



Red Hat Satellite 6.10

Satellite で Ansible を使用するための設定

Satellite で Ansible を設定し、反復タスクの自動化に役立てます。

Red Hat Satellite 6.10 Satellite で Ansible を使用するための設定

Satellite で Ansible を設定し、反復タスクの自動化に役立てます。

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2021 | You need to change the HOLDER entity in the en-US/Configuring_Satellite_to_use_Ansible.ent file |.

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、Ansible を使用してリモート実行を実行し、反復タスクを自動化するように Red Hat Satellite をセットアップおよび設定する方法を説明します。

目次

第1章 SATELLITE での ANSIBLE を使い始める	3
1.1. ANSIBLE ロールを実行するためのデプロイメントの設定	3
1.2. ANSIBLE ロールのインポート	3
1.3. ANSIBLE 変数のインポート	3
1.4. ANSIBLE 変数の作成	4
1.5. SATELLITE での ANSIBLE 変数のオーバーライド	4
1.6. RED HAT ENTERPRISE LINUX システムロールの追加	6
第2章 ANSIBLE ロールを使用した SATELLITE ホストの反復タスクの自動化	7
2.1. 既存ホストへの ANSIBLE ロールの割り当て	7
2.2. ホストでの ANSIBLE ロールの実行	7
2.3. ホストグループへの ANSIBLE ロールの割り当て	8
2.4. ホストグループでの ANSIBLE ロールの実行	8
第3章 リモートジョブの設定およびセットアップ	9
3.1. ホストでのジョブの実行について	9
3.2. リモート実行のワークフロー	9
3.3. リモート実行用のパーミッション	10
3.4. ジョブテンプレートを作成します。	11
3.5. SATELLITE での任意の CAPSULE へのフォールバックリモート実行の設定	12
3.6. SATELLITE でのグローバル CAPSULE リモート実行の設定	13
3.7. 代替ディレクトリーを使用してホストでリモートジョブを実行するための SATELLITE の設定	13
3.8. リモート実行のための SSH 鍵の配布	14
3.9. リモート実行用の SSH 鍵の手動での配布	14
3.10. SATELLITE API を使用したリモート実行用の SSH 鍵の取得	14
3.11. プロビジョニング中に SSH 鍵を配布するキックスタートテンプレートの設定	15
3.12. KERBEROS チケットを付与するための KEYTAB の設定	15
3.13. リモート実行用の KERBEROS 認証の設定	16
3.14. ジョブテンプレートのセットアップ	16
3.15. リモートジョブの実行	17
3.16. ジョブの監視	19
第4章 RED HAT SATELLITE と ANSIBLE TOWER の統合	20
4.1. SATELLITE SERVER を動的インベントリー項目として ANSIBLE TOWER に追加	20
4.2. ホストへのプロビジョニングコールバックの設定	21
付録A ジョブテンプレートの例および拡張	24
A.1. ジョブテンプレートのカスタマイズ	24
A.2. デフォルトのジョブテンプレートカテゴリ	24
A.3. RESTORECON テンプレートの例	25
A.4. RESTORECON テンプレートのレンダリング	25
A.5. 複数のホストでの RESTORECON テンプレートの実行	26
A.6. テンプレートにパワー操作を組み込む	26

第1章 SATELLITE での ANSIBLE を使い始める

本ガイドでは、Ansible を使用するように Satellite を設定し、リモート実行に Ansible を使用する方法を説明します。

1.1. ANSIBLE ロールを実行するためのデプロイメントの設定

Satellite では、Ansible ロールをインポートして、ルーティンタスクの自動化に役立てることができます。デフォルトでは Ansible は Satellite でのみ有効になっています。

この手順を使用して、Ansible ロールを実行するように Satellite デプロイメントを設定します。

手順

1. ロールの使用元にする Satellite と全 Capsule の `/etc/ansible/roles` ディレクトリーにロールを追加します。カスタムまたはサードパーティーの Ansible ロールを使用する場合には、外部のバージョン管理システムが Satellite と Capsule の間でロールを同期するように設定してください。
2. ホストで Ansible ロールの実行に使用する全 Capsule で、Ansible プラグインを有効にします。

```
# satellite-installer --scenario capsule \  
--enable-foreman-proxy-plugin-ansible
```

3. SSH 鍵を配布して、Capsule が SSH でホストに接続できるようにします。詳細は、『[ホストの管理](#)』ガイドの「[リモート実行のための SSH 鍵の配布](#)」を参照してください。Satellite は、リモート実行ジョブの実行と同様に、Ansible ロールを実行します。
4. Ansible ロールを Satellite にインポートします。
5. 『[Satellite で Ansible を使用するための設定](#)』の「[Ansible ロールを使用した Satellite ホスト上での反復タスクの自動化](#)」に進みます。

1.2. ANSIBLE ロールのインポート

Satellite または Ansible を有効にした Capsule の `/etc/ansible/roles` ディレクトリーから Ansible ロールをインポートできます。インポートするロールが、ロールを使用するすべての Capsule の `/etc/ansible/roles` ディレクトリーに配置されているようにしてください。

手順

1. Satellite Web UI で、[設定](#) > [ロール](#) に移動して、インポートするロールが含まれている Capsule をクリックします。
2. Ansible ロール一覧からインポートするロールのチェックボックスを選択し、[更新](#) をクリックします。

1.3. ANSIBLE 変数のインポート

Ansible ロールは、変数を使用して、具体的な要件のあるシステムの設定を微調整します。たとえば、システムの IP アドレスを特定し、別のシステムの設定値として使用する必要がある場合があります。

Ansible Playbook で Ansible 変数を使用する場合は、Capsule から Ansible 変数をインポートする必要があります。

手順

1. Satellite Web UI で、**設定 > 変数** に移動します。
2. ウィンドウの右上で、インポートする Ansible 変数を含む Capsule を選択します。
3. インポートする Ansible 変数を選択して、**更新** をクリックします。

1.4. ANSIBLE 変数の作成

Ansible ロールは、変数を使用して、具体的な要件のあるシステムの設定を微調整します。

通常、使用する Ansible ロール用に Ansible 変数をインポートします。システム設定をさらに調整する必要がある場合には、Satellite で Ansible 変数を作成することも可能です。

手順

1. Satellite Web UI で、**設定 > 変数** に移動します。
2. 右上隅で **新規 Ansible 変数** をクリックします。
3. キー フィールドで、変数の名前を入力します。この名前が Ansible ロール名を参照していることを確認します。
4. **説明** フィールドには、変数の説明を入力します。
5. **Ansible ロール** リストから、Ansible ロールを選択して変数を関連付けます。
6. オプション：Satellite を使用して Ansible 変数を上書きするには、「[Satellite での Ansible 変数の上書き](#)」を参照してください。
7. Ansible 変数を保存するには、**送信** をクリックします。

1.5. SATELLITE での ANSIBLE 変数のオーバーライド

Satellite で Ansible ロールを実行する場合は、Satellite を使用して、Ansible 変数よりも Ansible ロールを優先させます。

変数のオーバーライドの優先順位

Ansible ロールを使用して、**Effective User** 以外のユーザーとしてタスクを実行する場合、Ansible 変数をオーバーライドするための厳密な優先順位があります。オーバーライドする変数が正しい優先順位に従っていることを確認するには、『[Ansible ユーザーガイド](#)』の「[Variable precedence: Where should I put a variable?](#)」を参照してください。

前提条件

- Satellite に Ansible 変数がある。
 - Ansible 変数をインポートするには、「[Ansible 変数のインポート](#)」を参照してください。
 - Ansible 変数を作成するには、「[Ansible 変数の作成](#)」を参照してください。

以下の手順は、ホストとホストグループを参照します。ホストとホストグループの詳細は、『[ホストの管理](#)』ガイドを参照してください。

手順

1. Satellite Web UI で、**設定 > 変数** に移動します。
2. Satellite でオーバーライドし、管理する Ansible 変数を選択します。
3. **デフォルトの動作** エリアに移動して、**上書き** チェックボックスを選択します。
4. **パラメータータイプ** から、検証の値タイプを選択します。(例: **文字列** や **ブール値** の変数など)。
5. **デフォルト値** フィールドで、変数と一致するものがない場合に使用するデフォルト値を入力します。
6. オプション: Ansible 変数をプレーンテキストで表示したくない場合は、**値の非表示** のチェックボックスを選択し、Satellite Web UI で変数のコンテンツをアスタリスクとして表示します。
7. オーバーライド設定を保存するには、**送信** をクリックします。

Ansible 変数を使用するには、ホストまたはホストグループにパラメーターとして変数を追加するか、グローバルパラメーターとして変数を追加します。

ホストグループの場合:

1. Web UI で、**設定 > ホストグループ** に移動し、使用するホストグループ名を選択します。
2. **パラメーター** タブをクリックして、**ホストのグループパラメーター** のエリアで、**パラメーターの追加** をクリックします。
3. **名前** フィールドに Ansible 変数名を追加します。
4. **タイプ** リストから検証する変数タイプを選択します。
5. **値** フィールドに変数の値を入力します。

ホストの場合:

1. Satellite Web UI で **ホスト > すべてのホスト** に移動して、使用するホストで、**編集** ボタンをクリックします。
2. **パラメーター** タブをクリックして、**ホストのパラメーター** のエリアで、**パラメーターの追加** をクリックします。
3. **名前** フィールドに Ansible 変数名を追加します。
4. **タイプ** リストから検証する変数タイプを選択します。
5. **値** フィールドに変数の値を入力します。

グローバルパラメータとして追加する手順:

1. Satellite Web UI で **設定 > グローバルパラメーター** に移動して、**パラメーターの作成** をクリックします。
2. **名前** フィールドに Ansible 変数名を追加します。
3. **タイプ** リストから検証する変数タイプを選択します。

4. **値** フィールドに変数の値を入力します。
5. オプション: Ansible 変数をプレーンテキストで表示したくない場合は、**値の非表示** のチェックボックスを選択し、Satellite Web UI で変数のコンテンツをアスタリスクとして表示します。

1.6. RED HAT ENTERPRISE LINUX システムロールの追加

Red Hat Enterprise Linux システムロールは、Red Hat Enterprise Linux サーバーをリモートで管理するための設定インターフェースです。Red Hat Enterprise Linux システムロールを利用すると、Satellite に Ansible ロールを追加できます。Satellite で Ansible ロールを使用すると、設定がより速くかつ容易に実行できます。

一部の Red Hat Enterprise Linux システムロールのサポートレベルは、テクノロジープレビューにある場合があります。Red Hat Enterprise Linux システムロールに関するサポートレベルの最新情報および一般情報は、「[Red Hat Enterprise Linux システムロール](#)」を参照してください。

Extras チャンネルに登録する前に、「[Red Hat Enterprise Linux Extras の製品ライフサイクル](#)」の記事を参照してください。

手順

1. **rhel-7-server-extras-rpms** リポジトリが有効になっていることを確認します。

```
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

2. **rhel-system-roles** パッケージをインストールします。

```
# satellite-maintain packages install rhel-system-roles
```

rhel-system-roles パッケージは `/usr/share/ansible/roles/` にダウンロードされます。インポート前にファイルを表示して修正を加えることができます。

3. Satellite Web UI で、**設定 > ロール** に移動して、インポートするロールが含まれている Capsule をクリックします。
4. Ansible ロール一覧からインポートするロールのチェックボックスを選択し、**更新** をクリックします。

これで Ansible ロールをホストまたはホストグループに割り当てることができます。詳細は、『[Satellite で Ansible を使用するための設定](#)』の「[既存ホストへの Ansible ロールの割り当て](#)」を参照してください。

また、ロール内に含まれているモジュールを Ansible のジョブテンプレートに追加すると、それらを Ansible playbooks にも追加することもできます。ジョブテンプレートには **hosts:all** 行を含める必要があります。詳細は、「[Red Hat Enterprise Linux \(RHEL\) System Roles](#)」を参照してください。

第2章 ANSIBLE ロールを使用した SATELLITE ホストの反復タスクの自動化

2.1. 既存ホストへの ANSIBLE ロールの割り当て

Red Hat Enterprise Linux バージョン 8、7、6.9 以降では、リモート管理に Ansible ロールを使用できません。

前提条件

- Ansible ロールを設定し、インポートしている。

手順

1. Satellite Web UI で、**ホスト > すべてのホスト** に移動します。
2. Ansible ロールを割り当てるホストで、**編集** をクリックします。
3. **Ansible ロール** タブを選択して、**すべての項目** リストで、追加するロールを検索します。
4. 追加するロールを選択して、矢印アイコンをクリックし、**選択された項目** リストにロールを移動させます。
5. **送信** をクリックします。

ホストに Ansible ロールを割り当てると、リモート実行用に Ansible を使用できるようになります。詳細は、「[リモート実行のための SSH 鍵の配布](#)」を参照してください。

パラメーター変数の上書き

パラメーター タブで **パラメーターの追加** をクリックして、ランタイム時にジョブテンプレートにわたすパラメーター変数を追加します。これには、ホストに関連付ける全 Ansible Playbook パラメーター、ホストパラメーターが含まれます。Ansible のジョブテンプレートでパラメーター変数を使用するには、**ホストパラメーター** を追加する必要があります。

2.2. ホストでの ANSIBLE ロールの実行

Satellite Web UI を使用して、ホストで Ansible ロールを実行できます。

前提条件

- デプロイメントで Ansible ロールを実行するように設定しておく。詳細は、『[Satellite で Ansible を使用するための設定](#)』の「[Ansible ロールを実行するためのデプロイメントの設定](#)」を参照してください。
- Ansible ロールをホストに割り当てておく。

手順

1. Satellite Web UI で、**ホスト > すべてのホスト** に移動します。
2. 実行する Ansible ロールが含まれるホストのチェックボックスを選択します。
3. **アクションの選択** リストから **Ansible ロールのプレイ** を選択します。

Ansible ロールの実行 ページで、Ansible ジョブのステータスを表示できます。ジョブを返すには、**再実行** ボタンをクリックしてください。

2.3. ホストグループへの ANSIBLE ロールの割り当て

Red Hat Enterprise Linux バージョン 8、7、6.9 以降では、リモート管理に Ansible ロールを使用できません。

前提条件

- デプロイメントで Ansible ロールを実行するように設定しておく。詳細は、『**Satellite で Ansible を使用するための設定**』の「**Ansible ロールを実行するためのデプロイメントの設定**」を参照してください。

手順

1. Satellite Web UI で、**設定 > ホストグループ** に移動します。
2. ホストグループの一覧から、Ansible ロールを追加するホストグループ名をクリックします。
3. **Ansible ロール** タブを選択して、**すべての項目** リストで、追加するロールを検索します。
4. 追加するロールを選択して、矢印アイコンをクリックし、**選択された項目** リストにロールを移動させます。
5. **送信** をクリックします。

2.4. ホストグループでの ANSIBLE ロールの実行

Satellite Web UI を使用して、ホストグループで Ansible ロールを実行できます。

前提条件

- デプロイメントで Ansible ロールを実行するように設定しておく。詳細は、『**Satellite で Ansible を使用するための設定**』の「**Ansible ロールを実行するためのデプロイメントの設定**」を参照してください。
- Ansible ロールをホストグループに割り当てておく。
- ホストグループに最低でもホストを1台以上設定する必要がある。

手順

1. Satellite Web UI で、**設定 > ホストグループ** に移動します。
2. ホストグループの **アクション** コラムの一覧から、**すべての Ansible ロールの実行** を選択します。

Ansible ロールの実行 ページで、Ansible ジョブのステータスを表示できます。ジョブを返すには、**再実行** ボタンをクリックしてください。

第3章 リモートジョブの設定およびセットアップ

このセクションでは、リモートホストでジョブを実行するように Satellite を設定する方法を説明します。

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

3.1. ホストでのジョブの実行について

シェルスクリプトまたは Ansible タスクと Playbook を使用して、Capsules からリモートでホスト上でジョブを実行できます。これはリモート実行と呼ばれます。

作成したカスタムの Ansible ロール、またはダウンロードしたロールの場合、ロールを含むパッケージを Capsule のベースオペレーティングシステムにインストールする必要があります。Ansible ロールを使用する前に、ロールがインストールされている Capsule から Satellite にロールをインポートする必要があります。

通信は Capsule Server 経由で行われるので、Satellite Server にはターゲットホストへの直接のアクセスが不要で、スケーリングして多数のホストを管理できます。リモート実行は、SSH サービスを使用します。SSH サービスは、ターゲットホストで有効化され、実行されている必要があります。リモート実行 Capsule が、ターゲットホストのポート 22 にアクセスできることを確認してください。

Satellite は、Embedded Ruby (ERB) 構文ジョブテンプレートを使用します。詳細は、『[ホストの管理](#)』ガイドの「[テンプレート作成の参照](#)」を参照してください。

シェルスクリプトおよび Ansible のジョブテンプレートが複数、デフォルトで含まれています。詳細は、「[ジョブテンプレートのセットアップ](#)」を参照してください。

デフォルトでは、Satellite Server はリモート実行ではなく Katello エージェントを使用するように設定されています。この設定を変更するには、[管理](#) > [設定](#) に移動し、[コンテンツ](#) をクリックして、[Use remote execution by default](#) の設定を変更します。



注記

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

複数のホストでジョブを一度に実行でき、コマンドで変数を使用すると、実行するジョブをより詳細に制御できます。ホストファクトとパラメーターを使用して、変数の値を設定できます。

さらに、コマンドの実行時にテンプレートのカスタム値を指定できます。

詳細は、「[リモートジョブの実行](#)」を参照してください。

3.2. リモート実行のワークフロー

ホストでリモートジョブを実行すると、すべてのホストについて、Satellite は以下のアクションを実行して、使用するリモート実行 Capsule を検出します。

Satellite は、Ansible 機能が有効になっている Capsules のみを検索します。

1. Satellite は、[リモート実行](#) のチェックボックスが選択されているホストのインターフェースを検出します。

2. Satellite はこれらのインターフェースのサブネットを検出します。
3. Satellite は、これらのサブネットに割り当てられたリモート実行 Capsule を検出します。
4. この Capsule のセットから、Satellite は実行中のジョブの数が最も少ない Capsule を選択します。こうすることで、Satellite はリモート実行 Capsule 間でのジョブの負荷を確実に分散させます。
5. この段階で Satellite がリモート実行 Capsule を検出できず、**任意の Capsule へのフォールバック** 設定が有効になっている場合、Satellite は別の Capsule のセットを追加して、そこからリモート実行 Capsule を選択します。Satellite は、ホストに割り当てられている以下のタイプの Capsule の中から最も負荷の少ない Capsule を選択します。
 - ホストのサブネットに割り当てられた DHCP Capsule、DNS Capsule、および TFTP Capsule
 - ホストのドメインに割り当てられた DNS Capsule
 - ホストのレルムに割り当てられた Realm Capsule
 - Puppet Master Capsule
 - Puppet CA Capsule
 - OpenSCAP Capsule
6. この段階で Satellite がリモート実行 Capsule を検出せず、**グローバル Capsule の有効化** 設定が有効になっている場合、Satellite は、ホストの組織およびロケーションにあるすべての Capsule のセットから最も負荷の少ないリモート実行 Capsule を選択し、リモートジョブを実行します。

3.3. リモート実行用のパーミッション

ターゲットにするホストを含め、インフラストラクチャー内で実行するジョブとそれを実行するユーザーを制御できます。リモート実行機能は 2 つの組み込みロールを提供します。

- **Remote Execution Manager (リモート実行マネージャー)**: このロールは、すべてのリモート実行機能および機能性へのアクセスを許可します。
- **Remote Execution User (リモート実行ユーザー)**: このロールは、ジョブの実行のみを許可します。ジョブテンプレートを変更するパーミッションは提供されません。

Remote Execution User (リモート実行ユーザー) ロールのクローンを作成し、そのフィルターをカスタマイズして詳細度を高めることができます。**view_job_templates** パーミッションでフィルターを調整する場合、ユーザーは一致するジョブテンプレートに基づくジョブのみを確認し、トリガーすることが可能です。**view_hosts** パーミッションおよび **view_smart_proxies** パーミッションを使用すると、ロールに表示されるホストまたは Capsule を制限できます。

execute_template_invocation パーミッションは、ジョブの実行が開始する直前に確認される特殊なパーミッションです。このパーミッションは、特定のホストで実行できるジョブテンプレートを定義します。これにより、パーミッションの指定時に詳細度をさらに高めることができます。ロールおよびパーミッションの使用については、『[Administering Red Hat Satellite](#)』の「[Creating and Managing Roles](#)」を参照してください。

以下の例は、**execute_template_invocation** パーミッションのフィルターを示しています。

```
name = Reboot and host.name = staging.example.com
name = Reboot and host.name ~ *.staging.example.com
name = "Restart service" and host_group.name = webservers
```

上記の例の最初の行により、ユーザーは選択した1つのホストで **Reboot** テンプレートを適用できません。2番目の行は、**.staging.example.com** で終わる名前を持つホストのプールを定義します。3番目の行はテンプレートをホストグループにバインドします。



注記

ユーザーに割り当てるパーミッションは時間の経過と共に変更されます。ユーザーに将来実行するスケジュールされたジョブがあり、パーミッションが変更された場合、パーミッションはジョブ実行の直前にチェックされるため、実行が失敗する可能性があります。

3.4. ジョブテンプレートを作成します。

以下の手順を使用してジョブテンプレートを作成します。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. **ホスト > ジョブテンプレート** に移動します。
2. **新規ジョブテンプレート** をクリックします。
3. **テンプレート** タブをクリックして、**名前** フィールドにジョブテンプレートの一意名を入力します。
4. **デフォルト** を選択して、テンプレートをすべての組織およびロケーションで利用できるようにします。
5. テンプレートエディターで直接テンプレートを作成するか、**インポート** をクリックしてテキストファイルからテンプレートをアップロードします。
6. オプション: **監査コメント** フィールドで、変更に関する情報を追加します。
7. **ジョブ** タブをクリックして、**ジョブカテゴリ** フィールドに、独自のカテゴリを入力するか、「**デフォルトのジョブテンプレートカテゴリ**」に記載のデフォルトカテゴリから選択します。
8. オプション: **説明形式** フィールドで説明テンプレートを入力します。(例: **Install package % {package_name}**)。また、テンプレートでは **%{template_name}** および **%{job_category}** も使用できます。
9. **プロバイダタイプ** リストから、Shell スクリプトに **SSH** を、Ansible タスクまたは Playbook に **Ansible** を選択します。
10. オプション: **Timeout to kill** フィールドで、ジョブが完了しない場合に、ジョブを中断するタイムアウトの値を入力します。
11. オプション: **入力を追加** をクリックし、入力パラメーターを定義します。ジョブの実行時にパラメーターを要求し、テンプレートに定義する必要はありません。各種サンプルについては、**ヘルプ** タブを参照してください。

12. オプション: **外部入力セット** をクリックして、このジョブの他のテンプレートを追加します。
13. オプション: **実効ユーザー** エリアで、コマンドでデフォルトの `remote_execution_effective_user` 設定を使用できない場合に、ユーザーを設定します。
14. オプション: このテンプレートをスニペットとして他のテンプレートに追加する場合は、**タイプ** タブをクリックして、**スニペット** を選択します。
15. **ロケーション** タブをクリックして、テンプレートを使用するロケーションを追加します。
16. **組織** タブをクリックして、テンプレートを使用する組織を追加します。
17. **送信** をクリックして変更を保存します。

テンプレート構文に他のテンプレートを追加して、ジョブテンプレートを拡張およびカスタマイズできます。詳細は、『**ホストの管理**』ガイドの付録を参照してください。

CLI手順

1. テンプレート定義ファイルを使用してジョブテンプレートを作成するには、以下のコマンドを使用します。

```
# hammer job-template create \
--file "path_to_template_file" \
--name "template_name" \
--provider-type SSH \
--job-category "category_name"
```

3.5. SATELLITE での任意の CAPSULE へのフォールバックリモート実行の設定

任意の Capsule へのフォールバック設定を有効にして、ホストに割り当てられている Capsule の一覧からリモート実行 Capsule を検索するように Satellite を設定できます。これは、サブネットが設定されていないホストでリモートジョブを実行する必要がある場合、またはリモート実行機能が有効になっていない Capsule にホストのサブネットが割り当てられている場合に役立ちます。

任意の Capsule へのフォールバック設定が有効になっている場合、Satellite は別の Capsule のセットを追加して、そこからリモート実行 Capsule を選択します。また、以下のように Satellite は、ホストに割り当てられたすべての Capsule のセットから最も負荷の少ない Capsule を選択します。

- ホストのサブネットに割り当てられた DHCP Capsule、DNS Capsule、および TFTP Capsule
- ホストのドメインに割り当てられた DNS Capsule
- ホストのレルムに割り当てられた Realm Capsule
- Puppet Master Capsule
- Puppet CA Capsule
- OpenSCAP Capsule

手順

1. Satellite Web UI で、**管理 > 設定**に移動します。

2. **リモート実行** をクリックします。
3. **任意の Capsule へのフォールバック** を設定します。

CLI手順

Satellite で **hammer settings set** コマンドを入力して、**任意の Capsule へのフォールバック** を設定します。たとえば、値を **true** に設定するには、以下のコマンドを入力します。

```
# hammer settings set --name=remote_execution_fallback_proxy --value=true
```

3.6. SATELLITE でのグローバル CAPSULE リモート実行の設定

デフォルトで Satellite は、Capsule がホストのサブネットに割り当てられているかどうかに関係なく、ホストの組織とロケーションでリモート実行 Capsule を検索します。ホストのサブネットに割り当てられている Capsule に検索を限定する場合は、**グローバル Capsule の有効化** 設定を無効化することができます。

グローバル Capsule の有効化 設定が有効になっている場合、Satellite は別の Capsule のセットを追加して、そこからリモート実行 Capsule を選択します。また、Satellite は、ホストの組織およびロケーションにあるすべての Capsule のセットから最も負荷の少ないリモート実行 Capsule を選択し、リモートジョブを実行します。

手順

1. Satellite Web UI で、**管理 > 設定** に移動します。
2. **リモート実行** をクリックします。
3. **グローバル Capsule の有効化** を設定します。

CLI手順

- Satellite で **hammer settings set** コマンドを入力して、**Enable Global Capsule** を設定します。たとえば、値を **true** に設定するには、以下のコマンドを入力します。

```
# hammer settings set --name=remote_execution_global_proxy --value=true
```

3.7. 代替ディレクトリーを使用してホストでリモートジョブを実行するための SATELLITE の設定

Ansible は、必要な独自のファイルを **\$HOME/.ansible/tmp** ディレクトリーに配置します。**\$HOME** はリモートユーザーのホームディレクトリーに置き換えます。必要な場合は、別のディレクトリーを設定するオプションがあります。

手順

1. 新しいディレクトリーを作成します (例: **new_place**)。

```
# mkdir /remote_working_dir
```

2. デフォルトの **var** ディレクトリーから SELinux コンテキストをコピーします。

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

3. `/etc/foreman-proxy/settings.d/ansible.yml` ファイルの `ansible_working_dir` 設定を編集して、必要なディレクトリーにポイントさせます。以下に例を示します。

```
:ansible_working_dir: /remote_working_dir
```

3.8. リモート実行のための SSH 鍵の配布

リモート実行接続の認証に SSH 鍵を使用するには、Capsule から管理するアタッチ済みのホストに、公開 SSH 鍵を配布する必要があります。ホストで SSH サービスが有効化され、実行していることを確認します。ポート 22 にアクセスできるように、ネットワークまたはホストベースのファイアウォールを設定し、ポート 22 へのアクセスを有効化します。

Capsule からターゲットホストに公開 SSH 鍵を配布するには、以下のいずれか1つの方法を使用します。

1. 「リモート実行用の SSH 鍵の手動での配布」。
2. 「Satellite API を使用したリモート実行用の SSH 鍵の取得」。
3. 「プロビジョニング中に SSH 鍵を配布するキックスタートテンプレートの設定」。
4. 新しい Satellite ホストの場合、グローバル登録テンプレートを使用して、登録中に SSH 鍵を Satellite ホストにデプロイすることができます。詳細は、「[グローバル登録テンプレートを使ったホストの Red Hat Satellite への登録](#)」を参照してください。

Satellite は、リモート実行機能の SSH 鍵を、デフォルトで Satellite からプロビジョニングされたホストに配布します。

ホストが Amazon Web Services で実行されている場合は、パスワード認証を有効にします。詳細は、<https://aws.amazon.com/premiumsupport/knowledge-center/new-user-accounts-linux-instance> を参照してください。

3.9. リモート実行用の SSH 鍵の手動での配布

SSH 鍵を手動で配布するには、以下の手順を実行します。

手順

1. Capsule で以下のコマンドを入力します。管理するターゲットホストごとに繰り返します。

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

2. ターゲットホストに鍵が正常にコピーされたことを確認するには、Capsule で以下のコマンドを入力します。

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

3.10. SATELLITE API を使用したリモート実行用の SSH 鍵の取得

Satellite API を使用して Capsule から公開鍵をダウンロードするには、各ターゲットホストでこの手順を実行します。

手順

1. ターゲットホストで `~/.ssh` ディレクトリーを作成し、SSH 鍵を保存します。

```
# mkdir ~/.ssh
```

2. Capsule から SSH 鍵をダウンロードします。

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

3. `~/.ssh` ディレクトリーのパーミッションを設定します。

```
# chmod 700 ~/.ssh
```

4. `authorized_keys` ファイルのパーミッションを設定します。

```
# chmod 600 ~/.ssh/authorized_keys
```

3.11. プロビジョニング中に SSH 鍵を配布するキックスタートテンプレートの設定

`remote_execution_ssh_keys` スニペットをカスタムキックスタートテンプレートに追加して、プロビジョニング中に SSH 鍵をホストにデプロイできます。Satellite に同梱されるキックスタートテンプレートには、デフォルトでこのスニペットが含まれています。したがって、Satellite はプロビジョニング中にリモート実行用の SSH 鍵をシステムにコピーします。

手順

- 新しくプロビジョニングされたホストに公開鍵を含めるには、使用するキックスタートテンプレートに以下のスニペットを追加します。

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

3.12. KERBEROS チケットを付与するための KEYTAB の設定

以下の手順を使用して、Satellite が Keytab を使用して Kerberos 付与チケットを取得するように設定します。Keytab が設定されていないと、手動でチケットを取得する必要があります。

手順

1. `foreman-proxy` ユーザーの ID を特定します。

```
# id -u foreman-proxy
```

2. 新しいファイルのパーミッションが **600** になるように、`umask` の値を変更します。

```
# umask 077
```

3. Keytab のディレクトリーを作成します。

```
# mkdir -p "/var/kerberos/krb5/user/USER_ID"
```

4. Keytab を作成するか、既存の Keytab をディレクトリーにコピーします。

```
# cp your_client.keytab /var/kerberos/krb5/user/USER_ID/client.keytab
```

5. ディレクトリーの所有者を **foreman-proxy** ユーザーに変更します。

```
# chown -R foreman-proxy:foreman-proxy "/var/kerberos/krb5/user/USER_ID"
```

6. keytab ファイルが読み取り専用であることを確認します。

```
# chmod -wx "/var/kerberos/krb5/user/USER_ID/client.keytab"
```

7. SELinux コンテキストを復元します。

```
# restorecon -RvF /var/kerberos/krb5
```

3.13. リモート実行用の KERBEROS 認証の設定

Kerberos 認証を使用して、Satellite ホストでのリモート実行に SSH 接続を確立できます。

前提条件

- Kerberos サーバーに Satellite Server を登録する。
- Kerberos サーバーに Satellite ターゲットホストを登録する。
- リモート実行用に Kerberos ユーザーアカウントを設定して初期化する。
- Satellite の foreman-proxy ユーザーに、チケットを付与する有効な Kerberos チケットがあることを確認する。

手順

1. リモート実行に Kerberos 認証をインストールおよび有効にするには、以下のコマンドを入力します。

```
# satellite-installer --scenario satellite \
--foreman-proxy-plugin-remote-execution-ssh-ssh-kerberos-auth true
```

2. リモート実行のデフォルトユーザーを編集するには、Satellite Web UI で **管理** > **設定** に移動して、**リモート実行** タブをクリックします。SSH User 行で 2 番目のコラムを編集し、Kerberos アカウントのユーザー名を追加します。
3. **remote_execution_effective_user** に移動し、2 番目のコラムを編集して、Kerberos アカウントのユーザー名を追加します。

Kerberos 認証が使用できることを確認するには、ホストでリモートジョブを実行します。

3.14. ジョブテンプレートのセットアップ

Satellite は、ジョブ実行に使用可能なデフォルトのジョブテンプレートを提供します。ジョブテンプレートの一覧を表示するには、**ホスト** > **ジョブテンプレート** に移動します。変更せずにテンプレートを使用する場合には、「[Executing a Remote Job](#)」に進んでください。

デフォルトテンプレートは独自のテンプレートを作成するためのベースとして使用することができます。デフォルトのジョブテンプレートは、編集できないようにロックされています。テンプレートのクローンを作成して、作成したクローンを編集します。

手順

1. テンプレートのクローンを作成するには、**アクション** コラムで、**クローン** を選択します。
2. クローンに一意名を指定して、**送信** をクリックして変更を保存します。

ジョブテンプレートは、Embedded Ruby (ERB) 構文を使用します。テンプレートの作成に関する詳細は、『[ホストの管理](#)』ガイドの「[テンプレート作成の参照](#)」を参照してください。

Ansible の考慮事項

Ansible ジョブテンプレートを作成するには、以下の手順を使用し、ERB 構文ではなく YAML 構文を使用します。テンプレートは `---` で開始します。Ansible playbook YAML ファイルをジョブテンプレートの本文に埋め込みます。また、ERB 構文を追加して、YAML Ansible テンプレートをカスタマイズすることも可能です。Satellite に Ansible Playbook をインポートすることも可能です。詳細情報は、『[ホストの管理](#)』ガイドの「[テンプレートリポジトリーの同期](#)」を参照してください。

パラメーター変数

ランタイム時に、ジョブテンプレートはホストに定義するパラメーター変数を受け取ることができます。ホストの編集ページの **パラメーター** タブで表示されるパラメーターのみが、ジョブテンプレートの入力パラメーターとして使用できる点にご留意ください。ランタイム時に Ansible ジョブテンプレートがパラメーター変数を受け取らない場合には、Satellite Web UI で **管理** > **設定** に移動して、**Ansible** タブをクリックします。トップレベルの **Ansible 変数** の行で、**Value** パラメーターを **No** に変更します。

3.15. リモートジョブの実行

1つ以上のホストに対してジョブテンプレートに基づくジョブを実行することができます。

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. **ホスト** > **すべてのホスト** に移動し、リモートジョブを実行するターゲットホストを選択します。検索フィールドを使用してホストの一覧を絞り込むことができます。
2. **アクションの選択** リストから **リモートジョブのスケジュール** を選択します。
3. **ジョブ呼び出し** ページで、主なジョブ設定を定義します。
4. 使用する **ジョブカテゴリー** および **ジョブテンプレート** を選択します。
5. オプション: **ブックマーク** リストに保存された検索文字列を選択し、ターゲットホストを指定します。
6. オプション: **検索クエリー** を入力し、ターゲットホストの範囲をさらに狭めることができます。**解決** 行には、クエリーの影響を受けるホストの数が表示されます。更新ボタンを押して、クエリー変更後の数を再計算します。プレビューアイコンにはターゲットホストが一覧表示されます。
7. 残りの設定は、選択したジョブテンプレートによって異なります。カスタムパラメーターをテンプレートに追加する方法は「[ジョブテンプレートの作成](#)」を参照してください。

8. オプション: ジョブの詳細設定を行うには、**詳細フィールドの表示** をクリックします。一部の詳細設定はジョブテンプレートによって異なります。以下は一般的な設定です。
 - **実効ユーザー**: ジョブを実行するためにユーザーを定義します。デフォルトは SSH ユーザーです。
 - **同時実行レベル** は一度に実行するジョブの最大数を定義します。この定義で、多数のホストでジョブを実行する時に、システムのリソースに負荷が過剰にかかるのを防ぐことができます。
 - **Timeout to kill** は、ジョブが終了しない場合に、ジョブの強制終了までの間隔 (秒単位) を定義します。以前のタスクが終了するまで時間がかかりすぎているなど、定義した間隔で起動できなかったタスクはキャンセルされます。
 - **クエリーのタイプ**: 検索クエリーが評価されるタイミングを定義します。これは、スケジュールされているタスクに対してクエリーが常に最新の状態に保つのに役立ちます。
 - **実行順序** は、ジョブがホストで実行される順序を決定します (アルファベット順またはランダム)。
同時実行レベル および **Timeout to kill** 設定により、お使いのインフラストラクチャーハードウェアおよびニーズに合わせてジョブ実行を調整することができます。
9. ジョブをすぐに実行する場合は、**スケジュール** が **今すぐ実行** に設定されていることを確認します。さらに、1 回限りのジョブを未来の日付で定義したり、再帰的に実行するジョブを設定することもできます。再帰的に実行するジョブについては、開始日と終了日、実行回数と頻度を定義できます。また cron 構文を使用して繰り返しを定義することもできます。cron の詳細は、Red Hat Enterprise Linux 7 の『**システム管理者のガイド**』の「**システムタスクの自動化**」セクションを参照してください。
10. **送信** をクリックします。これにより **ジョブの概要** ページが表示され、ジョブの完了時にはジョブのステータスも表示されます。

CLI 手順

- Satellite で以下のコマンドを入力します。

```
# hammer settings set --name=remote_execution_global_proxy --value=false
```

カスタムパラメーターを使用してリモートジョブを実行するには、以下の手順を実行します。

1. 使用するジョブテンプレートの ID を検出します。

```
# hammer job-template list
```

2. テンプレートの詳細を表示して、テンプレートに必要なパラメーターを確認します。

```
# hammer job-template info --id template_ID
```

3. カスタムパラメーターでリモートジョブを実行します。

```
# hammer job-invocation create \  
--job-template "template_name" \  
--inputs key1="value",key2="value",... \  
--search-query "query"
```

`query` は、ホストを定義するフィルター式に置き換えます ("`name ~ rex01`" など)。hammer を使用したリモートコマンド実行に関する詳細については、`hammer job-template --help` および `hammer job-invocation --help` を入力します。

3.16. ジョブの監視

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

Ansible ジョブは、ホスト 100 台で一括して実行するので、特定のホストで実行するジョブをキャンセルできません。Ansible Playbook を全ホスト上で一括して実行してからでないと、ジョブは完了しません。

手順

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

CLI 手順

実行中のジョブの進捗を監視するには、次の手順を実行します。

1. ジョブの ID を検出します。

```
# hammer job-invocation list
```

2. ジョブの出力を監視します。

```
# hammer job-invocation output \  
--id job_ID \  
--host host_name
```

3. オプション: ジョブをキャンセルするには、次のコマンドを入力します。

```
# hammer job-invocation cancel \  
--id job_ID
```

第4章 RED HAT SATELLITE と ANSIBLE TOWER の統合

Red Hat Satellite と Ansible Tower を統合して、Ansible Tower の動的インベントリーソースとして Satellite Server を使用します。

また、ホストまたは Ansible Tower のいずれかから、Satellite が管理するホストで Playbook を実行するようにプロビジョニングコールバック機能を使用できます。Satellite Server から新しいホストをプロビジョニングする場合は、プロビジョニングコールバック機能により、Ansible Tower から Playbook の実行をトリガーできます。Playbook は、キックスタートのデプロイメント後にホストを設定します。

4.1. SATELLITE SERVER を動的インベントリー項目として ANSIBLE TOWER に追加

Satellite Server をダイナミックインベントリー項目として Ansible Tower に追加するには、Ansible Tower に Satellite Server ユーザーの認証情報を作成し、Ansible Tower ユーザーをその認証情報に追加してから、インベントリーソースを設定する必要があります。

前提条件

- 数万台のホストを管理するなど、Satellite のデプロイメントが大規模な場合には、管理者以外のユーザーを使用すると、認証の確認中にタイムペナルティーが発生するため、パフォーマンスに悪影響を及ぼす可能性があります。大規模なデプロイメントでは、管理ユーザーの使用を検討する。
- 管理者以外のユーザーは、Satellite Server ユーザーに、**Ansible Tower Inventory Reader** ロールを割り当てる。ユーザー、ロール、パーミッションフィルターの管理に関する詳細は、『Red Hat Satellite の管理』の「[ロールの作成および管理](#)」を参照してください。
- Satellite Server と Ansible Tower を同じネットワークまたはサブネット上にホストすること。

手順

1. Ansible Tower Web UI で、Satellite に対して認証情報を作成します。認証情報の作成方法は『[Ansible Tower ユーザーガイド](#)』の「[新規認証情報の追加](#)」および「[Red Hat Satellite 6 認証情報](#)」を参照してください。

表4.1 Satellite の認証情報

認証情報の種類	Red Hat Satellite 6
Satellite 6 URL:	https://satellite.example.com
ユーザー名	統合ロールを持つ Satellite ユーザーのユーザー名
パスワード	Satellite ユーザーのパスワード

2. 新しい認証情報に Ansible Tower ユーザーを追加します。ユーザーを認証情報に追加する方法は『[Ansible Tower ユーザーガイド](#)』の「[認証情報の使用開始](#)」を参照してください。
3. 新しいインベントリーを追加します。詳細は『[Ansible Tower ユーザーガイド](#)』の「[新規インベントリーの追加](#)」を参照してください。

4. 新規インベントリーで、Satellite Server をインベントリースourceとして追加し、以下のインベントリースourceオプションを指定します。詳細は、『Ansible Tower ユーザーガイド』の「[ソースの追加](#)」を参照してください。

表4.2 インベントリースourceオプション

ソース	Red Hat Satellite 6
認証情報	Satellite Server 用に作成した認証情報
上書き	選択
変数の上書き	選択
起動時の更新	選択
キャッシュのタイムアウト	90

5. 追加したソースを同期するようにしてください。

4.2. ホストへのプロビジョニングコールバックの設定

Satellite でホストを作成すると、Ansible Tower を使用して Playbook を実行して新たに作成したホストを設定できます。これは、Ansible Tower では、**プロビジョニングコールバック** と呼ばれます。

プロビジョニングコールバック機能を使用して、プロビジョニングプロセスの一部として、Ansible Tower から Playbook の実行をトリガーします。Playbook は、キックスタートのデプロイメント後にホストを設定します。

プロビジョニングコールバックの詳細は『Ansible Tower ユーザーガイド』の「[プロビジョニングコールバック](#)」を参照してください。

Satellite Server の **Kickstart Default** テンプレートおよび **Kickstart Default Finish** テンプレートには、以下の3つのスニペットが含まれます。

1. **ansible_provisioning_callback**
2. **ansible_tower_callback_script**
3. **ansible_tower_callback_service**

ホストまたはホストグループにパラメーターを追加して、新規作成されたホストでの Ansible Playbook の実行にスニペットが使用可能な認証情報を指定できます。

前提条件

プロビジョニングコールバックを設定する前に、Satellite を動的インベントリーとして Ansible Tower に追加する必要があります。詳細は、「[Satellite と Ansible Tower の統合](#)」を参照してください。

Ansible Tower Web UI で、以下のタスクを実行する必要があります。

1. 新規ホスト向けに、マシンの認証情報を作成します。Satellite で作成したホストに割り当てる予定の認証情報に、同じパスワードを入力するようにしてください。詳細は、『Ansible Tower ユーザーガイド』の「[新規認証情報の追加](#)」を参照してください。

2. プロジェクトを作成します。詳細は、『Ansible Tower ユーザーガイド』の「プロジェクト」を参照してください。
3. ジョブテンプレートをプロジェクトに追加します。詳細情報は、『Ansible Tower ユーザーガイド』の「ジョブテンプレート」を参照してください。
4. ジョブテンプレートで、プロビジョニングコールバックを有効にし、ホストの設定キーを生成して、ジョブテンプレートの `template_ID` をメモする必要があります。ジョブテンプレートの詳細は『Ansible Tower ユーザーガイド』の「ジョブテンプレート」を参照してください。

手順

1. Red Hat Satellite Web UI で、**設定** > **ホストグループ** に移動します。
2. ホストグループを作成するか、既存のホストグループを編集します。
3. ホストグループウィンドウで **パラメーター** タブをクリックします。
4. **パラメーターの追加** をクリックします。
5. 新規パラメーターごとに、以下の情報を入力します。

表4.3 ホストパラメーター

名前	値	説明
ansible_tower_provisioning	true	プロビジョニングコールバックを有効にします。
ansible_tower_fqdn	tower.example.com	Ansible Tower の完全修飾ドメイン名 (FQDN)。 https は Ansible Tower が追加するので、追加する必要はありません。
ansible_job_template_id	template_ID	テンプレートの URL で確認できるプロビジョニングテンプレートの ID (<code>/templates/job_template/5</code>)。
ansible_host_config_key	config_KEY	Ansible Tower のジョブテンプレートが作成したホスト設定キー。

6. **送信** をクリックします。
7. ホストグループを使用してホストを作成します。
8. 新規ホストで、以下のコマンドを入力して、**ansible-callback** サービスを開始します。

```
# systemctl start ansible-callback
```

9. 新規ホストで、以下のコマンドを入力して、**ansible-callback** サービスのステータスを出力します。

```
# systemctl status ansible-callback
```

プロビジョニングコールバックが正しく設定されていると、以下の出力が返ります。

```
SAT_host systemd[1]: Started Provisioning callback to Ansible Tower...
```

手動でのプロビジョニングコールバック

プロビジョニングコールバック URL およびホストからのホスト設定キーを使用して Ansible Tower を呼び出します。例:

```
# curl -k -s --data curl --insecure --data host_config_key=my_config_key \  
https://tower.example.com/api/v2/job_templates/8/callback/
```

プロビジョニングコールバック URL の入力時には、**https** を使用するように入力してください。

これにより、ホストに対して、テンプレートで指定した Playbook の実行がトリガーされます。

付録A ジョブテンプレートの例および拡張

このセクションは、要件に合わせたジョブテンプレートの修正、カスタマイズ、および拡張に役立つ参照情報として使用できます。

A.1. ジョブテンプレートのカスタマイズ

ジョブテンプレートの作成時に、テンプレートエディターフィールドで既存のテンプレートを追加できます。こうすることで、テンプレートを組み合わせたり、一般的なテンプレートからより具体的なテンプレートを作成したりできます。

以下のテンプレートを使用してデフォルトのテンプレートを組み合わせ、Red Hat Enterprise Linux システムに `httpd` サービスをインストールして起動できます。

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

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

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

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

A.2. デフォルトのジョブテンプレートカテゴリー

ジョブテンプレートのカテゴリー	説明
Packages	パッケージ関連のアクションを実行するためのテンプレートです。デフォルトで、インストール、更新、および削除アクションが含まれています。
Puppet	ターゲットホストで Puppet ホストを実行するためのテンプレートです。
Power	パワー関連のアクションを実行するためのテンプレートです。デフォルトで、再起動およびシャットダウンアクションが含まれます。
Commands	リモートホストでカスタムコマンドを実行するためのテンプレートです。
Services	サービス関連のアクションを実行するためのテンプレートです。デフォルトで、開始、停止、再起動、およびステータスアクションが含まれます。

ジョブテンプレートのカテゴリー	説明
Katello	コンテンツ関連のアクションを実行するためのテンプレートです。これらのテンプレートは主として Satellite Web UI の各種の場所 (たとえば、コンテンツホストの一括操作のための UI など) で使用されますが、エラータのインストールなどの各種操作を実行するために個別に使用できます。

A.3. RESTORECON テンプレートの例

この例は、**Run Command - restorecon**というテンプレートを作成する方法を示します。これは、ターゲットホストで選択したディレクトリー内の全ファイルに対して、デフォルトの SELinux コンテキストを復元します。

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

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

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

3. **ジョブ** タブで、**ジョブカテゴリー** を **Commands** に設定します。
4. **入力を追加** をクリックして、ジョブのカスタマイズを可能にします。 **名前** フィールドに **directory** と入力します。入力する名前は、テンプレートエディターで指定した値と一致している必要があります。
5. **必須** をクリックし、ユーザーがパラメーターを指定しなければコマンドが実行しないようにします。
6. **入力タイプ** リストから **ユーザー入力** を選択します。ジョブの呼び出し中に表示する説明を入力します (例: **Target directory for restorecon**)。
7. **送信** をクリックします。

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

A.4. RESTORECON テンプレートのレンダリング

この例は、「[restorecon テンプレートの例](#)」で作成される **Run command - restorecon** テンプレートから派生するテンプレートを作成する方法を示しています。このテンプレートでは、ジョブ実行時にユーザーが入力する必要はなく、ターゲットホストの `/home/` ディレクトリー下のすべてのファイルで SELinux コンテキストを復元します。

「[ジョブテンプレートのセットアップ](#)」に従って新規テンプレートを作成し、テンプレートエディター画面で以下の文字列を指定します。

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

A.5. 複数のホストでの RESTORECON テンプレートの実行

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

1. **ホスト** > **すべてのホスト** に移動し、ターゲットホストを選択します。アクションの選択リストで、**リモートジョブのスケジュール** を選択します。
2. **ジョブ呼び出し** ページで、**Commands** ジョブカテゴリーを選択し、**Run Command - restorecon** ジョブテンプレートを選択します。
3. **ディレクトリ** フィールドに `/home` と入力します。
4. **スケジュール** を **Execute now** に設定します。
5. **送信** をクリックします。ジョブ呼び出し ページに移動します。ここでジョブ実行のステータスを監視できます。

A.6. テンプレートにパワー操作を組み込む

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

「[ジョブテンプレートのセットアップ](#)」に従って新規テンプレートを作成し、テンプレートエディター画面で以下の文字列を指定します。

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