UI では、コンプライアンスポリシーはページ分割されているため、レポートを1ページずつ削除する必要があります。すべての OpenSCAP レポートを削除する場合は、[API ガイドの OpenSCAP レポートの削除](#)にあるスクリプトを使用します。

前提条件

- ユーザーアカウントに **view_arf_reports** 権限と **destroy_arf_reports** 権限がある。

手順

1. Satellite Web UI で、**Hosts > Reports** に移動します。
2. コンプライアンスレポートウィンドウで、削除するコンプライアンスレポートを選択します。
3. リストの右上の **Delete reports** を選択します。
4. 削除するページ数だけ、この手順を繰り返します。

第8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

バックアップツールを使用して作成したインスタンスを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。バックアップを復元した後は、古いインスタンスをすべて廃止する必要があります。

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 7 System Administrator's Guide](#) の [System Backup and Recovery](#) 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
```

8.1. 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。

Red Hat Enterprise Linux 8 の場合:

```
# 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
```

Red Hat Enterprise Linux 7 の場合:

```
# du -sh /var/opt/rh/rh-postgresql12/lib/pgsql/data /var/lib/pulp
100G /var/opt/rh/rh-postgresql12/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8.1 Red Hat Enterprise Linux 8 のバックアップデータ圧縮率

データのタイプ	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	<code>/var/lib/pgsql/data</code>	80 - 85%	100 GB → 20 GB
Pulp RPM ファイル	<code>/var/lib/pulp</code>	(非圧縮)	100 GB
設定ファイル	<code>/var/lib/qpidd</code> <code>/var/lib/tftpboot</code> <code>/etc</code> <code>/root/ssl-build</code> <code>/var/www/html/pub</code> <code>/opt/puppetlabs</code>	85%	942 MB → 141 MB

表8.2 Red Hat Enterprise Linux 7 のバックアップデータ圧縮率

データのタイプ	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	<code>/var/opt/rh/rh-postgresql12/lib/pgsql/data</code>	80 - 85%	100 GB → 20 GB
Pulp RPM ファイル	<code>/var/lib/pulp</code>	(非圧縮)	100 GB
設定ファイル	<code>/var/lib/qpidd</code> <code>/var/lib/tftpboot</code> <code>/etc</code> <code>/root/ssl-build</code> <code>/var/www/html/pub</code> <code>/opt/puppetlabs</code>	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。

- 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。

8.2. 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せん。

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ため、バックアップまたは増分バックアップから復元する際には、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。

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
```

8.3. 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
```

8.4. 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

- 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
```

- 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
```

- 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
```

- 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
```

8.5. 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。

8.6. 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
```

8.7. 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

- システムがスナップショットするディレクトリーに LVM を使用していること (`/var/lib/pulp/` および `/var/opt/rh/rh-postgresql12/lib/pgsql`)。
- 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。

8.8. 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 -y /var/backup_directory
```

第9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、[8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。

9.1. 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 -Rnv /
```

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からの 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。

9.2. 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。

9.3. 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と同じ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。[Installing Capsule Server](#) 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、「[外部 Capsule の同期](#)」の手順で Satellite から同期します。

9.3.1. 外部 Capsule の同期

外部 Capsule と Satellite を同期します。

手順

1. 外部 Capsule から同期するには、Satellite web UI で関連する組織とロケーションを選択するか、**Any Organization** と **Any Location** を選択します。
2. Satellite web UI で、**Infrastructure > Capsules** に移動し、Capsule の名前をクリックして同期します。
3. **概要** タブで **同期** を選択します。

第10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** スクリプトを使用する必要があります。

Satellite Server の名前を変更する場合は、すべての Satellite クライアントを再登録し、新しい Satellite ホスト名を指すように各 Capsule Server を設定する必要があります。カスタム SSL 証明書を使用する場合は、新しいホスト名で再生成する必要があります。virt-who 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する必要があります。

Capsule Server の名前を変更する場合は、すべての Capsule クライアントを再登録し、Satellite Web UI で Capsule ホスト名を更新する必要があります。カスタム SSL 証明書を使用する場合は、新しいホスト名で再生成する必要があります。



警告

名前変更プロセスを実行すると、変更対象であるホストの Satellite Server 上の全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、全サービスが再開されます。

10.1. 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、新規ホスト名への参照をすべて更新することができます。

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 を再設定する必要があります。**satellite-change-hostname** スクリプトは、Satellite Server 用の外部認証を破棄してしまいます。外部認証の設定の詳細は、[14章 外部認証の設定](#) を参照してください。

virt-who を使用する場合、**satellite-change-hostname** スクリプトを実行した後、新しいホスト名で virt-who 設定ファイルを更新する必要があります。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。

前提条件

- **hostname** コマンドと **hostname -f** コマンドの両方で、Satellite Server の FQDN を返す必要があります。そうしないと、**satellite-change-hostname** スクリプトが完了しません。**hostname** コマンドが FQDN の代わりに Satellite Server のショートネームを返す場合、**satellite-change-hostname** スクリプトの使用を試みる前に、**hostnamectl set-hostname 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。
- ホスト名を変更する前に、Satellite Server のバックアップを実行してください。名前変更プロセスが失敗した場合は、バックアップから復元してください。詳細は、[8章 Satellite Server および Capsule Server のバックアップ](#) を参照してください。
- オプション: Satellite Server がカスタムの SSL 証明書をインストールしている場合は、ホストの新しい名前用に新しい証明書を取得すること。詳細は、[Installing Satellite Server from a](#)

Connected Network の [Configuring Satellite Server with a Custom SSL Certificate](#) を参照してください。

手順

1. Satellite Server で **satellite-change-hostname** スクリプトを実行する適切な方法を選択して、新しいホスト名と Satellite 認証情報を提供します。
 - 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 証明書で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. オプション: 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、[Installing Satellite Server from a Connected Network of Deploying a Custom SSL Certificate to Satellite Server](#) 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 capsule_id_number
```

10.2. 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、新規ホスト名への参照をすべて更新することができます。



注記

- **hostname** コマンドと **hostname -f** コマンドの両方で、Capsule Server の FQDN を返す必要があります。そうしないと、**satellite-change-hostname** スクリプトが完了しません。
- **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 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。

前提条件

- Capsule Server をバックアップすること。**satellite-change-hostname** スクリプトを実行すると、Capsule Server への変更は元に戻せません。名前変更プロセスが失敗した場合は、バックアップから復元してください。
ホスト名を変更する前にバックアップを実行してください。詳細は、[8章 Satellite Server および Capsule Server のバックアップ](#)を参照してください。

手順

1. 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。
2. Satellite Server 上で、証明書アーカイブファイルを Capsule Server にコピーし、プロンプトが表示されたら、**root** ユーザーのパスワードを提供します。この例では、アーカイブファイルは **root** ユーザーのホームディレクトリーにコピーされますが、別の場所にコピーすることもできます。

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

3. 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。

4. オプション: Capsule Server で、Capsule Server のカスタム証明書を作成した場合、証明書をデプロイするには、**capsule-certs-generate** コマンドが返す **satellite-installer** コマンドを入力します。詳細は、[Installing Capsule Server の Deploying a Custom SSL Certificate to Capsule Server](#)を参照してください。

- 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、[Managing Hosts Guide の Configuring and Setting up Remote Jobs](#) を参照してください。

```
# yum remove -y katello-ca-consumer*

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

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

# subscription-manager refresh
```

- Satellite Web UI で、**Infrastructure** > **Capsules** に移動します。
- リストで Capsule Server を見つけ、右側にある **編集** をクリックします。
- 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。
- DNS サーバーで、Capsule Server の新規ホスト名用のレコードを追加し、古いホスト名のレコードを削除します。

第11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。

11.1. 監査レコードの削除

監査レコードは Satellite で自動作成されます。**foreman-rake audits:expire** コマンドを使うと、監査はいつでも取り消すことができます。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 audits:expire** 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削除されます。**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削除する場合は、以下のコマンドを実行します。

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

11.2. 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
```

11.3. 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
```

11.4. 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
```

11.5. 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リング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'
```

11.6. 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。

11.7. 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 7 [論理ボリュームマネージャーの管理の論理ボリュームの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。

11.8. 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。

手順

- Satellite Server または CapsuleServer にパッケージをインストールするには、次のコマンドを入力します。

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

- Satellite Server または CapsuleServer で特定のパッケージを更新するには、次のコマンドを入力します。

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

- Satellite Server または CapsuleServer 上のすべてのパッケージを更新するには、次のコマンドを入力します。

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

yum を使用したパッケージ更新の確認

yum を使用して更新を確認する場合は、コマンドを入力して手動でパッケージをインストールおよび更新してから、**yum** を使用して更新を確認できます。

```
# satellite-maintain packages unlock
# yum check update
# satellite-maintain packages lock
```

パッケージを個別に更新すると、Satellite Server または CapsuleServer 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 でのパッケージの更新の詳細については、[Satellite Server の更新](#) を参照してください。

Satellite Server または Capsule Server Package Management の yum を有効にする

yum を使用して直接システムにパッケージをインストールして更新し、システムの安定性をご自身で管理する場合は、以下のコマンドを入力します。

```
# satellite-maintain packages unlock
```

パッケージ管理のデフォルト設定への復元

デフォルト設定を復元して Satellite Server または Capsule Server を有効にし、ユーザーが **yum** を使用してパッケージをインストールおよび更新できないようにし、システムの安定性を確保する必要がある場合は、以下のコマンドを入力します。

```
# satellite-maintain packages lock
```

11.9. 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。この手順を使用して、Satellite でこのディスク領域の一部を確保します。

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確保します。

```
# 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
```

11.10. 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、スペースを再利用できます。

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12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。

12.1. 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を署名する必要があります。Server 証明書には、必要な値で X.509 v3 **Key Usage** および **Extended Key Usage** エクステンションが必要であるため、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、Satellite Server 証明書と CA バンドルを受け取ります。

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_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。

12.2. 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、必要な値で X.509 v3 **Key Usage** および **Extended Key Usage** エクステンションが必要であるため、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 --foreman-proxy-fqdn "capsule.example.com" \
--certs-tar "/root/My_Certificates/capsule.example.com-certs.tar" \
--server-cert "/root/My_Certificates/capsule_cert.pem" \
--server-key "/root/My_Certificates/capsule_cert_key.pem" \
--server-ca-cert "/root/My_Certificates/ca_cert_bundle.pem" \
--certs-update-server
```

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



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13章 問題のログとレポート

本章では、関連するログファイルに関する情報、デバッグロギングを有効にする方法、サポートケースを開き、関連するログ tar ファイルを添付する方法、Satellite Web UI 内でサポートケースにアクセスする方法など、Satellite における問題のログおよびレポート方法について説明します。

本章で説明されたログファイルと他の情報を使用して独自にトラブルシューティングを行ったり、サポートが必要な場合は、これらの情報と他の多くのファイルとともに診断および設定情報を取得して Red Hat サポートに送信することができます。

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
```

13.1. デバッグロギングの有効化

デバッグロギングでは、最も詳細にわたるログ情報が提供され、Satellite とそのコンポーネントで発生する可能性がある問題のトラブルシューティングが簡単になります。Satellite CLI で、デバッグロギングを有効にして、Satellite の詳細なデバッグ情報をログに記録します。

手順

1. デバッグロギングを有効にするには、次のコマンドを入力します。

```
# 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
```

13.2. デバッグに役立つログレベルの増加

デフォルトでは、Satellite には **:INFO** レベルのロギングが有効になっています。Satellite のログレベルを増減できます。

すべてのコンポーネントでデバッグレベルのログを有効にする

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

特定のコンポーネントのデバッグレベルのログを有効にする

```
# hammer admin logging --components "Component" --level-debug
```

デバッグレベルのログを **INFO** に戻す

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

すべてのコンポーネントと変更された設定ファイルのリスト表示

```
# hammer admin logging --list
-----|-----|-----
COMPONENT | AUTO-DETECTED BY EXISTENCE OF | DESTINATIONS
-----|-----|-----
dhcpd     | /etc/dhcp/dhcpd.conf           | syslog /var/log/dhcpd-debug.log
postgres  | /var/lib/pgsql/data/postgresql.conf | syslog /var/lib/pgsql/data/pg_log/
proxy     | /etc/foreman-proxy/settings.yml  | /var/log/foreman-proxy/proxy.log
qpidd     | /etc/qpidd/qpidd.conf           | syslog
rails     | /etc/foreman/settings.yml        | /var/log/foreman/production.log
tomcat    | /etc/candlepin/candlepin.conf    | /var/log/candlepin/ /var/log/tomcat/
virt-who  | /etc/sysconfig/virt-who         | syslog
-----|-----|-----
```

13.2.1. Hammer のログレベルの引き上げ

Hammer のログは `~/hammer/log/hammer.log` にあります。 `/etc/hammer/cli_config.yml` を編集し、 `:log_level:` を設定します。

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

13.2.2. Capsule のログレベルの引き上げ

Capsule のログは、 `/var/log/foreman-proxy/proxy.log` にあります。 `/etc/foreman-proxy/settings.yml` の **DEBUG** 行のコメントを解除します。

```
:log_level: DEBUG
```

その後、必ず **foreman-proxy** サービスを再起動してください。

```
# systemctl restart foreman-proxy
```

注意

インストーラーを実行すると、この変更が元に戻ります。

13.2.3. Candlepin のログレベルの引き上げ

Candlepin のログは `/var/log/candlepin/candlepin.log` にあります。 `/var/log/candlepin/error.log` のデバッグを容易にするために、エラーも別のファイルに記録されます。

`/etc/candlepin/candlepin.conf` を拡張します:

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

その後、必ず **tomcat** サービスを再起動してください。

```
# 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することができます。

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

13.2.4. Satellite のログレベルの引き上げ

Satellite のログは `/var/log/foreman/production.log` にあります。

Satellite は Apache のログを次の場所に保存します。

- `/var/log/httpd/foreman_error.log`
- `/var/log/httpd/foreman_access.log`
- `/var/log/httpd/foreman_ssl_error.log`
- `/var/log/httpd/foreman_ssl_access.log`

手順

1. `/etc/foreman/settings.yaml` でログレベルを設定します:

```
:logging:
  :production:
    :type: file
    :layout: pattern
    :level: debug
```

2. `/etc/foreman/settings.yaml` で選択したロガーを有効にします:

```
:loggers:
  :ldap:
    :enabled: true
  :permissions:
    :enabled: true
  :sql:
    :enabled: true
```

一部の領域からのログを表示するには、デバッグログを設定する必要があることに注意してください。

3. Satellite サービスを再起動します。

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

デフォルト値を持つロガーの完全なリストは、`Foreman::Logging.add_loggers` コマンドの `/usr/share/foreman/config/application.rb` にあります。

13.2.5. Qpid ディスパッチルーターのログレベルの引き上げ

Qpid は syslog にログを記録し、`/var/log/messages` または `journalctl` で表示できます。`/etc/qpid-dispatch/qdrouterd.conf` でデバッグログを有効にします。

```
enable: debug+
```

その後、必ず Qpid Dispatch Router を再起動してください。

```
# systemctl restart qdrouterd
```

注意

インストーラーを実行すると、この変更が元に戻ります。

13.2.6. Qpid Broker のログレベルの引き上げ

Qpid は syslog にログを記録し、`/var/log/messages` または `journalctl` で表示できます。`/etc/qpid/qpidd.conf` でログレベルを設定します。

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

その後、必ず Qpid Broker を再起動してください。

```
# systemctl restart qpidd
```

注意

インストーラーを実行すると、この変更が元に戻ります。

13.2.7. 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。`/etc/opt/rh/rh-redis5/redis.conf` でログレベルを設定します。

```
loglevel debug
```

その後、必ず Redis サービスを再起動してください。

```
# systemctl restart rh-redis5-redis
```

13.2.8. Postgres のログレベルを上げる

Postgres のログは `/var/opt/rh/rh-postgresql12/lib/pgsql/data/log/` にあります。`/var/opt/rh/rh-postgresql12/lib/pgsql/data/postgresql.conf` の `log_statement` のコメントを解除します。

```
log_statement = 'all'
```

後で必ず Satellite サービスを再起動してください。

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

注意

Satellite インストールのサイズに基づいて、これによりディスクスペースがすぐにいっぱいになる可能性があります。どうしても必要な場合にのみこれをオンにしてください。

デバッグログ設定の詳細は、[Postgresqlのドキュメント](#) を参照してください。

13.2.9. 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。インストール中に Satellite Installer のログレベルを上げるには:

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

13.2.10. 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、`/var/log/messages` または `journalctl` で表示できます。次の設定を `/etc/pulp/settings.py` ファイルに追加します。

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

その後、必ず Pulp サービスを再起動してください。

```
# systemctl restart \
pulpcore-api \
pulpcore-content \
pulpcore-resource-manager \
pulpcore-worker@1 \
pulpcore-worker@2 \
rh-redis5-redis
```

13.2.11. Puppet Agent のログレベルの引き上げ

Satellite Server の Puppet エージェントのログレベルを上げることができます。

手順

1. `/etc/puppetlabs/puppet/puppet.conf` ファイルの `[agent]` ブロックに次の行を追加します。

```
[agent]
log_level = debug
```

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

13.2.12. Puppet サーバーのログレベルの引き上げ

Satellite Server の Puppet サーバーのログレベルを上げることができます。

前提条件

- Satellite で Puppet が有効になっている。詳細は、[Red Hat Satellite で Puppet インテグレーションを使用した設定の管理](#) の [Satellite と Puppet のインテグレーションの有効化](#) を参照してください。

手順

1. `/etc/puppetlabs/puppet/puppet.conf` ファイルの `[master]` ブロックに次の行を追加します。

```
[master]
log_level = debug
```

2. Puppet サーバーを再起動します。

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

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

13.3. サービスのステータスの取得

手順

1. Satellite web UI で、**Administer > About** に移動します。
2. **Smart Proxies** タブでは、すべての Capsules のステータスを表示できます。
3. **Compute Resources** タブで、接続されているコンピュートリソースプロバイダーのステータスを表示できます。
4. **Backend System Status** テーブルでは、すべてのバックエンドサービスのステータスを表示できます。

CLI 手順

- **hammer ping** を実行して、データベースと Satellite サービスから情報を取得します。

```
# hammer ping
```

- **satellite-maintain** を使用して、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
```

- 可用性チェックを実行するには、**satellite-maintain** を使用します。

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

13.4. 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てタスクを実行します。Satellite に問題が発生した場合は、Satellite サービスのステータスを確認してください。

手順

- **satellite-maintain** を使用して Satellite サービスを再起動します。

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

ヒント

詳細については、**foreman-maintain --help** を実行してください。

13.5. 個別のロガーの有効化

個別のロガーを有効にして、一部のロギングを選択的に有効にできます。Satellite では、以下のロガーを使用します。

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

通知

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。

手順

1. 任意の個別ロガーを有効化します。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: デフォルト値にロガーをリセットするには、以下のコマンドを入力します。

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

13.6. 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** ユーティリティーを使用します。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。

13.7. SATELLITE が提供するログファイルディレクトリー

Red Hat Satellite は、システム情報を通知とログファイルの形式で提供します。

表13.1 レポートおよびトラブルシューティングのログファイルディレクトリー

ログファイルディレクトリー	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
<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>	設定管理
<code>/var/log/rhsm</code>	サブスクリプションの管理
<code>/var/log/tomcat</code>	Candlepin Web サービスログ

`foreman-tail` コマンドを使用して、Satellite に関連する多くのログファイルを追跡することもできます。 `foreman-tail -l` を実行すると、追跡するプロセスとサービスがリストされます。

13.8. ログ情報の収集ユーティリティー

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

収集プロセスでは、情報の収集中にパスワード、トークン、キーなどのセキュリティ情報が削除されます。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。Red Hat では、この情報をパブリックではなく特定の受信者に直接送信することを推奨します。

13.9. システムジャーナルメタデータ

次の表に、Satellite で `journald` 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。

表13.2 システムジャーナルメタデータ

名前	説明
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

名前	説明
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14章 外部認証の設定

外部認証を使用して、外部 ID プロバイダーのユーザーグループメンバーシップからユーザーとユーザーグループのパーミッションを派生させることができます。外部認証を使用する場合には、このようなユーザーを作成したり、グループメンバーシップを Satellite Server で手動で保守したりする必要はありません。外部ソースでメールが提供されない場合は、Satellite Web UI で最初のログイン時に要求されます。

重要なユーザーおよびグループアカウント情報

ユーザーおよびグループアカウントはすべて、ローカルアカウントである必要があります。これにより、Satellite Server 上のローカルアカウントと Active Directory ドメイン内のアカウントによる認証競合が避けられます。

ユーザーおよびグループアカウントが `/etc/passwd` と `/etc/group` ファイルの両方に存在すれば、この競合によってシステムが影響を受けることはありません。たとえば、**puppet**、**apache**、**foreman** および **foreman-proxy** グループのエントリーが `/etc/passwd` と `/etc/group` の両ファイルに存在することを確認するには、以下のコマンドを実行します。

```
# cat /etc/passwd | grep 'puppet\|apache\|foreman\|foreman-proxy'  
# cat /etc/group | grep 'puppet\|apache\|foreman\|foreman-proxy'
```

外部認証の設定シナリオ

Red Hat Satellite では、外部認証の設定において以下の一般的なシナリオがサポートされます。

- **Lightweight Directory Access Protocol (LDAP)** サーバーを外部 ID プロバイダーとして使用するシナリオ。LDAP は、一元的に保存された情報にネットワークを介してアクセスするために使用されるオープンプロトコルセットです。Satellite では、Satellite Web UI を介して LDAP 全体を管理できます。詳細は、[「LDAP の使用」](#) を参照してください。LDAP を使用して Red Hat Identity Management または AD サーバーに接続できませんが、セットアップでは、Satellite の Web UI でのサーバー検出、フォレスト間信頼、または Kerberos を使用したシングルサインオンはサポートされません。
- Red Hat Identity Management サーバーを外部 ID プロバイダーとして使用するシナリオ。Red Hat Identity Management は、ネットワーク環境で使用される個別 ID、認証情報、および権限を管理します。Red Hat Identity Management を使用した設定は、Satellite Web UI のみを使用して完了できず、CLI との対話が必要です。詳細は、[「Red Hat Identity Management の使用」](#) を参照してください。
- フォレスト間 Kerberos 信頼を介して Red Hat Identity Management に統合された **Active Directory (AD)** を外部 ID プロバイダーとして使用するシナリオ。詳細は、[「フォレスト間信頼を使用する Active Directory」](#) を参照してください。
- Red Hat Single Sign On を Satellite への外部認証用の OpenID プロバイダーとして使用するシナリオ。詳細は、[「Red Hat Single Sign On 認証を使用した Satellite の設定」](#) を参照してください。
- TOTP を使用した Satellite への外部認証に Red Hat Single Sign-On を OpenID プロバイダーとして使用するシナリオ。詳細は、[「TOTP での Red Hat Single Sign On 認証の設定」](#) を参照してください。

Satellite でプロビジョニングしたホストは、Satellite Server にアクセスできるだけでなく、Red Hat Identity Management レルムと統合することもできます。Red Hat Satellite には、レルムまたはドメインプロバイダーに登録されたシステムのライフサイクルを自動的に管理するレルム機能があります。詳細は、[「プロビジョニングされたホストの外部認証」](#) を参照してください。

表14.1 認証の概要

タイプ	認証	ユーザーグループ
Red Hat Identity Management	Kerberos または LDAP	あり
Active Directory	Kerberos または LDAP	あり
POSIX	LDAP	あり

14.1. LDAP の使用

Satellite は、1つまたは複数の LDAP ディレクトリーを使用した LDAP 認証をサポートします。

Red Hat Satellite で **TLS** を使用してセキュアな LDAP 接続 (LDAPS) を確立する必要がある場合は、まず、接続先の LDAP サーバーで使用する証明書を取得して、以下の説明のように Satellite Server のベースオペレーティングシステムでこの証明書を信頼済みとしてマークします。LDAP サーバーで中間認証局との証明書チェーンを使用する場合は、すべての証明書が取得されるように、チェーン内のすべてのルートおよび中間証明書が信頼済みである必要があります。この時点でセキュアな LDAP を必要としない場合は、「[Red Hat Satellite で LDAP を使用する設定](#)」に進みます。

SSSD 設定の使用

このセクションでは、LDAP の直接統合について説明しますが、Red Hat では、SSSD を使用して Red Hat Identity Management、AD、または LDAP サーバーに設定することを推奨します。SSSD により、認証プロセスの一貫性が向上されます。推奨設定の詳細は、「[Active Directory の使用](#)」を参照してください。SSSD 認証情報をキャッシュして LDAP 認証に使用することもできます。SSSD の詳細は、[Red Hat Enterprise Linux 7 System-Level Authentication Guide](#)の [Configuring SSSD](#) を参照してください。



重要

ユーザーは、Red Hat Identity Management と LDAP の両方を認証方法として使用することはできません。ユーザーが1つの方法を使用して認証されると、他の方法を使用することはできません。

ユーザーの認証方法を変更するには、自動的に作成されたユーザーを Satellite から削除する必要があります。

Red Hat Identity Management を認証方法として使用する方法の詳細については、「[Red Hat Identity Management の使用](#)」を参照してください。

14.1.1. セキュア LDAP 向けの TLS の設定

Satellite CLI を使用して、セキュア LDAP (LDAPS) 向けに TLS を設定します。

手順

- LDAP サーバーから証明書を取得します。
 - Active Directory 証明書サービスを使用する場合は、ベース 64 エンコード X.509 形式を使用してエンタープライズ PKI CA 証明書をエクスポートします。Active Directory サーバーでの CA 証明書の作成およびエクスポートについては、[How to configure Active Directory](#)

[authentication with TLS on Satellite](#) を参照してください。

- b. LDAP サーバー証明書を一時的な場所の Satellite Server にダウンロードし、終了したら削除します。
たとえば、`/tmp/example.crt` です。ファイル名の拡張子を `.cer` と `.crt` にすることが唯一の規則であり、この拡張子は、DER バイナリーまたは PEM ASCII の形式の証明書を参照できます。
2. LDAP サーバーからの証明書を信頼します。
Satellite Server では、LDAP 認証用の CA 証明書は `/etc/pki/tls/certs/` ディレクトリー内の個別ファイルである必要があります。

- a. **install** コマンドを使用して適切なパーミッションでインポート済みの証明書を `/etc/pki/tls/certs/` ディレクトリーにインストールします。

```
# install /tmp/example.crt /etc/pki/tls/certs/
```

- b. **root** で以下のコマンドを実行して、LDAP サーバーから取得した `example.crt` 証明書を信頼します。

```
# ln -s example.crt /etc/pki/tls/certs/$(openssl \
x509 -noout -hash -in \
/etc/pki/tls/certs/example.crt).0
```

- c. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

14.1.2. Red Hat Satellite で LDAP を使用する設定

Satellite Web UI で、LDAP を使用するように Satellite を設定します。

Satellite web UI で Kerberos を使用したシングルサインオン機能が必要な場合は、代わりに Red Hat Identity Management および AD 外部認証を使用する必要があることに注意してください。詳細は以下を参照してください。

- [「Red Hat Identity Management の使用」](#)
- [「Active Directory の使用」](#)

手順

1. Network Information System (NIS) サービスのブール値を `true` に設定して SELinux により LDAP の送信接続がブロックされないようにします。

```
# setsebool -P nis_enabled on
```

2. Satellite Web UI で、**Administer > Authentication Sources** に移動します。
3. **Create LDAP Authentication Source** をクリックします。
4. **LDAP サーバー** タブで LDAP サーバーの名前、ホスト名、ポート、およびサーバータイプを入力します。デフォルトポートは 389、デフォルトサーバータイプは POSIX です (認証サーバーのタイプに応じて FreeIPA または Active Directory を選択することもできます)。TLS 暗号化接

続に対しては、**LDAPS** チェックボックスを選択して暗号化を有効にします。ポートは LDAPS のデフォルト値である 636 に変更されるはずですが。

5. **アカウント** タブで、アカウント情報とドメイン名の詳細を入力します。説明と例については、「[LDAP 設定の説明](#)」を参照してください。
6. **属性マッピング** タブで、LDAP 属性を Satellite 属性にマッピングします。ログイン名、名、姓、メールアドレス、および写真の属性をマッピングできます。サンプルについては、「[LDAP 接続の設定例](#)」を参照してください。
7. **ロケーション** タブで、左側の表からロケーションを選択します。選択したロケーションは、LDAP 認証ソースから作成されたユーザーに割り当てられ、初回ログイン以降、利用可能となります。
8. **組織** タブで、左側の表から組織を選択します。選択した組織は、LDAP 認証ソースから作成されたユーザーに割り当てられ、初回ログイン以降、利用可能となります。
9. **Submit** をクリックします。
10. LDAP ユーザーの新しいアカウントを設定します。
 - **Automatically Create Accounts In Satellite**のチェックボックスを選択していない場合は、「[ユーザーの作成](#)」を参照してユーザーアカウントを手動で作成してください。
 - **Satellite でアカウントを自動作成する**のチェックボックスを選択した場合は、LDAP ユーザーは LDAP アカウントおよびパスワードを使用して Satellite にログインできます。初回ログイン後に、Satellite 管理者はロールを手動で割り当てる必要があります。Satellite でユーザーアカウントに適切なロールを割り当てる方法は、「[ユーザーへのロールの割り当て](#)」を参照してください。

14.1.3. LDAP 設定の説明

以下の表は、**アカウント** タブの各設定の説明を示しています。

表14.2 アカウントタブの設定

設定	説明
アカウント	<p>LDAP サーバーへの読み取りアクセス権のある LDAP アカウントのユーザー名。ユーザー名は、サーバーで匿名の読み取りが許可されている場合は必要ありません。以下に例を示します。</p> <pre>uid=\$login,cn=users,cn=accounts,dc=example,dc=com</pre> <p>\$login 変数には、ログインページで入力されたユーザー名がリテラル文字列として格納されます。この値は、変数がデプロイメントされたときにアクセスされます。</p> <p>この変数は、LDAP ソースからの外部ユーザーグループとは使用できません。ユーザーがログインしていない場合、Satellite はグループリストを取得する必要があります。匿名または専用サービスユーザーを使用してください。</p>
アカウントパスワード	<p>アカウント フィールドで定義されたユーザーの LDAP パスワード。アカウント が \$login 変数を使用している場合は、このフィールドを空白にすることができます。</p>

設定	説明
ベース DN	LDAP ディレクトリーの最上位のドメイン名。
グローバルベース DN	グループが含まれる LDAP ディレクトリーツリーの最上位のドメイン名。
LDAP フィルター	LDAP クエリーを制限するフィルター。
Satellite でアカウントを自動作成する	このチェックボックスを選択した場合には、LDAP ユーザーが Satellite に最初にログインしたときに、Satellite によりユーザーアカウントが作成されます。初回ログイン後に、Satellite 管理者はロールを手動で割り当てる必要があります。Satellite でユーザーアカウントに適切なロールを割り当てる方法は、「 ユーザーへのロールの割り当て 」を参照してください。
ユーザーグループの同期	このオプションが選択された場合は、ユーザーのログイン時にユーザーのユーザーグループメンバーシップが自動的に同期され、メンバーシップは常に最新の状態になります。このオプションが選択されていない場合は、Satellite で cron ジョブを使用してグループメンバーシップを定期的 (デフォルトでは 30 分ごと) に同期します。詳細は、「 外部ユーザーグループの設定 」を参照してください。

14.1.4. LDAP 接続の設定例

以下の表は、異なる種類の LDAP 接続の設定例を示しています。以下の例では、ユーザーおよびグループのエントリーに対してバインド、読み取り、および検索のパーミッションを持つ **redhat** という名前の専用サービスアカウントを使用します。LDAP 属性名は、大文字と小文字が区別されることに注意してください。

表14.3 Active Directory、Free IPA または Red Hat Identity Management、POSIX LDAP 接続の設定例

設定	Active Directory	FreeIPA または Red Hat Identity Management	POSIX (OpenLDAP)
アカウント	DOMAIN\redhat	uid=redhat,cn=users, cn=accounts,dc=example, dc=com	uid=redhat,ou=users, dc=example,dc=com
アカウントパスワード	P@ssword	-	-
ベース DN	DC=example,DC=COM	dc=example,dc=com	dc=example,dc=com
グループベース DN	CN=Users,DC=example,DC=com	cn=groups,cn=accounts, dc=example,dc=com	cn=employee,ou=userclass, dc=example,dc=com
ログイン名属性	userPrincipalName	uid	uid
名属性	givenName	givenName	givenName

設定	Active Directory	FreeIPA または Red Hat Identity Management	POSIX (OpenLDAP)
姓属性	sn	sn	sn
メールアドレス属性	mail	mail	mail
写真属性	thumbnailPhoto	-	-



注記

userPrincipalName では、ユーザー名に空白文字を使用できます。ログイン名属性 **sAMAccountName** (上記の表にはリストされていない) は、レガシーの Microsoft システムとの後方互換性を提供します。**sAMAccountName** では、ユーザー名に空白文字を使用できません。

14.1.5. LDAP フィルターの例

管理者は LDAP フィルターを作成することで、特定のユーザーの Satellite へのアクセスを制限することができます。

表14.4 特定ユーザーのログインを許可するフィルターの例

ユーザー	フィルター
User1	(distinguishedName=cn=User1,cn=Users,dc=domain,dc=example)
User1、 User3	(memberOf=cn=Group1,cn=Users,dc=domain,dc=example)
User2、 User3	(memberOf=cn=Group2,cn=Users,dc=domain,dc=example)
User1、 User2、 User3	((memberOf=cn=Group1,cn=Users,dc=domain,dc=example) (memberOf=cn=Group2,cn=Users,dc=domain,dc=example))
User1、 User2、 User3	(memberOf:1.2.840.113556.1.4.1941:=cn=Users,dc=domain,dc=example)



注記

グループの **Users** は、グループ **Group1** と **Group2** を含むネストされたグループです。ネストされたグループからすべてのユーザーをフィルターする場合は、ネストされたグループ名の前に **memberOf:1.2.840.113556.1.4.1941:=** を追加する必要があります。上の表の最後の例を参照してください。

LDAP ディレクトリー構造

上記の例のフィルターで使用される LDAP ディレクトリー構造

DC=Domain,DC=Example

|

```

|----- CN=Users
|
|----- CN=Group1
|----- CN=Group2
|----- CN=User1
|----- CN=User2
|----- CN=User3

```

LDAP グループメンバーシップ

上記の例のフィルターで 사용되는グループメンバーシップ

グループ	メンバー
Group1	User1, User3
Group2	User2, User3

14.2. RED HAT IDENTITY MANAGEMENT の使用

本項では、Satellite Server と Red Hat Identity Management サーバーを統合する方法とホストベースアクセス制御を有効にする方法を示します。



注記

Red Hat Identity Management は、外部認証ソースとして、シングルサインオンサポートなしで接続できます。詳細は、[「LDAP の使用」](#) を参照してください。



重要

ユーザーは、Red Hat Identity Management と LDAP の両方を認証方法として使用することはできません。ユーザーが1つの方法を使用して認証されると、他の方法を使用することはできません。

ユーザーの認証方法を変更するには、自動的に作成されたユーザーを Satellite から削除する必要があります。

前提条件

- Satellite Server のベースオペレーティングシステムが、組織の Red Hat Identity Management 管理者によって Red Hat Identity Management ドメインに登録されていること。

この章の例では、Red Hat Identity Management と Satellite の設定が分離されていることを前提としています。ただし、両方のサーバーの管理者権限がある場合は、[Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy Guide](#) の説明に従って Red Hat Identity Management を設定できます。

14.2.1. Satellite Server での Red Hat Identity Management 認証の設定

Satellite CLI で、まず Red Hat Identity Management サーバーにホストエントリーを作成して、Red Hat Identity Management 認証を設定します。

手順

1. Red Hat Identity Management サーバーで、次のコマンドを入力し、プロンプトが表示されたら、パスワードを入力して、認証します。

```
# kinit admin
```

2. 認証されたことを確認するには、次のコマンドを入力します。

```
# klist
```

3. 以下のように、Red Hat Identity Management サーバー上で Satellite Server のホストエントリを作成し、ワンタイムパスワードを生成します。

```
# ipa host-add --random hostname
```



注記

Red Hat Identity Management 登録を完了するには、生成されたワンタイムパスワードをクライアントで使用する必要があります。

ホスト設定プロパティの詳細は、[Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy](#) の [About Host Entry Configuration Properties](#) を参照してください。

4. 以下のように、Satellite Server 向けの HTTP サービスを作成します。

```
# ipa service-add HTTP/hostname
```

サービス管理の詳細については、[Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy](#) ガイドの [Managing Services](#) を参照してください。

5. Satellite Server で、IPA クライアントをインストールします。



警告

このコマンドは、パッケージのインストール中に Satellite サービスを再起動する可能性があります。Satellite でのパッケージのインストールと更新に関する詳細は、「[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](#)」を参照してください。

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

6. Satellite Server で、以下のコマンドを root として入力し、Red Hat Identity Management 登録を設定します。

```
# ipa-client-install --password OTP
```

OTP を、Red Hat Identity Management 管理者により提供されたワンタイムパスワードに置き換えます。

7. Satellite Server が Red Hat Enterprise Linux 7 で実行されている場合は、以下のコマンドを実行します。

```
# subscription-manager repos --enable rhel-7-server-optional-rpms
```

インストーラーは、オプションのリポジトリ **rhel-7-server-optional-rpms** (Red Hat Enterprise Linux 7 の場合) に含まれるパッケージに依存します。

8. 以下のコマンドを使用して、**foreman-ipa-authentication** を true に設定します。

```
# satellite-installer --foreman-ipa-authentication=true
```

9. Satellite サービスを再起動します。

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

この時点で、外部ユーザーは Red Hat Identity Management 認証情報を使用して Satellite にログインできます。ユーザー名とパスワードを使用して直接 Satellite Server にログインするか、設定済みの Kerberos シングルサインオンを活用してクライアントマシンでチケットを取得し、自動的にログインするかを選択できます。また、ワンタイムパスワードを使用した二要素認証 (2FA OTP) もサポートされます。Red Hat Identity Management 内のユーザーが 2FA 向けに設定され、Satellite Server が Red Hat Enterprise Linux 7 で実行されている場合には、このユーザーは OTP を使用して Satellite に対して認証することもできます。

14.2.2. ホストベースの認証制御の設定

HBAC ルールでは、Red Hat Identity Management ユーザーがドメイン内のどのマシンにアクセスできるかを定義します。一部のユーザーが Satellite Server にアクセスできないように、Red Hat Identity Management サーバーで HBAC を設定できます。この方法では、ログインが許可されていないユーザーのデータベースエントリを、Satellite で作成できないようにします。HBAC の詳細は、**Red Hat Enterprise Linux 7 Linux Domain Identity, Authentication, and Policy** ガイドの [Configuring Host-Based Access Control](#) を参照してください。

Red Hat Identity Management サーバーで、ホストベースの認証制御 (HBAC) を設定します。

手順

1. Red Hat Identity Management サーバーで、次のコマンドを入力し、プロンプトが表示されたら、パスワードを入力して、認証します。

```
# kinit admin
```

2. 認証されたことを確認するには、次のコマンドを入力します。

```
# klist
```

3. HBAC サービスおよびルールを Red Hat Identity Management サーバーで作成し、リンクします。以下の例では、**satellite-prod** という PAM サービス名を使用しています。Red Hat Identity Management サーバー上で以下のコマンドを実行してください。

```
# ipa hbacsvc-add satellite-prod
# ipa hbacrule-add allow_satellite_prod
# ipa hbacrule-add-service allow_satellite_prod --hbacsvcs=satellite-prod
```

4. satellite-prod サービスへのアクセス権があるユーザーと Satellite Server のホスト名を追加します。

```
# ipa hbacrule-add-user allow_satellite_prod --user=username
# ipa hbacrule-add-host allow_satellite_prod --hosts=satellite.example.com
```

または、`allow_satellite_prod` ルールにホストグループとユーザーグループを追加できます。

5. ルールのステータスを確認するために、以下のコマンドを実行します。

```
# ipa hbacrule-find satellite-prod
# ipa hbactest --user=username --host=satellite.example.com --service=satellite-prod
```

6. Red Hat Identity Management サーバーで `allow_all` ルールが無効であることを確認します。他のサービスに影響を与えずにこのルールを無効にする方法については、Red Hat カスタマーポータルの記事 [How to configure HBAC rules in IdM to allow specific users to login to clients via ssh](#) を参照してください。
7. 「[Satellite Server での Red Hat Identity Management 認証の設定](#)」で説明されているように、Satellite Server で Red Hat Identity Management 統合を設定します。Satellite Server で、root として PAM サービスを定義します。

```
# satellite-installer --foreman-pam-service=satellite-prod
```

14.3. ACTIVE DIRECTORY の使用

このセクションでは、Satellite Server 用の外部認証ソースとして直接 Active Directory (AD) を使用する方法を示します。



注記

シングルサインオンサポートなしで、Active Directory を外部認証ソースとして接続できません。詳細は、「[LDAP の使用](#)」を参照してください。設定例については、「[How to configure Active Directory authentication with TLS on Satellite](#)」を参照してください。

直接 AD 統合では、ID が保存されている AD ドメインに Satellite Server が直接参加します。推奨の設定には、以下の 2 つの手順が含まれます。

- 「[Satellite Server の AD サーバーへの登録](#)」の説明に従って、Active Directory サーバーに Satellite Server を登録します。
- 「[GSS-Proxy を使用した直接 AD 統合の設定](#)」の説明に従って、GSS-proxy との直接 Active Directory 統合を設定します。

14.3.1. GSS-Proxy

Apache での Kerberos 認証の従来のプロセスでは、Apache プロセスが keytab ファイルへの読み取りアクセスを持っている必要があります。GSS-Proxy を使用すると、Kerberos 認証機能を保持しつつ keytab ファイルへのアクセスを削除することにより Apache サーバーに対してより厳密な権限の分離を

実行できます。AD を Satellite の外部認証ソースとして使用する場合は、keytab ファイルのキーがホストキーと同じであるため、GSS-proxy を実装することが推奨されます。

Satellite Server のベースオペレーティングシステムとして動作する Red Hat Enterprise Linux で以下の手順を実行します。本セクションの例では、**EXAMPLE.ORG** が AD ドメインの Kerberos レalm です。手順を完了すると、EXAMPLE.ORG レalm に属するユーザーは Satellite Server にログインできます。

14.3.2. Satellite Server の AD サーバーへの登録

Satellite CLI で、Active Directory サーバーに Satellite Server を登録します。

前提条件

- GSS-proxy と nfs-utils がインストールされていること。
GSS-proxy と nfs-utils をインストールします。

```
# satellite-maintain packages install gssproxy nfs-utils
```

手順

1. 必要なパッケージをインストールします。

```
# satellite-maintain packages install sssd adcli realmd ipa-python-compat krb5-workstation  
samba-common-tools
```

2. Satellite Server を AD サーバーに登録します。以下のコマンドを実行するには、管理者パーミッションが必要な場合があります。

```
# realm join -v EXAMPLE.ORG --membership-software=samba -U Administrator
```

14.3.3. GSS-Proxy を使用した直接 AD 統合の設定

Satellite CLI で、GSS-proxy を使用する直接 Active Directory 統合を設定します。

前提条件

- Satellite が、Active Directory サーバーに登録されていること。詳細は、[「Satellite Server の AD サーバーへの登録」](#) を参照してください。

手順

1. **/etc/ipa/** ディレクトリーと **default.conf** ファイルを作成します。

```
# mkdir /etc/ipa  
# touch /etc/ipa/default.conf
```

2. **default.conf** ファイルに以下のコンテンツを追加します。

```
[global]  
server = unused  
realm = EXAMPLE.ORG
```

3. 以下の内容で `/etc/net-keytab.conf` ファイルを作成します。

```
[global]
workgroup = EXAMPLE
realm = EXAMPLE.ORG
kerberos method = system keytab
security = ads
```

4. Apache ユーザーの有効なユーザー ID を特定します。

```
# id apache
```

Apache ユーザーには keytab ファイルへのアクセス権を割り当てないでください。

5. 以下の内容で `/etc/gssproxy/00-http.conf` ファイルを作成します。

```
[service/HTTP]
mechs = krb5
cred_store = keytab:/etc/httpd/conf/http.keytab
cred_store = ccache:/var/lib/gssproxy/clients/krb5cc_%U
euid = ID_of_Apache_User
```

6. keytab エントリーを作成します。

```
# KRB5_KTNAME=FILE:/etc/httpd/conf/http.keytab net ads keytab add HTTP -U
administrator -d3 -s /etc/net-keytab.conf
# chown root.apache /etc/httpd/conf/http.keytab
# chmod 640 /etc/httpd/conf/http.keytab
```

7. Satellite で IPA 認証を有効にします。

```
# satellite-installer --foreman-ipa-authentication=true
```

8. **gssproxy** サービスを起動して、有効にします。

```
# systemctl restart gssproxy.service
# systemctl enable gssproxy.service
```

9. Apache サーバーが **gssproxy** サービスを使用するように設定するには、**systemd** ドロップインファイルを作成し、以下の内容を追加します。

```
# mkdir -p /etc/systemd/system/httpd.service.d/
# vi /etc/systemd/system/httpd.service.d/gssproxy.conf
[Service]
Environment=GSS_USE_PROXY=1
```

10. 変更をサービスに適用します。

```
# systemctl daemon-reload
```

11. **httpd** サービスを起動して、有効にします。

```
# systemctl restart httpd.service
```

重要

直接 AD 統合では、Red Hat Identity Management を介した HBAC は利用できません。代わりに、管理者が AD 環境でポリシーを一元管理することを可能にする Group Policy Objects (GPO) を使用できます。GPO から PAM サービスへのマッピングが正しいことを確認するには、以下の SSSD 設定を `/etc/sss/sss.conf` に追加します。

```
access_provider = ad
ad_gpo_access_control = enforcing
ad_gpo_map_service = +foreman
```

ここでは、`foreman` は PAM サービスの名前です。GPO の詳細は、[RHEL システムと Windows Active Directory を直接統合の SSSD が GPO アクセス制御ルールを解釈する方法](#) を参照してください。

検証

SSO が想定どおりに動作していることを確認します。

Apache サーバーが実行中であり、クライアントに有効な Kerberos チケットがある場合、サーバーに対して HTTP 要求を行うユーザーは認証されます。

1. 次のコマンドを使用して、LDAP ユーザーの Kerberos チケットを取得します。

```
# kinit ldapuser
```

2. 以下のコマンドを使用して、Kerberos チケットを表示します。

```
# klist
```

3. 以下のコマンドを使用して、SSO 認証に成功時の出力を表示します。

```
# curl -k -u : --negotiate https://satellite.example.com/users/extlogin
```

これにより、以下の応答が返されます。

```
<html><body>You are being <a href="https://satellite.example.com/users/4-ldapuserexample-com/edit">redirected</a>.</body></html>
```

14.3.4. Web ブラウザーでの Kerberos の設定

Firefox ブラウザーの設定の詳細は、[Red Hat Enterprise Linux System-Level Authenticationの Configuring Firefox to Use Kerberos for Single Sign-On](#) を参照してください。

Internet Explorer ブラウザーを使用している場合は、Satellite Server をローカルイントラネットまたは信頼済みサイトのリストに追加し、[統合 Windows 認証を使用する](#) の設定にチェックを入れます。詳細については、Internet Explorer のマニュアルを参照してください。

14.3.5. フォレスト間信頼を使用する Active Directory

Kerberos では、**cross-forest trust** を作成して、2つの異なるドメインフォレスト間の関係を定義できます。ドメインフォレストとは、ドメインの階層構造のことで、AD と Red Hat Identity Management

の両方でフォレストが形成されます。AD と Red Hat Identity Management との間での有効な信頼関係により、AD のユーザーは一連の認証情報を使用して Linux ホストおよびサービスにアクセスできます。フォレスト間信頼の詳細は、[Red Hat Enterprise Linux Windows Integration の Creating Cross-forest Trusts with Active Directory and Identity Management](#) を参照してください。

Satellite 側から見ると、設定プロセスは、フォレスト間の信頼を設定せずに Red Hat Identity Management サーバーと統合する場合と同じです。Satellite Server は IdM ドメインに登録し、「[Red Hat Identity Management の使用](#)」で説明されているように統合する必要があります。

14.3.6. フォレスト間信頼を使用するための Red Hat Identity Management サーバーの設定

Red Hat Identity Management サーバーで、**cross-forest trust** を使用するようにサーバーを設定します。

手順

1. HBAC を有効にします。
 - a. 外部グループを作成して、この外部グループに AD グループを追加します。
 - b. 新しい外部グループを POSIX グループに追加します。
 - c. HBAC ルールで POSIX グループを使用します。
2. AD ユーザーの追加属性を転送するよう sssd を設定します。
 - この AD ユーザー属性を `/etc/sss/sss.conf` の `nss` セクションと `domain` セクションに追加します。以下に例を示します。

```
[nss]
user_attributes=+mail, +sn, +givenname
[domain/EXAMPLE.com]
...
krb5_store_password_if_offline = True
ldap_user_extra_attrs=email:mail, lastname:sn, firstname:givenname

[ifp]
allowed_uids = ipaapi, root
user_attributes=+email, +firstname, +lastname
```

- AD 属性値を確認します。

```
# dbus-send --print-reply --system --dest=org.freedesktop.sssd.infopipe
/org/freedesktop/sss/infopipe org.freedesktop.sssd.infopipe.GetUserAttr string:ad-
user@ad-domain array:string:email,firstname,lastname
```

14.4. 外部ユーザーグループの設定

Satellite は、自動的に、外部ユーザーグループに外部ユーザーを関連付けることはありません。Satellite 上の外部ソースと同じ名前のユーザーグループを作成する必要があります。こうすることで、外部ユーザーグループのメンバーは、自動的に Satellite ユーザーグループのメンバーになり、関連するパーミッションが付与されます。

外部ユーザーグループの設定は、外部認証の種類によって異なります。

外部ユーザーに追加のパーミッションを割り当てるには、外部マッピングが指定されていない内部ユーザーグループに、このユーザーを追加します。次に、このグループに必要なロールを割り当てます。

前提条件

- LDAP サーバーを使用する場合は、Satellite が LDAP 認証を使用するように設定する。詳細は、「[LDAP の使用](#)」を参照してください。
LDAP ソースから外部ユーザーグループを使用する場合は、アカウントユーザー名の代わりにとして `$login` 変数を使用できず、匿名または専用サービスユーザーを使用する。
- Red Hat Identity Management または AD サーバーを使用する場合は、Satellite が Red Hat Identity Management または AD 認証を使用するように設定する。詳細は、[14章 外部認証の設定](#)を参照してください。
- 少なくとも1人の外部ユーザーが初回認証されることを確認する。
- 使用する外部グループ名をメモする。外部ユーザーのグループメンバーシップを確認するには、以下のコマンドを入力します。

```
# id username
```

手順

1. Satellite Web UI で、**管理 > ユーザーグループ** に移動して、**ユーザーグループの作成** をクリックします。
2. 新規ユーザーグループの名前を指定します。外部ユーザーグループのリフレッシュ時にユーザーが自動的に追加されるのを避けるため、ユーザーを選択しないでください。
3. **ロール** タブをクリックし、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** の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。
4. **外部グループ** タブで、**外部ユーザーグループの追加** をクリックして、**認証ソース** ドロップダウンメニューから認証ソースを選択します。
名前 フィールドに外部グループの名前を指定します。
5. **Submit** をクリックします。

14.5. LDAP の外部ユーザーグループのリフレッシュ

ユーザーのログイン時にユーザーのグループメンバーシップを自動的に同期するように LDAP ソースを設定するには、**認証ソース** ページで、**ユーザーグループの同期** オプションを選択します。このオプションが選択されていない場合、デフォルトで、LDAP ユーザーグループは、30 分ごとに LDAP 認証ソースを同期するようにスケジュールされた cron ジョブで自動的にリフレッシュされます。

LDAP 認証ソースのユーザーグループが、次のスケジュールタスクが実行されるまでの間に変更された場合に、ユーザーが不正な外部ユーザーグループに割り当てられることがあります。これはスケジュールされたタスクの実行時に、自動的に修正されます。

以下の手順を使用して、LDAP ソースを手動でリフレッシュします。

手順

1. Satellite web UI で、**Administer > Usergroups** に移動し、ユーザーグループを選択します。

2. **External Groups** タブで、必要なユーザーグループの右側にある **Refresh** をクリックします。

CLI 手順

- 以下のコマンドを入力します。

```
# foreman-rake ldap:refresh_usergroups
```

14.6. RED HAT IDENTITY MANAGEMENT または AD の外部ユーザーグループの更新

Red Hat Identity Management または AD ベースの外部ユーザーグループは、グループメンバーが Satellite にログインしたときのみリフレッシュされます。Satellite Web UI で、外部ユーザーグループのユーザーメンバーシップを変更することはできず、このような変更がされた場合には、次のグループリフレッシュ時に上書きされます。

14.7. プロビジョンされたホストの外部認証

以下のセクションを使用して、Red Hat Identity Management レルムサポート用の Satellite Server または Capsule Server を設定します。続いて、Red Hat Identity Management レルムグループにホストを追加します。

前提条件

- Satellite Server をコンテンツ配信ネットワークに登録しておくか、外部の Capsule Server を Satellite Server に登録しておく。
- Red Hat Identity Management などのレルムまたはドメインプロバイダーがデプロイされていること。

Satellite Server または Capsule Server に Identity Management パッケージをインストールして設定するには:

プロビジョニングされたホストに Identity Management を使用するには、次の手順を実行して、Satellite Server または Capsule Server に Identity Management パッケージをインストールおよび設定します。

1. Satellite Server または Capsule Server に **ipa-client** パッケージをインストールします。

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

2. サーバーを Red Hat Identity Management クライアントとして設定します。

```
# ipa-client-install
```

3. Red Hat Identity Management でレルムプロキシユーザー **realm-capsule** と、関連のロールを作成します。

```
# foreman-prepare-realm admin realm-capsule
```

以下の手順で必要となるので、返されたプリンシパル名と、Red Hat Identity Management サーバーの設定情報をメモします。

Red Hat Identity Management レルムサポート用の Red Hat Satellite Server または Capsule Server の設定方法

使用する Satellite および全 Capsule で次の手順を実行します。

1. 同じプリンシパルおよびレルムに追加する Capsule Server に、`/root/freeipa.keytab` ファイルをコピーします。

```
# scp /root/freeipa.keytab root@capsule.example.com:/etc/foreman-proxy/freeipa.keytab
```

2. `/root/freeipa.keytab` ファイルを `/etc/foreman-proxy` ディレクトリーに移動して、所有者を `foreman-proxy` ユーザーに設定します。

```
# mv /root/freeipa.keytab /etc/foreman-proxy
# chown foreman-proxy:foreman-proxy /etc/foreman-proxy/freeipa.keytab
```

3. レルムに追加する全 Capsule で、以下のコマンドを入力します。Satellite に統合された Capsule を使用する場合には、Satellite Server でこのコマンドを入力します。

```
# satellite-installer --foreman-proxy-realm true \
--foreman-proxy-realm-keytab /etc/foreman-proxy/freeipa.keytab \
--foreman-proxy-realm-principal realm-capsule@EXAMPLE.COM \
--foreman-proxy-realm-provider freeipa
```

これらのオプションは、Satellite Server を初めて設定する場合にも使用できます。

4. `ca-certificates` パッケージの最新バージョンがインストールされていることを確認し、Red Hat Identity Management 認証局を信頼します。

```
# cp /etc/ipa/ca.crt /etc/pki/ca-trust/source/anchors/ipa.crt
# update-ca-trust enable
# update-ca-trust
```

5. オプション: 既存の Satellite Server または Capsule Server で Red Hat Identity Management を設定する場合には、以下の手順を実行して、設定の変更が適用されていることを確認します。

- a. `foreman-proxy` サービスを再起動します。

```
# systemctl restart foreman-proxy
```

- b. Satellite Web UI で、**Infrastructure > Capsules** に移動します。

- c. Red Hat Identity Management 用に設定した Capsule の場所を特定して、**アクション** コラムのリストから **リフレッシュ** を選択します。

Red Hat Identity Management 対応の Capsule のレルムの作成方法

統合型または外部の Capsule に Red Hat Identity Management を設定した後に、レルムを作成して、Red Hat Identity Management が設定された Capsule をレルムに追加する必要があります。

手順

1. Satellite Web UI で、**インフラストラクチャー > レルム** に移動して、**レルムの作成** をクリックします。

2. **名前** フィールドには、レルムの名前を入力します。
3. **レルムのタイプ** リストから、レルムのタイプを選択します。
4. **Realm Capsule** リストから、Red Hat Identity Management を設定した Capsule Server を選択します。
5. **ロケーション** タブをクリックして、**ロケーション** リストから、新しいレルムを追加するロケーションを選択します。
6. **組織** タブをクリックして、**組織** リストから、新規レルムを追加する組織を選択します。
7. **Submit** をクリックします。

レルム情報でのホストグループの更新

使用するホストグループを、新しいレルム情報で更新する必要があります。

1. Satellite web UI で、**Configure** > **Host Groups** に移動し、更新するホストグループを選択して、**Network** タブをクリックします。
2. **レルム** リストから、この手順の一部で作成するレルムを選択して **送信** をクリックします。

Red Hat Identity Management ホストグループへのホストの追加

Red Hat Identity Management では、システムの属性に基づいて自動メンバーシップルールをセットアップできます。Red Hat Satellite のレルム機能は、管理者に対し、Red Hat Satellite ホストグループを Red Hat Identity Management パラメーター **userclass** にマップする機能を提供します。これにより、管理者は automembership を設定することができます。

ネスト化されたホストグループが使用される場合、それらは Red Hat Satellite ユーザーインターフェイスに表示され、Red Hat Identity Management サーバーに送信されます。たとえば、"Parent/Child/Child" のように表示されます。

Satellite Server または Capsule Server は更新を Red Hat Identity Management サーバーに送信しますが、automembership のルールは、初回登録時にのみ適用されます。

Red Hat Identity Management ホストグループにホストを追加するには、以下を実行します。

1. Red Hat Identity Management サーバーで、ホストグループを作成します。

```
# ipa hostgroup-add hostgroup_name --desc=hostgroup_description
```

2. **automembership** ルールを作成します。

```
# ipa automember-add --type=hostgroup hostgroup_name automember_rule
```

以下のオプションを使用できる場所:

- **automember-add** は automember グループとしてグループにフラグを立てます。
 - **--type=hostgroup** は、ターゲットグループがユーザーグループではなく、ホストグループであることを特定します。
 - **automember_rule** は、automember ルールの識別に使用する名前を追加します。
3. **userclass** 属性に基づいて automembership の条件を定義します。

```
# ipa automember-add-condition --key=userclass --type=hostgroup --inclusive-
regex=^webserver hostgroup_name
-----
Added condition(s) to "hostgroup_name"
-----
Automember Rule: automember_rule
Inclusive Regex: userclass=^webserver
-----
Number of conditions added 1
-----
```

以下のオプションを使用できる場所:

- **automember-add-condition** では、グループメンバーを識別する正規表現の条件を追加します。
- **--key=userclass** はキー属性を **userclass** に指定します。
- **--type=hostgroup** は、ターゲットグループがユーザーグループではなく、ホストグループであることを特定します。
- **--inclusive-regex= ^webserver** は、正規表現パターンで一致する値を識別します。
- **hostgroup_name**: ターゲットホストグループの名前を識別します。

システムが Satellite Server の **hostgroup_name** ホストグループに追加されると、そのシステムは、Red Hat Identity Management サーバーの "**hostgroup_name**" ホストグループに自動的に追加されます。Red Hat Identity Management ホストグループは、HBAC (ホストベースアクセス制御)、sudo ポリシー、およびその他の Red Hat Identity Management 機能を許可します。

14.8. RED HAT SINGLE SIGN ON 認証を使用した SATELLITE の設定

Red Hat Single Sign On を外部認証用の OpenID プロバイダーとして使用するよう Satellite を設定するには、以下のセクションを使用します。

14.8.1. Red Hat Single Sign On 認証を使用した Satellite の設定時の前提条件

Red Hat Single Sign-On 外部認証を使用して Satellite を設定する前に、以下の要件を満たすようにしてください。

- HTTP ではなく、HTTPS を使用する Red Hat Single Sign On サーバーを正常にインストールしている。
- 管理者権限を持つ Red Hat Single Sign-On アカウント。
- Red Hat Single Sign-On で作成した Satellite ユーザーアカウントのレلم。
- 証明書または CA が自己署名されている場合は、それらがエンドユーザー証明書トラストストアに追加されていることを確認する。
- ユーザーが Red Hat Single Sign-On にインポートまたは追加されている。
LDAP や Kerberos などの既存のユーザーデータベースが設定されている場合は、ユーザーのフェデレーションを設定することでユーザーをインポートできます。詳細は、[Red Hat Single Sign On サーバー管理ガイドの ユーザーストレージフェデレーション](#) を参照してください。

既存のユーザーデータベースが設定されていない場合は、Red Hat Single Sign-On でユーザーを手作業で作成できます。詳細は、[Red Hat Single Sign On サーバー管理ガイドの 新規ユーザーの作成](#) を参照してください。

14.8.2. Satellite の Red Hat Single Sign-On クライアントとして登録

以下の手順を使用して、Satellite をクライアントとして Red Hat Single Sign-On に登録し、認証ソースとして Red Hat Single Sign-On を使用するように Satellite を設定します。

Satellite と Red Hat Single Sign-On は、2 つの異なる認証方法で設定できます。

1. Satellite Web UI を使用した Satellite への認証。
2. Satellite CLI を使用した Satellite への認証。

どちらの方法でも、異なる Satellite クライアントを Red Hat Single Sign-On に登録して設定する必要があるため、ユーザーの認証方法を事前に決定する必要があります。この手順では、Red Hat Single Sign-On の Satellite クライアントの登録および設定方法が区別されています。

両認証方法を使用して、どちらのクライアントも適宜設定する場合には、Red Hat Single Sign-On に異なる Satellite クライアントを 2 つ登録することも可能です。

手順

1. Satellite Server で、以下のパッケージをインストールします。

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install
```

2. Satellite をクライアントとして Red Hat Single Sign-On に登録します。Web UI と CLI とでログインの登録プロセスが異なる点に注意してください。Red Hat Single Sign-On に 2 つの Satellite クライアントを登録すると、Web UI と CLI から Satellite にログインできます。

- Web UI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

次に、認証ソースとして Red Hat Single Sign On を使用するように Satellite を設定します。

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

- CLI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name hammer-openidc \
```

```
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

3. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

14.8.3. Red Hat Single Sign-On での Satellite クライアントの設定

以下の手順を使用して、Red Hat Single Sign-On Web UI で Satellite クライアントを設定し、Satellite クライアントのグループおよびオーディエンスマッパーを作成します。

手順

1. Red Hat Single Sign-On Web UI で、**クライアント** に移動し、Satellite クライアントをクリックします。
2. アクセスタイプを設定します。
 - Satellite web UI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **機密** を選択します。
 - CLI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **公開** を選択します。
3. **有効なリダイレクト URI** フィールドに有効なリダイレクト URI を追加します。

- Satellite web UI を使用して Satellite への認証を行うには、**https://satellite.example.com/users/extlogin** の形式で URI を入力します。Satellite FQDN の後に **/users/extlogin** の文字列を追加する必要があります。この手順の完了後に、Satellite クライアントが Satellite web UI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri
https://satellite.example.com/users/extlogin
```

- CLI を使用してユーザーが Satellite への認証を行うには、既存の URI の下の空白フィールドに **urn:ietf:wg:oauth:2.0:oob** を入力します。この手順の完了後に、Satellite クライアントが CLI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri
urn:ietf:wg:oauth:2.0:oob
```

4. **Save** をクリックします。
5. **マッパー** タブ、**作成** の順にクリックし、オーディエンスマッパーを追加します。

6. **名前** フィールドに、オーディエンスマッパーの名前を入力します。
7. **マッパータイプ** リストから、**オーディエンス** を選択します。
8. **組み込み済みクライアントオーディエンス** リストから、**Satellite クライアント** を選択します。
9. **Save** をクリックします。
10. **作成** をクリックして、**グループメンバーシップ** をもとに **Satellite** の認証を指定できるように **グループマッパー** を追加します。
11. **名前** フィールドに **グループマッパー** の名前を入力します。
12. **マッパータイプ** リストから、**グループメンバーシップ** を選択します。
13. **トークンクレーム名** に **groups** と入力します。
14. **フルグループパス** のトグルを **OFF** に設定します。
15. **Save** をクリックします。

14.8.4. Red Hat Single Sign-On 認証用の Satellite オプションの設定

このセクションでは、Satellite Web UI または CLI を使用して Red Hat Single Sign-On 認証用に Satellite を設定します。

14.8.4.1. Web UI を使用した Red Hat Single Sign-On 認証用の Satellite オプションの設定

以下の手順に従って、Satellite Web UI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **機密** に設定されていることを確認します。

手順

1. Satellite Web UI で、**管理 > 設定** に移動して、**認証** タブをクリックします。
2. **ログイン委任の認証** の行を見つけ、**値** コラムで **Yes** に値を設定します。
3. **Authorize login delegation auth source user autcreate** 行を見つけ、**値** コラムで **External** に値を設定します。
4. **Login delegation logout URL** の行を見つけ、**Value** 列で、値を <https://satellite.example.com/users/extlogout> に設定します。
5. **OIDC アルゴリズム** の行を見つけ、**値** コラムで、Red Hat Single Sign On のエンコーディングのアルゴリズムを設定します (例: **RS256**)。
6. **OIDC オーディエンス** 行を見つけ、**値** コラムで、値を Red Hat Single Sign On のクライアント ID に設定します。

7. **OIDC Issuer** 行を見つけ、**Value** 列で、値を https://RHSSO.example.com/auth/realms/Satellite_Realm に設定します。
8. **OIDC JWKs URL** 行を見つけ、**Value** 列で、値を https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs に設定します。
9. Satellite Web UI で、**Administer** > **Authentication Sources** に移動し、**External** カードの垂直省略記号をクリックして、**Edit** を選択します。
10. **場所** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる場所を追加します。
11. **組織** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる組織を追加します。
12. **Submit** をクリックします。

14.8.4.2. CLI を使用した Red Hat Single Sign-On 認証用の Satellite オプションの設定

以下の手順に従って、Satellite CLI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **公開** に設定されていることを確認します。

手順

1. Satellite で、ログイン委任を **true** に設定し、ユーザーが Open IDc プロトコルを使用して認証できるようにします。

```
# hammer settings set --name authorize_login_delegation --value true
```

2. ログイン委任のログアウト URL を以下のように設定します。

```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

3. Red Hat Single Sign On のエンコーディングのアルゴリズムを設定します (例: **RS256**)。

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

4. URL RHSSO.example.com/auth/realms/RHSSO_REALM/.well-known/openid-configuration を開いて値をメモし、以下のステップのオプションに入力します。

5. Open IDc オーディエンスに Hammer クライアントの値を追加します。

```
# hammer settings set --name oidc_audience \
--value "['satellite.example.com-hammer-openidc']"
```



注記

複数の Red Hat Single Sign-On クライアントを Satellite に登録する場合は、以下のように、アレイに全オーディエンスを必ず追加してください。以下に例を示します。

```
# hammer settings set --name oidc_audience \
--value "[satellite.example.com-foreman-openidc, 'satellite.example.com-
hammer-openidc']"
```

6. Open IDC 発行者の値を設定します。

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

7. Open IDC Java Web Token (JWT) の値を設定します。

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

8. Red Hat Single Sign-On 認証ソースの ID を取得します。

```
# hammer auth-source external list
```

9. ロケーションと組織を設定します。

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

14.8.5. Red Hat Single Sign-On を使用した Satellite web UI へのログイン

以下の手順に従って、Red Hat Single Sign-On を使用して Satellite Web UI にログインします。

手順

- ブラウザーで Satellite にログインし、認証情報を入力します。

14.8.6. Red Hat Single Sign-On を使用した Satellite CLI へのログイン

以下の手順に従って、コード付与タイプを使用して Satellite CLI への認証を行います。

手順

1. コード付与タイプを使用して Satellite CLI への認証を行うには、以下のコマンドを入力します。

```
# hammer auth login oauth \
--two-factor \
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-
connect/token' \
```

```
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \  
--oidc-client-id 'satellite.example.com-foreman-openidc' \  
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

このコマンドは、サクセスコードの入力を要求します。

2. サクセスコードを取得するには、コマンドが返す URL に移動し、必要な情報を提供します。
3. Web UI が返すサクセスコードをコピーします。
4. **hammer auth login oauth** のコマンドプロンプトに、サクセスコードを入力して Satellite CLI に対して認証を行います。

14.8.7. Red Hat シングルサインオン認証用のグループマッピングの設定

必要に応じて、ロールベースのアクセス制御 (RBAC) を実装するには、Satellite でグループを作成し、このグループにロールを割り当ててから Active Directory グループを Satellite グループにマッピングします。これにより、Red Hat Single Sign-On の指定のグループに所属する場合には、該当する Satellite グループでログインします。この例では、Active Directory の Satellite-admin ユーザーグループのユーザーを設定し、Satellite で管理者権限を持つユーザーとして認証されるようにします。

手順

1. Satellite Web UI で、**管理 > ユーザーグループ** に移動して、**ユーザーグループの作成** ボタンをクリックします。
2. **名前** フィールドにユーザーグループの名前を入力します。名前は Active Directory と同じにしないでください。
3. 右側の列には、ユーザーおよびユーザーグループを追加しないでください。**ロール** タブをクリックします。
4. **管理** チェックボックスを選択します。
5. **外部グループ** タブをクリックします。
6. **外部ユーザーグループの追加** をクリックします。
7. **名前** フィールドに、Active Directory の名前を入力します。
8. リストから **外部** を選択します。

14.9. TOTP での RED HAT SINGLE SIGN ON 認証の設定

TOTP カードを使用した外部認証用の OpenID プロバイダーとして Red Hat Single Sign-On を使用するように Satellite を設定するには、以下のセクションを使用します。

14.9.1. Red Hat Single Sign On 認証を使用した Satellite の設定時の前提条件

Red Hat Single Sign-On 外部認証を使用して Satellite を設定する前に、以下の要件を満たすようにしてください。

- HTTP ではなく、HTTPS を使用する Red Hat Single Sign On サーバーを正常にインストールしている。

- 管理者権限を持つ Red Hat Single Sign-On アカウント。
- Red Hat Single Sign-On で作成した Satellite ユーザーアカウントのレلمム。
- 証明書または CA が自己署名されている場合は、それらがエンドユーザー証明書トラストストアに追加されていることを確認する。
- ユーザーが Red Hat Single Sign-On にインポートまたは追加されている。LDAP や Kerberos などの既存のユーザーデータベースが設定されている場合は、ユーザーのフェデレーションを設定することでユーザーをインポートできます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の [ユーザーストレージフェデレーション](#) を参照してください。

既存のユーザーデータベースが設定されていない場合は、Red Hat Single Sign-On でユーザーを手作業で作成できます。詳細は、**Red Hat Single Sign On サーバー管理ガイド**の [新規ユーザーの作成](#) を参照してください。

14.9.2. Satellite の Red Hat Single Sign-On クライアントとして登録

以下の手順を使用して、Satellite をクライアントとして Red Hat Single Sign-On に登録し、認証ソースとして Red Hat Single Sign-On を使用するように Satellite を設定します。

Satellite と Red Hat Single Sign-On は、2つの異なる認証方法で設定できます。

1. Satellite Web UI を使用した Satellite への認証。
2. Satellite CLI を使用した Satellite への認証。

どちらの方法でも、異なる Satellite クライアントを Red Hat Single Sign-On に登録して設定する必要があるため、ユーザーの認証方法を事前に決定する必要があります。この手順では、Red Hat Single Sign-On の Satellite クライアントの登録および設定方法が区別されています。

両認証方法を使用して、どちらのクライアントも適宜設定する場合には、Red Hat Single Sign-On に異なる Satellite クライアントを2つ登録することも可能です。

手順

1. Satellite Server で、以下のパッケージをインストールします。

```
# satellite-maintain packages install mod_auth_openidc keycloak-httpd-client-install
```

2. Satellite をクライアントとして Red Hat Single Sign-On に登録します。Web UI と CLI とでログインの登録プロセスが異なる点に注意してください。Red Hat Single Sign-On に2つの Satellite クライアントを登録すると、Web UI と CLI から Satellite にログインできます。

- Web UI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name foreman-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

次に、認証ソースとして Red Hat Single Sign On を使用するように Satellite を設定します。

```
# satellite-installer --foreman-keycloak true \
--foreman-keycloak-app-name "foreman-openidc" \
--foreman-keycloak-realm "Satellite_Realm"
```

- CLI で Satellite への認証を行う場合は、以下のようにクライアントを作成します。

```
# keycloak-httpd-client-install --app-name hammer-openidc \
--keycloak-server-url "https://RHSSO.example.com" \
--keycloak-admin-username "admin" \
--keycloak-realm "Satellite_Realm" \
--keycloak-admin-realm master \
--keycloak-auth-role root-admin \
-t openidc -l /users/extlogin --force
```

プロンプトが表示されたら、管理アカウントのパスワードを入力します。このコマンドは、Red Hat Single Sign On で Satellite のクライアントを作成します。

3. **httpd** サービスを再起動します。

```
# systemctl restart httpd
```

14.9.3. Red Hat Single Sign-On での Satellite クライアントの設定

以下の手順を使用して、Red Hat Single Sign-On Web UI で Satellite クライアントを設定し、Satellite クライアントのグループおよびオーディエンスマッパーを作成します。

手順

1. Red Hat Single Sign-On Web UI で、**クライアント** に移動し、Satellite クライアントをクリックします。
2. アクセスタイプを設定します。
 - Satellite web UI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **機密** を選択します。
 - CLI を使用して Satellite への認証を行うには、**アクセスタイプ** リストから **公開** を選択します。
3. **有効なリダイレクト URI** フィールドに有効なリダイレクト URI を追加します。
 - Satellite web UI を使用して Satellite への認証を行うには、**https://satellite.example.com/users/extlogin** の形式で URI を入力します。Satellite FQDN の後に **/users/extlogin** の文字列を追加する必要があります。この手順の完了後に、Satellite クライアントが Satellite web UI を使用してログインするには以下の **有効なリダイレクト URI** が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri
https://satellite.example.com/users/extlogin
```

- CLI を使用してユーザーが Satellite への認証を行うには、既存の URI の下の空白フィールドに `urn:ietf:wg:oauth:2.0:oob` を入力します。
この手順の完了後に、Satellite クライアントが CLI を使用してログインするには以下の有効なリダイレクト URI が必要です。

```
https://satellite.example.com/users/extlogin/redirect_uri  
urn:ietf:wg:oauth:2.0:oob
```

4. **Save** をクリックします。
5. **マッパー** タブ、**作成** の順にクリックし、オーディエンスマッパーを追加します。
6. **名前** フィールドに、オーディエンスマッパーの名前を入力します。
7. **マッパータイプ** リストから、**オーディエンス** を選択します。
8. **組み込み済みクライアントオーディエンス** リストから、Satellite クライアントを選択します。
9. **Save** をクリックします。
10. **作成** をクリックして、グループメンバーシップをもとに Satellite の認証を指定できるようにグループマッパーを追加します。
11. **名前** フィールドにグループマッパーの名前を入力します。
12. **マッパータイプ** リストから、**グループメンバーシップ** を選択します。
13. **トークンクレーム名** に `groups` と入力します。
14. **フルグループパス** のトグルを OFF に設定します。
15. **Save** をクリックします。

14.9.4. Red Hat Single Sign-On 認証用の Satellite オプションの設定

このセクションでは、Satellite Web UI または CLI を使用して Red Hat Single Sign-On 認証用に Satellite を設定します。

14.9.4.1. Web UI を使用した Red Hat Single Sign-On 認証用の Satellite オプションの設定

以下の手順に従って、Satellite Web UI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レルム内の `https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration` の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **機密** に設定されていることを確認します。

手順

1. Satellite Web UI で、**管理** > **設定** に移動して、**認証** タブをクリックします。
2. **ログイン委任の認証** の行を見つけ、**値** コラムで **Yes** に値を設定します。

3. **Authorize login delegation auth source user autocreate**行を見つけ、**値** コラムで **External** に値を設定します。
4. **Login delegation logout URL** の行を見つけ、**Value** 列で、値を `https://satellite.example.com/users/extlogout` に設定します。
5. **OIDC アルゴリズム** の行を見つけ、**値** コラムで、Red Hat Single Sign On のエンコーディングのアルゴリズムを設定します (例: **RS256**)。
6. **OIDC オーディエンス** 行を見つけ、**値** コラムで、値を Red Hat Single Sign On のクライアント ID に設定します。
7. **OIDC Issuer** 行を見つけ、**Value** 列で、値を `https://RHSSO.example.com/auth/realms/Satellite_Realm` に設定します。
8. **OIDC JWKs URL** 行を見つけ、**Value** 列で、値を `https://RHSSO.example.com/auth/realms/Satellite_Realm/protocol/openid-connect/certs` に設定します。
9. Satellite Web UI で、**Administer > Authentication Sources** に移動し、**External** カードの垂直省略記号をクリックして、**Edit** を選択します。
10. **場所** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる場所を追加します。
11. **組織** タブをクリックして、Red Hat Single Sign-On 認証ソースを使用できる組織を追加します。
12. **Submit** をクリックします。

14.9.4.2. CLI を使用した Red Hat Single Sign-On 認証用の Satellite オプションの設定

以下の手順に従って、Satellite CLI を使用して Red Hat Single Sign-On 認証向けに Satellite を設定します。

レーム内の `https://RHSSO.example.com/auth/realms/Satellite_Realm/.well-known/openid-configuration` の URL に移動し、値を取得して Satellite を設定できる点に留意してください。

前提条件

- Red Hat Single Sign-On Web UI の Satellite クライアントでの **アクセスタイプ** 設定が **公開** に設定されていることを確認します。

手順

1. Satellite で、ログイン委任を **true** に設定し、ユーザーが Open IDc プロトコルを使用して認証できるようにします。

```
# hammer settings set --name authorize_login_delegation --value true
```

2. ログイン委任のログアウト URL を以下のように設定します。

```
# hammer settings set --name login_delegation_logout_url \
--value https://satellite.example.com/users/extlogout
```

- Red Hat Single Sign On のエンコーディングのアルゴリズムを設定します (例: **RS256**)。

```
# hammer settings set --name oidc_algorithm --value 'RS256'
```

- URL **RHSSO.example.com/auth/realms/RHSSO_REALM/.well-known/openid-configuration** を開いて値をメモし、以下のステップのオプションに入力します。
- Open IDC オーディエンスに Hammer クライアントの値を追加します。

```
# hammer settings set --name oidc_audience \
--value "[satellite.example.com-hammer-openidc]"
```



注記

複数の Red Hat Single Sign-On クライアントを Satellite に登録する場合は、以下のように、アレイに全オーディエンスを必ず追加してください。以下に例を示します。

```
# hammer settings set --name oidc_audience \
--value "[satellite.example.com-foreman-openidc', 'satellite.example.com-hammer-openidc']"
```

- Open IDC 発行者の値を設定します。

```
# hammer settings set --name oidc_issuer \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm"
```

- Open IDC Java Web Token (JWT) の値を設定します。

```
# hammer settings set --name oidc_jwks_url \
--value "RHSSO.example.com/auth/realms/RHSSO_Realm/protocol/openid-connect/certs"
```

- Red Hat Single Sign-On 認証ソースの ID を取得します。

```
# hammer auth-source external list
```

- ロケーションと組織を設定します。

```
# hammer auth-source external update --id Authentication Source ID \
--location-ids Location ID --organization-ids Organization ID
```

14.9.5. TOTP での Red Hat Single Sign On 認証を使用した Satellite の設定

Time-based One-time Password (TOTP) を使用した外部認証用の OpenID プロバイダーとして Red Hat Single Sign-On を使用するように Satellite を設定するには、以下のセクションを使用します。

手順

- Red Hat Single Sign-On Web UI で、Satellite レルムに移動します。
- Authentication** に移動し、**OTP Policy** タブをクリックします。

3. **サポートされるアプリケーション** フィールドに FreeOTP または Google Authenticator が含まれていることを確認します。
4. 要件に合わせて OTP を設定します。
5. 必要に応じて、すべてのユーザーのデフォルト認証方法として TOTP 認証を使用する場合は、**Flows** タブをクリックして **OTP Form** 設定の右側にある **REQUIRED** を選択します。
6. **Required Actions** タブをクリックします。
7. **Configure OTP** 行の右側にある **Default Action** チェックボックスを選択します。

14.9.6. Red Hat Single Sign-On TOTP 認証を使用した Satellite web UI へのログイン

以下の手順に従って、Red Hat Single Sign-On TOTP 認証で Satellite Web UI にログインします。

手順

1. Satellite にログインすると、Satellite は Red Hat Single Sign-On のログイン画面にリダイレクトします。
2. ユーザー名とパスワードを入力し、**ログイン** をクリックします。
3. 初回ログインの場合には、Red Hat Single Sign-On により、バーコードをスキャンし、表示された暗証番号を入力してクライアントを設定するように求められます。
4. クライアントを設定して有効な暗証番号を入力すると、Red Hat Single Sign-On で Satellite にリダイレクト後にログインされます。

14.9.7. Red Hat Single Sign-On を使用した Satellite CLI へのログイン

以下の手順に従って、コード付与タイプを使用して Satellite CLI への認証を行います。

手順

1. コード付与タイプを使用して Satellite CLI への認証を行うには、以下のコマンドを入力します。

```
# hammer auth login oauth \  
--two-factor \  
--oidc-token-endpoint 'https://RHSSO.example.com/auth/realms/ssl-realm/protocol/openid-  
connect/token' \  
--oidc-authorization-endpoint 'https://RHSSO.example.com/auth' \  
--oidc-client-id 'satellite.example.com-foreman-openidc' \  
--oidc-redirect-uri urn:ietf:wg:oauth:2.0:oob
```

このコマンドは、サクセスコードの入力を要求します。

2. サクセスコードを取得するには、コマンドが返す URL に移動し、必要な情報を提供します。
3. Web UI が返すサクセスコードをコピーします。
4. **hammer auth login oauth** のコマンドプロンプトに、サクセスコードを入力して Satellite CLI に対して認証を行います。

14.9.8. Red Hat シングルサインオン認証用のグループマッピングの設定

必要に応じて、ロールベースのアクセス制御 (RBAC) を実装するには、Satellite でグループを作成し、このグループにロールを割り当ててから Active Directory グループを Satellite グループにマッピングします。これにより、Red Hat Single Sign-On の指定のグループに所属する場合には、該当する Satellite グループでログインします。この例では、Active Directory の Satellite-admin ユーザーグループのユーザーを設定し、Satellite で管理者権限を持つユーザーとして認証されるようにします。

手順

1. Satellite Web UI で、**管理 > ユーザーグループ** に移動して、**ユーザーグループの作成** ボタンをクリックします。
2. **名前** フィールドにユーザーグループの名前を入力します。名前は Active Directory と同じにしないでください。
3. 右側の列には、ユーザーおよびユーザーグループを追加しないでください。**ロール** タブをクリックします。
4. **管理** チェックボックスを選択します。
5. **外部グループ** タブをクリックします。
6. **外部ユーザーグループの追加** をクリックします。
7. **名前** フィールドに、Active Directory の名前を入力します。
8. リストから **外部** を選択します。

14.10. RED HAT SINGLE SIGN ON 認証の無効化

Satellite で Red Hat Single Sign On 認証を無効化するには、以下の手順を実行します。

手順

- Red Hat Single Sign On 認証を無効化するには、以下のコマンドを入力します。

```
# satellite-installer --reset-foreman-keycloak
```

第15章 リソースの監視

本章では、管理システムの監視とレポートの設定方法について説明します。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。

15.1. 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15.1 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ー修正などが利用できるようになります。以下の表はサブスクリプションの状態の種類を示しています。

表15.2 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15.3 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。

15.1.1. 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` を入力し、**検索** をクリックします。検索では、説明を含む 4 つの設定が返されます。
3. **値** のコラムで、数字の横にあるアイコンをクリックして編集します。
4. 希望する秒数を入力したら、**保存** をクリックします。



注記

低帯域幅の場合は `%_finish_timeout` 値の編集が役に立つ場合があります。待ち時間が長い場合は `%_accept_timeout` 値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` を入力し、**検索** をクリックします。
3. **値** コラム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。

15.2. 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、**値** コラム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、**値** コラム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。

15.3. SATELLITE SERVER の監視

Satellite web UI の **概要** ページで、以下の概要情報を確認できます。

- システムステータス (Capsule、利用可能なプロバイダー、コンピュートリソース、およびプラグインを含む)
- サポート情報
- System Information
- バックエンドシステムの状態
- インストール済みパッケージ

概要 ページに移動するには:

- Satellite web UI の右上隅で **管理 > 概要** をクリックします。



注記

Pulp の失敗後は、同期の遅延のため、最大 10 分間 Pulp のステータスが **エラー** ではなく、**OK** と表示される場合があります。

15.4. 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。

15.4.1. 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、[Installing Capsule Server](#) 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。

15.4.2. 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。

15.4.3. 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た設定の管理](#) の [Satellite と Puppet のインテグレーションの有効化](#) を参照してください。

第16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。

16.1. 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。

16.2. 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: 次のコマンドを使用して CLI プラグインをインストールできます。

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

16.3. 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。

詳細については、**Managing Hosts** の [Template Writing Reference](#) 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` にアクセスしてください。

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。

16.4. 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 を設定する場合は、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の Satellite へのアクセスを確保するように SELinux を設定する手順](#) を参照してください。

16.5. 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
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

イベント名	説明	ペイロード
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
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)。

イベント名	説明	ペイロード
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
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

16.6. 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。

16.7. 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
```

16.8. 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 %>"
}
```

16.9. 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
```

16.10. 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17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。

17.1. 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、入力を継続するかのいずれかのオプションを選択できます。検索エンジンがフリーテキストを解釈する方法については、「[フリーテキスト検索の使用](#)」を参照してください。

17.1.1. 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。

17.1.2. クエリー Operator

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

表17.1 検索で使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) を受け入れません。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17.2 検索で使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

17.1.3. 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。

17.2. 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。

17.3. 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。

17.3.1. 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。

17.3.2. 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、**Administer > Settings** に移動して、Satellite web UI で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	いいえ	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	いいえ	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。
ホストの電源ステータスを表示する	あり	ホストインデックスページに電源ステータスを表示します。この機能は、ホストリストページのパフォーマンスの低下につながる可能性のあるコンピュータリソースプロバイダーを呼び出します。
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せん。

設定	デフォルト値	説明
Experimental Labs を表示する	いいえ	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)。
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	あり	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。

設定	デフォルト値	説明
dynflow コンソールを有効にする	あり	デバッグ用に dynflow コンソール (<code>/foreman_tasks/dynflow</code>) を有効にします。
dynflow コンソールに認証が必要	あり	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	あり	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	いいえ	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	デフォルトのメタデータエクスポートモード。 可能なオプション: refresh により、メタデータが再レンダリングされます。 keep は既存のメタデータを保持します。 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。
否定	いいえ	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>/tmp/dir</p> <p>git://example.com</p> <p>https://example.com</p> <p>ssh://example.com</p> <p>/tmp 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	いいえ	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	いいえ	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せん。
すべてのファクトをクリーンアップする	いいえ	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	いいえ	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	あり	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	いいえ	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使用される PXEGrub2 テンプレート。
DNS を強制する	あり	検出されたホストをプロビジョニングするとき、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種類	ファクト + 接頭辞	検出ホスト名の命名パターン。
Prefer IPv6	いいえ	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	あり	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	あり	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	いいえ	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	いいえ	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	いいえ	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	あり	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	いいえ	これが有効になっている場合、複合コンテンツビューに含まれるコンポーネント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	あり	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	いいえ	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	いいえ	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リプションされたホスト	デフォルトのロケーション	登録後に新しいサブスクリプション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	いいえ	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	あり	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	いいえ	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	いいえ	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	あり	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	あり	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なす時間 (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	いいえ	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	あり	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	いいえ	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	あり	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	あり	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される 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	あり	VNC/SPICE WebSocket プロキシコンソールアクセス暗号化 (<code>websockets_ssl_key/cert</code> 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	外部	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	いいえ	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	いいえ	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	あり	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 https://keycloak.example.com/auth/realms/<realm name>/protocol/openid-connect/certs 。
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 /.well-known/openid-configuration 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	いいえ	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	あり	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	<code>/usr/sbin/sendmail</code>	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	あり	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	あり	<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	いいえ	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	いいえ	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	いいえ	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	いいえ	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	あり	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	いいえ	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 はコンピューティングリソースに残り、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、 <code>satellite::dropped_subtree_facts</code> 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	あり	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	いいえ	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	いいえ	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	あり	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	あり	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	いいえ	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	いいえ	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	なし	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	いいえ	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	あり	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	あり	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	あり	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	いいえ	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	いいえ	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ワークがない場合、またはサブネットワーク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	あり	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	あり	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	いいえ	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	いいえ	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 鍵のパスワード	*****	SSH に使用するデフォルトのキーパスワード。 remote_execution_ssh_key_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。
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	あり	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。

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。
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。

設定	デフォルト値	説明
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	無効	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	いいえ	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。