Red Hat Satellite 6.12

Red Hat Satellite での Ansible 統合を使用した設定の管理

Satellite で Ansible 統合を使用してホストシステムを設定する

Last Updated: 2024-02-15

Red Hat Satellite 6.12 Red Hat Satellite での Ansible 統合を使用した設定の管理

Satellite で Ansible 統合を使用してホストシステムを設定する

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

このガイドでは、Red Hat Satellite を Ansible と統合する方法と、Satellite で Ansible ロールと Playbook 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方法を説明します。

目次

RED HAT ドキュメントへのフィードバック (英語のみ) 4

第1章 SATELLITE での ANSIBLE を使い始める 5
1.1. サポートされている ANSIBLE のバージョン 5
1.2. ANSIBLE ロールを実行するための SATELLITE の設定 5
1.3. ANSIBLE と SATELLITE の統合の有効化 6
1.4. ANSIBLE のロールと変数のインポート 6
1.5. SATELLITE での ANSIBLE 変数のオーバーライド 6
1.6. RED HAT ENTERPRISE LINUX システムロールの追加 8
1.7. ANSIBLE コレクションの同期 9

第2章 ANSIBLE ロールを使用したクライアント上の反復タスクの自動化 11
2.1. 既存ホストへの ANSIBLE ロールの割り当て 11
2.2. ホストからの ANSIBLE ロールの削除 11
2.3. ANSIBLE ロールの順序を変更 11
2.4. ホストでの ANSIBLE ロールの実行 12
2.5. ホストグループへの ANSIBLE ロールの割り当て 12
2.6. ホストグループでの ANSIBLE ロールの実行 13
2.7. チェックモードでの ANSIBLE ロールの実行 13

第3章 リモートジョブの設定およびセットアップ 14
3.1. ホストでのジョブの実行について 14
3.2. リモート実行のワークフロー 14
3.3. リモート実行用のパーミッション 15
3.4. リモート実行用のトランスポートモード 16
3.5. プル要求を使用するためのホストの設定 17
3.6. ジョブテンプレートの作成 18
3.7. 名前別での ANSIBLE PLAYBOOK のインポート 19
3.8. 利用可能なすべての ANSIBLE PLAYBOOK のインポート 20
3.9. 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 20
3.10. SATELLITE でのグローバル CAPSULE リモート実行の設定 21
3.11. 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 21
3.12. 権限昇格方法の変更 22
3.13. リモート実行のための SSH 鍵の配布 22
3.14. リモート実行用の SSH 鍵の手動での配布 23
3.15. SATELLITE API を使用したリモート実行用の SSH 鍵の取得 23
3.16. プロビジョニング中に SSH キーを配布するためのキックスタートテンプレートの設定 23
3.17. KERBEROS チケットを付与するための KEYTAB の設定 24
3.18. リモート実行用の KERBEROS 認証の設定 24
3.19. ジョブテンプレートのセットアップ 25
3.20. リモートジョブの実行 26
3.21. ホストの定期的な ANSIBLE ジョブのスケジュール 27
3.22. ホストグループの定期的な ANSIBLE ジョブのスケジュール 28
3.23. ジョブの監視 28

第4章 RED HAT SATELLITE と ANSIBLE AUTOMATION CONTROLLER の統合 30
4.1. SATELLITE SERVER を動的インベントリー項目として ANSIBLE AUTOMATION CONTROLLER に追加する 30
4.2. ホストへのプロビジョニングコールバックの設定 31

第5章 ジョブテンプレートの例および拡張 34
5.1. ジョブテンプレートのカスタマイズ 34
5.2. デフォルトのジョブテンプレートカテゴリー 34

5.3. RESTORECON テンプレートの例 35
5.4. RESTORECON テンプレートのレンダリング 35
5.5. 複数のホストでの RESTORECON テンプレートの実行 36
5.6. テンプレートにパワー操作を組み込む 36

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 SATELLITE での ANSIBLE を使い始める

このガイドを使用して、リモート実行に Ansible を使用するように Satellite を設定します。

1.1. サポートされている ANSIBLE のバージョン

Satellite は、Satellite Server のベースオペレーティングシステムまたはリモート実行用の Capsule によって提供される Ansible を使用します。したがって、サポートされる Ansible のバージョンは、ベース OS の設定によって異なります。

1.2. ANSIBLE ロールを実行するための SATELLITE 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。Ansible は、Satellite Server でデフォルトで有効になっています。

Ansible パス

Satellite は、**/etc/ansible/ansible.cfg** の設定に基づいてパスから Ansible ロールと変数をインポートします。その後、Satellite は **/etc/foreman-proxy/ansible.cfg** の設定に基づいてパスからインポートされたロールを実行します。どちらの場合も、Satellite は **roles_path** および **collections_path** ディレクティブからパスを読み取ります。これら 2 つの cfg ファイルを同期させておいてください。そうしないと、実行できないロールをインポートしたり、実行できるロールが表示されなくなったりする可能性があります。

設定ファイルでパスが指定されていない場合は、次のデフォルトパスが使用されます。

- **/etc/ansible/roles**
- **/usr/share/ansible/roles**
- **/etc/ansible/collections**
- **/usr/share/ansible/collections**

手順

1. ロールを使用する Satellite Server およびすべての Capsule Server で、Ansible パス を設定します。
2. ロールを使用する場所から、Satellite Server およびすべての Capsule Server の Ansible パス 内の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 Server と Capsule Server の間でロールを同期するように設定してください。
3. ホストで Ansible ロールの実行に使用するすべての Capsule Server で、Ansible プラグインを有効にします。

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

4. 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** 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。
5. Ansible ロールを Satellite にインポートします。

6. *2章Ansible ロールを使用したクライアント上の反復タスクの自動化* に進みます。

1.3. ANSIBLE と SATELLITE の統合の有効化

Satellite Server で Ansible プラグインを有効にするには、次の手順を実行します。

手順

- Satellite Server で Ansible プラグインを有効にします。

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

1.4. ANSIBLE のロールと変数のインポート

Ansible が有効になっている Satellite Server または Capsule の Ansible パス から、Ansible のロールと変数をインポートできます。

一部のロールは、他のロールよりもインポートに時間がかかることに注意してください。

前提条件

- インポートするロールおよび変数が、ロールを使用するすべての Capsule の Ansible パス に配置されているようにしてください。

手順

1. Satellite Web UI で、**Configure** > **Roles** に移動します。
2. **Import** をクリックして、インポート元の Capsule を選択します。
3. インポートするロールを選択します。
4. **Submit** 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。

次の手順では、ホストとホストグループについて説明します。詳細は、**ホストの管理** を参照してください。

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で行われるようにするには、変数の優先順位: 変数をどこに置くべきか を参照してください。

前提条件

- Satellite に Ansible 変数がある。詳細は、「Ansible のロールと変数のインポート」 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. **Default Behavior** エリアで、**Override** チェックボックスを選択します。
4. **Parameter Type** フィールドで、**string** や **boolean** などの検証用の値のタイプを選択します。**配列** タイプと **ハッシュ** タイプには、変数の一致を処理するための追加のオプションがあります。詳細は、以下の Prioritize Attribute Order エリア を参照してください。
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: 変数の値を Satellite Web UI でプレーンテキストとして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、変数のコンテンツをアスタリスクとして表示します。これは、パスワードやシークレットトークンなどの機密性の高い値に役立ちます。
7. オプション: **オプションの入力バリデーター** エリアをデプロイメントし、変数の具体的な値を検証するために使用される条件を指定します。
   - ユーザーにこの変数への入力を強制する場合は、**Required** を選択します。
   - **Validator Type** フィールドで、値を検証する方法を選択します。
     - **list** - 値は、許可された値の列挙に対して検証されます。
     - **regex** - 値は正規表現パターンに対して検証されます。
8. オプション: **Prioritize Attribute Order** 領域では、オーバーライドをホストと一致させる優先順位をホスト属性別に指定します。上から順に優先されます。最初に一致したものが優先されます。
   AND 演算としてコンマを使用して、複数の属性を 1 つのマッチャーキーにまとめることができます。たとえば、**hostgroup, environment** のマッチャーキーは、**hostgroup = "web servers"** および **environment = production** などのマッチャーを想定します。

   パラメータータイプ **array** または **hash** を使用する場合は、さらに以下を設定できます。

   - **オーバーライドのマージ** - 配列全体またはハッシュを置き換える代わりに、配列/ハッシュのメンバーをマージします。ハッシュに同じキーが含まれている場合、値はホストの値で上書きされます。
   - **デフォルトのマージ** - デフォルト値を配列またはハッシュに追加します。
   - **重複の回避** - 配列またはハッシュの値が一意であることを確認します。
9. オプション: **マッチャーの指定** エリアをデプロイメントし、変数がオーバーライドするホストを選択するための基準を指定します。
10. 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。

ホストへの変数の追加

1. Satellite Web UI で、**ホスト** > **すべてのホスト** に移動し、使用するホストを選択します。

2. **Ansible** タブをクリックし、**Variables** 領域で、鉛筆アイコンをクリックして変数の値を編集します。

3. チェックアイコンをクリックして変更された変数の値を受け入れるか、クロスアイコンをクリックして変更をキャンセルします。

ホストグループへの変数の追加

1. Satellite Web UI で、**Configure** > **Host Groups** に移動し、使用するホストグループを選択します。

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。

変数をグローバルパラメーターとして追加する

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イスです。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. 次のリポジトリーが有効になっていることを確認します。

   - Red Hat Enterprise Linux 8 で、**Appstream** リポジトリーが有効になっていることを確認します。

     ```
     # subscription-manager repos --enable=rhel-8-for-x86_64-appstream-rpms
     ```

–

アーキテクチャー用に指定された Appstream リポジトリーを有効にする必要があります。詳細は、RHEL 8 リポジトリー を参照してください。

- Red Hat Enterprise Linux 7 では、**Extras** リポジトリー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 ロールリスト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てから、**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、「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。

1.7. ANSIBLE コレクションの同期

Satellite では、Private Automation Hub、**console.redhat.com**、およびその他の Satellite インスタンスからの Ansible コレクションを同期できます。Ansible コレクションは、同期後、**コンテンツ** の下の Satellite Web UI メニューの新しいリポジトリータイプとしてSatellite に表示されます。

手順

1. Satellite Web UI で、**Content** > **Products** に移動します。

2. 必要な製品名を選択します。

3. **製品** ウィンドウで、リポジトリーを作成する製品の名前を選択します。

4. **リポジトリー** タブをクリックして、**新規リポジトリー** をクリックします。

5. **Name** フィールドに、リポジトリーの名前を入力します。
ラベル フィールドには、名前に基づいて自動的に入力されます。

6. **タイプ** リストから、**ansible コレクション** を選択します。

7. **アップストリーム URL** フィールドに、アップストリームコレクションリポジトリーの URL を入力します。
URL には、任意の Ansible Galaxy エンドポイントを指定できます。例:
https://console.redhat.com/api/automation-hub/

8. オプション: **Requirements.yml** フィールドで、エンドポイントから同期するコレクションのリストとそのバージョンを指定できます。
   コレクションのリストを指定しない場合は、エンドポイントのすべてが同期されます。

```
---
collections:
- name: my_namespace.my_collection
  version: 1.2.3
```

   詳細は、**Galaxy ユーザーガイド** の 要件ファイルを使用した複数のコレクションのインストール を参照してください。

9. 認証します。

   a. **Private Automation Hub** から Satellite を同期するには、 **Auth Token** フィールドにトークンを入力します。
      詳細は、**Connect to Hub** ガイドの Connect Private Automation Hub を参照してください。

   b. **console.redhat.com** から Satellite を同期するには、 **Auth Token** フィールドにトークンを入力し、**Auth URL** フィールドに SSO URL を入力します。
      詳細は、**Automation Hub の概要** を参照してください。

   c. Satellite から Satellite を同期するには、両方の認証フィールドを空白のままにします。

10. **保存** をクリックします。

11. Ansible Collections リポジトリーに移動します。

12. **アクションの選択** メニューから **同期開始** を選択します。

第2章 ANSIBLE ロールを使用したクライアント上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

Ansible ロールを使用して、Satellite クライアントのリモート管理を行うことができます。

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、**Hosts > All Hosts** に移動します。
2. ホストを選択し、**編集** をクリックします。
3. **Ansible Roles** タブで、**Available Ansible Roles** のリストから追加するロールを選択します。
4. **+** アイコンをクリックして、ロールをホストに追加します。複数のロールを追加できます。
5. **Submit** 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オーバーライド

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. ホストからの ANSIBLE ロールの削除

次の手順を使用して、Ansible ロールをホストから削除します。

手順

1. Satellite Web UI で、**Hosts > All Hosts** に移動します。
2. ホストを選択し、**編集** をクリックします。
3. **Ansible Roles** タブを選択します。
4. **Assigned Ansible Roles** エリアで、**-** アイコンをクリックして、ホストからロールを削除します。さらにロールを削除するには、これを繰り返します。
5. **Submit** をクリックします。

2.3. ANSIBLE ロールの順序を変更

次の手順を使用して、ホストに適用される Ansible ロールの順序を変更します。

手順

1. Satellite Web UI で、**Hosts > All Hosts** に移動します。
2. ホストを選択します。
3. **Ansible Roles** タブを選択します。
4. **Assigned Ansible Roles** エリアで、ロールを適切な位置にドラッグアンドドロップすることで、ロールの順序を変更できます。
5. **Submit** をクリックして、Ansible ロールの順序を保存します。

2.4. 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、「Ansible ロールを実行するための Satellite 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、**Hosts** > **All Hosts** 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. **Select Action** リストから、**Run all Ansible roles** 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。

2.5. ホストグループへの ANSIBLE ロールの割り当て

Ansible ロールを使用して、Satellite クライアントのリモート管理を行うことができます。

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、「Ansible ロールを実行するための Satellite の設定」 を参照してください。

手順

1. Satellite Web UI で、**Configure** > **Host Groups** に移動します。
2. Ansible ロールを割り当てるホストグループ名をクリックします。
3. **Ansible Roles** タブで、**Available Ansible Roles** のリストから追加するロールを選択します。
4. **+** アイコンをクリックして、ロールをホストグループに追加します。複数のロールを追加できます。

5. **Submit** をクリックします。

2.6. 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、「Ansible ロールを実行するための Satellite 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、**Configure** > **Host Groups** に移動します。
2. ホストグループの **Actions** コラムのリストから、**Run all Ansible roles** 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。

2.7. チェックモードでの ANSIBLE ロールの実行

Satellite Web UI を使用して、チェックモードで 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、「Ansible ロールを実行するための Satellite 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、**ホスト** > **すべてのホスト** に 移動 します。
2. チェックモードを有効にするホストの **編集** をクリックします。
3. **パラメーター** タブで、ホストに **ansible_roles_check_mode** という名前のパラメーターがあり、**boolean** 型が **true** に設定されていることを確認します。
4. **Submit** をクリックします。

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。詳細は、「リモート実行用のトランスポートモード」 を参照してください。

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** 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、**ホストの管理** の ジョブテンプレートのセットアップ を参照してください。

注記

Capsule Server のベース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、**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。

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、**Remote execution** 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イスを検出します。
2. Satellite はこれらのインターフェイ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。

Prefer registered through Capsule for remote executionを有効にした場合、Satellite はホストが登録されている Capsule を使用して REX ジョブを実行します。

デフォルトでは、**Prefer registered through Capsule for remote execution**は **No** に設定されています。これを有効にするには、Satellite Web UI で **Administer** > **Settings** に移動し、**Content** タブで、**Prefer registered through Capsule for remote execution** を **Yes** に設定します。これにより、Satellite が登録先の Capsule によってホスト上で REX ジョブを実行することが保証されます。

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 server Capsule
- Puppet CA Capsule
- OpenSCAP Capsule

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

インフラストラクチャー内でどのロールをどのジョブを実行できるか (ターゲットにするホストを含めて) 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**: すべてのリモート実行機能にアクセスできます。
- **リモート実行ユーザー**: ジョブの実行のみ可能です。

Remote Execution User 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。カスタマイズされたロールで **view_job_templates** パーミッションを使用して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。

execute_jobs_on_infrastructure_hosts パーミッションを使用して、Red Hat Satellite にホストとして登録された Red Hat Satellite および Capsule に対してリモート実行ジョブを実行できます。標準の **Manager** と **Site Manager** のロールには、デフォルトでこのパーミッションがあります。**Manager** ま

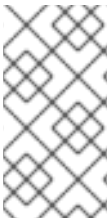
たは **Site Manager** ロールのいずれかを使用する場合、または **execute_jobs_on_infrastructure_hosts** パーミッションを持つカスタムロールを使用する場合は、登録済みの Red Hat Satellite および Capsule ホストに対してリモートジョブを実行できます。

ロールおよびパーミッションの使用に関する詳細は、**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。

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
```

この例の最初の行を使用して、選択した 1 つのホストに **Reboot** テンプレートを適用します。2 行目を使用して、名前の末尾が **.staging.example.com** となるように、ホストのプールを定義します。3 行目を使用して、テンプレートをホストグループにバインドします。



注記

これらのロールを持つユーザーに割り当てられるパーミッションは、時間の経過とともに変化する可能性があります。将来実行するジョブがすでにスケジュールされている場合に、パーミッションが変更されると、ジョブの実行直前にパーミッションが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. リモート実行用のトランスポートモード

Satellite が、リモートジョブ実行に 2 つの異なるトランスポートモードを使用するように設定できます。

ssh モードの Capsule では、リモート実行は SSH サービスを使用してジョブの詳細を転送します。これは、デフォルトのトランスポートモードです。ターゲットホストで SSH サービスを有効にし、アクティブにする必要があります。リモート実行 Capsule は、ターゲットホストの SSH ポートにアクセスできる必要があります。別の設定がない限り、標準の SSH ポートは 22 です。

pull-mqtt モードの Capsules では、リモート実行は Message Queueing Telemetry Transport (MQTT) を使用して Satellite Server から受信するジョブを公開します。ホストは、**yggdrasil** プルクライアントを使用してジョブ通知のために Capsule の MQTT ブローカーをサブスクライブします。ホストが通知を受信したら、HTTPS 経由で Capsule からジョブ詳細をプルし、ジョブを実行して結果を Capsule に報告します。

pull-mqtt モードを使用するには、Capsule Server でこれを有効にし、ターゲットホストでプルクライアントを設定する必要があります。

関連情報

- Capsule Server でプルモードを有効にするには、**Capsule Server のインストール** の プルクライアントのリモート実行の設定 を参照してください。
- 既存のホストでプルモードを有効にするには、「プル要求を使用するためのホストの設定」 に進みます。
- Katello Agent からホストを移行するには、**ホストの管理** の Katello Agent からリモート実行への移行 を参照してください。

- 新しいホストでプルモードを有効にするには、**ホストの管理** に記載された次のいずれかの手順に進みます。
  - ホストの作成
  - ホストの登録

3.5. プル要求を使用するためのホストの設定

pull-mqtt モードを使用するように設定された Capsule の場合には、ホストはリモート実行プルクライアントを使用してリモートジョブにサブスクライブできます。管理対象ホストでは、Capsule Server への SSH 接続は必要ありません。

前提条件

- ホストを Satellite に登録している。
- ホストの Capsule が **pull-mqtt** モードを使用するように設定されている。詳細は、**Capsule Server のインストール** の プルクライアントのリモート実行の設定 を参照してください。
- Red Hat Satellite Client 6 リポジトリーは、Satellite Server で有効化および同期され、ホストで有効化されている。
- ホストは、ポート **1883** を使用して MQTT 経由で Capsule と通信できる。
- ホストは、HTTPS 経由で Capsule と通信できる。

注記

katello-pull-transport-migrate パッケージが作成され、ユーザーはプルクライアントを使用して Katello Agent からリモート実行に移行できるようになりました。ただし、ホストに Katello Agent をインストールする必要はありません。Katello Agent がインストールされているかどうかに関係なく、**katello-pull-transport-migrate** を使用できます。

手順

1. ホストに **katello-pull-transport-migrate** パッケージをインストールします。
   - Red Hat Enterprise Linux 8 ホストおよび Red Hat Enterprise Linux 9 ホストの場合:

     ```
     # dnf install katello-pull-transport-migrate
     ```

   - Red Hat Enterprise Linux 7 ホストの場合:

     ```
     # yum install katello-pull-transport-migrate
     ```

   このパッケージは、依存関係として **foreman_ygg_worker** および **yggdrasil** をインストールし、ホストでプルモードを有効にします。ホストの **subscription-manager** 設定およびコンシューマー証明書は、ホストで **yggdrasil** クライアントを設定するために使用され、プルモードのクライアントワーカーを起動します。

2. オプション: プルクライアントが実行中であり、適切に設定されていることを確認するには、**yggdrasild** サービスのステータスを確認します。

   ```
   # systemctl status yggdrasild
   ```

–

3. オプション: パッケージのインストール後に、ホストから **katello-agent** を削除できます。

> **警告**
>
> お使いのホストが Red Hat Virtualization のバージョン 4.4 以下にインストールされている場合は、削除された依存関係によりホストが破損するので、**katello-agent** パッケージを削除しないでください。

3.6. ジョブテンプレートの作成

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

手順

1. Satellite Web UI で、**ホスト** > **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. **Job** タブをクリックし、**Job category** フィールドに独自のカテゴリーを入力するか、**ホストの管理** の デフォルトのジョブテンプレートカテゴリー にリストされているデフォルトカテゴリーから選択します。
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. **プロバイダータイプ** 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** の テンプレート作成リファレンス および ジョブテンプレートの例と拡張機能 を参照してください。

CLI 手順

- 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_My_Template_File" \
--job-category "My_Category_Name" \
--name "My_Template_Name" \
--provider-type SSH
```

3.7. 名前別での ANSIBLE PLAYBOOK のインポート

Capsule にインストールされているコレクションから Satellite に、名前を使用して Ansible Playbook をインポートできます。

前提条件

- Satellite の Ansible プラグインが有効になっている

手順

1. 以下の API 要求を使用して、利用可能な Ansible Playbook を取得します。

```
# curl -X GET 'Content-Type: application/json'
https://satellite.example.com/ansible/api/v2/ansible_playbooks/fetch?
proxy_id=My_capsule_ID
```

2. インポートする Ansible Playbook を選択し、その名前を書き留めます。

3. その名前を使用して Ansible Playbook をインポートします。

```
# curl -X PUT 'Content-Type: application/json' -d '{ "playbook_names":
["My_Playbook_Name"] }'
https://satellite.example.com/ansible/api/v2/ansible_playbooks/sync?
proxy_id=My_capsule_ID
```

インポートが完了すると、Satellite Web UI で通知が表示されます。

3.8. 利用可能なすべての ANSIBLE PLAYBOOK のインポート

利用可能なすべての Ansible Playbook は、Capsule にインストールされているコレクションから Satellite にインポートできます。

前提条件

- Satellite の Ansible プラグインが有効になっている

手順

- 以下の API 要求を使用して Ansible Playbook をインポートします。

```
# curl -X PUT 'Content-Type: application/json'
https://satellite.example.com/ansible/api/v2/ansible_playbooks/sync?proxy_id=My-capsule-ID
```

  インポートが完了すると、Satellite Web UI で通知が表示されます。

3.9. 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 のリストからリモート実行 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 server Capsule
- Puppet CA Capsule
- OpenSCAP Capsule

手順

1. Satellite Web UI で、**Administer** > **Settings** に移動します。
2. **Remote Execution** 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。値を **true** に設定するには、次のコマンドを入力します。

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

3.10. 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、**Administer** > **Settings** に移動します。
2. **Remote Execution** 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。値を **true** に設定するには、次のコマンドを入力します。

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

3.11. 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。

```
# mkdir /My_Remote_Working_Directory
```

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 /My_Remote_Working_Directory
```

3. システムを設定します。

```
# satellite-installer \
--foreman-proxy-plugin-ansible-working-dir /My_Remote_Working_Directory
```

3.12. 権限昇格方法の変更

デフォルトでは、プッシュベースのリモート実行は **sudo** を使用して、SSH ユーザーから、ホスト上でスクリプトを実行する実効ユーザーに切り替えます。状況によっては、**su** や **dzdo** などの別の方法の使用が必要になる場合があります。Satellite 設定で、別の方法をシステム全体に設定できます。

前提条件

- ユーザーアカウントに、**view_settings** および **edit_settings** 権限を付与するロールが割り当てられている。
- Ansible ジョブに **dzdo** を使用する場合は、必要な **dzdo** become プラグインを含む **community.general** Ansible コレクションがインストールされていることを確認する。詳細は、**Ansible ドキュメント** の Installing collections を参照してください。

手順

1. **Administer** > **Settings** に移動します。
2. **Remote Execution** タブを選択します。
3. **Effective User Method** 設定の値をクリックします。
4. 新しい値を選択します。
5. **Submit** をクリックします。

3.13. リモート実行のための SSH 鍵の配布

ssh モードの Capsule の場合、リモート実行接続は SSH を使用して認証されます。Capsule からの公開 SSH 鍵は、管理するアタッチ済みのホストに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 キーを配布するための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、**ホストの管理** の グローバル登録テンプレートを使用した Red Hat Satellite へのホストの登録 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、New User Accounts を参照してください。

3.14. 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

- SSH 公開キーを Capsule からターゲットホストにコピーします。

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

  管理するターゲットホストごとにこの手順を繰り返します。

検証

- 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@client.example.com
```

3.15. 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
```

3.16. プロビジョニング中に SSH キーを配布するための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 キー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。Satellite は、プロビジョニング中にリ

モート実行用の SSH キー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' %>
```

3.17. 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/My_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 My_Client.keytab /var/kerberos/krb5/user/My_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/My_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/My_User_ID/client.keytab"
```

7. SELinux コンテキストを復元します。

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

3.18. 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-script-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 **Administer** > **Settings** に移動して、**Remote Execution** 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。

検証

- 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。詳細は、**Red Hat Satellite での Ansible Integration を使用した設定の管理**の リモートジョブの実行 を参照してください。

3.19. 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リストを表示するには、**ホスト** > **ジョブテンプレート** に移動します。テンプレートを変更せずに使用する場合は、**ホストの管理** の リモートジョブの実行 に進みます。

デフォルトテンプレートは独自のテンプレートを作成するためのベースとして使用することができます。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。テンプレート作成の詳細は、**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。詳細は、**ホストの管理** の レポジトリーテンプレートの同期 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。

3.20. 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。

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

手順

1. Satellite Web UI で、**ホスト** > **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リスト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 **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。プレビューアーイコンにはターゲットホストがリスト表示されます。

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の詳細は、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、定期的なジョブを設定することもできます。定期的なジョブについては、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、cron での Linux システムタスクの自動化 を参照してください。

10. **Submit** をクリックします。同じページの **Recent Jobs** セクションでジョブのステータスを表示できます。

CLI 手順

1. 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
```

2. 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
```

3. 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 My_Template_ID
```

4. 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 \
--inputs My_Key_1="My_Value_1",My_Key_2="My_Value_2",... \
--job-template "My_Template_Name" \
--search-query "My_Search_Query"
```

My_Search_Query は、ホストを定義するフィルター式 (例: **"name ~ My_Pattern ")** に置き換えます。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。

3.21. ホストの定期的な ANSIBLE ジョブのスケジュール

ホストで Ansible ロールを実行する定期的なジョブをスケジュールできます。

前提条件

- **view_foreman_tasks**、**view_job_invocations**、および **view_repeat_logics** 権限があることを確認する。

手順

1. Satellite Web UI で、**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。

2. **Ansible** タブで、**ジョブ** を選択します。

3. **Schedule recurring job** をクリックします。
4. **Create New Recurring Ansible Run**ウィンドウで、最初の実行の繰り返し頻度、開始時刻、日付を定義します。
5. **Submit** をクリックします。
6. オプション: ホストの概要で、または **Ansible** > **Jobs** に移動して、スケジュールされた Ansible ジョブを表示します。

3.22. ホストグループの定期的な ANSIBLE ジョブのスケジュール

ホストグループで Ansible ロールを実行する定期的なジョブをスケジュールできます。

手順

1. Satellite Web UI で、**Configure** > **Host groups** に移動します。
2. **Actions** 列で、Ansible ロールの実行をスケジュールするホストグループの **Configure Ansible Job** を選択します。
3. **Schedule recurring job** をクリックします。
4. **Create New Recurring Ansible Run**ウィンドウで、最初の実行の繰り返し頻度、開始時刻、日付を定義します。
5. **Submit** をクリックします。

3.23. ジョブの監視

ジョブの実行中に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. Satellite Web UI で、**Monitor > Jobs**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リストが表示されます。
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

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 \
--host "My_Host_Name" \
--id My_Job_ID
```

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 My_Job_ID
```

第4章 RED HAT SATELLITE と ANSIBLE AUTOMATION CONTROLLER の統合

Red Hat Satellite と Ansible Automation Controller を統合して、Satellite Server を Ansible Automation Controller の動的インベントリーソースとして使用できます。Ansible Automation Controller は、Red Hat Ansible Automation Platform のコンポーネントです。

また、ホストまたは Ansible Automation Controller 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 Automation Controller から Playbook の実行をトリガーできます。Playbook は、プロビジョニングプロセスの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 AUTOMATION CONTROLLER に追加する

Satellite Server を動的インベントリー項目として Ansible Automation Controller に追加するには、Ansible Automation Controller で Satellite Server ユーザーの認証情報を作成し、Ansible Automation Controller ユーザーを認証情報に追加して、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 Automation Controller を同じネットワークまたはサブネット上でホストする必要があります。

手順

1. Ansible Automation Controller Web UI で、Satellite の認証情報を作成します。認証情報の作成の詳細は、**Automation Controller ユーザーガイド** の 新しい認証情報の追加 および Red Hat Satellite 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 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. Ansible Automation Controller ユーザーを新しい認証情報に追加します。認証情報へのユーザーの追加の詳細については、**Automation Controller ユーザーガイド** の 認証情報の開始 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、**Automation Controller ユーザーガイド** の 新しい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 をインベントリーソースとして追加し、以下のインベントリーソースオプションを指定します。詳細は、**Automation Controller 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 インベントリーソースオプション

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場合、Ansible Automation Controller を使用して Playbook を実行し、新しく作成したホストを設定できます。これは、Ansible Automation Controller 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関数は、プロビジョニングプロセスの一部として Ansible Automation Controller から Playbook の実行をトリガーします。Playbook は、プロビジョニングプロセスの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、**Automation Controller 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 Ansible Automation Controller の動的インベントリーとして追加する必要があります。詳細は、Satellite と Ansible Automation Controller の統合を参照してください。

Ansible Automation Controller Web UI では、次のタスクを完了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、**Automation Controller ユーザーガイド** の 新しい認証情報の追加 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 Automation Controller 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。詳細は、**Automation Controller 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、**Automation Controller 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. 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	**controller.example.com**	Ansible Automation Controller の完全修飾ドメイン名 (FQDN)。**https** は Satellite によって追加されるため、追加しないでください。
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 (**/templates/job_template/5**)。
ansible_host_config_key	**config_KEY**	ジョブテンプレートが Ansible Automation Controller で生成するホスト設定キー。

6. **Submit** 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 Automation Controller...
```

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 とホストからのホスト設定キーを使用して、Ansible Automation Controller を呼び出すことができます。

```
# curl -k -s --data curl --insecure --data host_config_key=my_config_key \
https://controller.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。

第5章 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。

5.1. 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。

次のテンプレートは、クライアントに **nginx** サービスをインストールして開始するためのデフォルトのテンプレートを組み合わせたものです。

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

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、コンマ区切りのリストを指定することができます。

5.2. 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。

5.3. 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。

手順

1. Satellite Web UI で、**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. **Submit** をクリックします。詳細は、**ホストの管理** の 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。

5.4. 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。このテンプレートはジョブの実行時にユーザー入力を必要としません。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に SELinux コンテキストが復元されます。

ジョブテンプレートの設定 に従って新しい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") %>
```

5.5. 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/** ディレクトリー内にある全ファイルの SELinux コンテキストを復元します。

手順

1. Satellite Web UI で、**Hosts** > **All hosts** に移動して、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. **ディレクトリー** 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. **Submit** 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。

5.6. 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設定 に従って新しい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") %>
```