



# Red Hat Ansible Automation Platform 2.2

## Automation Services Catalog と IT Service Management (ITSM) システムの統合

オーダープロセスおよび置換可能な変数で定義したワークフローを使用して、IT Service Management システムツールチェーンに Automation Services Catalog を組み込む



## Red Hat Ansible Automation Platform 2.2 Automation Services Catalog と IT Service Management (ITSM) システムの統合

---

オーダープロセスおよび置換可能な変数で定義したワークフローを使用して、IT Service Management システムツールチェーンに Automation Services Catalog を組み込む

## 法律上の通知

Copyright © 2023 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js<sup>®</sup> 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<sup>®</sup> 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

本書では、オーダープロセスおよび置換可能な変数を使用して、Ansible Automation Platform のジョブテンプレートから IT Service Management (ITSM) システムにデータを渡す方法について説明します。

---

## 目次

はじめに .....	3
多様性を受け入れるオープンソースの強化 .....	4
第1章 ITSM と AUTOMATION SERVICES CATALOG との統合の計画 .....	5
第2章 AUTOMATION SERVICES CATALOG との統合をサポートするための PLAYBOOK SET_STATS フィールドの更新 .....	6
2.1. 値を AUTOMATION SERVICES CATALOG に戻す PLAYBOOK の作成 .....	6
第3章 オーダープロセスを使用した ITSM 統合での順次アクションの定義 .....	8
3.1. オーダープロセスの作成 .....	8
3.2. 製品の順序プロセスの設定 .....	9
第4章 オーダープロセスワークフロー全体で置換可能な変数の有効化 .....	10
4.1. オーダープロセスワークフローへの置換可能な変数の実装 .....	10
4.2. 置換変数の参照 .....	12



## はじめに

Automation Services Catalog のオーダープロセス機能を使用して、ServiceNow などの Information Technology Service Management (ITSM) システムと統合できます。



### 重要

自動化サービスカタログのサポートは、2.4 以降の Ansible Automation Platform では利用できなくなりました。

## 多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) を参照してください。



## 第1章 ITSM と AUTOMATION SERVICES CATALOG との統合の計画

Ansible Automation Platform および Playbook インベントリに必要な更新と変更について計画し、ITSM システムと統合します。変更によっては、Ansible Tower と Automation Services Catalog の両方で、特定レベルの管理パーミッションが必要になる場合があります。

本セクションでは、更新の実行に必要な実際のロールと、変更する必要があるアーティファクトについて詳しく説明します。

### 必須のパーミッション

インテグレーションワークフローを完了するには、以下のロールが必要です。

- ITSM 統合に使用する Playbook を作成または更新できます。
- Ansible Tower の管理者で、ジョブテンプレートを作成してサーベイを追加できます。
- カタログ管理者は、ポートフォリオの作成、サーベイの編集、およびオーダープロセスの設定が可能です。

### Playbook

ITSM と Automation Services Catalog との統合に使用する予定のジョブテンプレートおよびワークフローで使用される Playbook を作成または更新します。

### Ansible Tower のジョブおよびワークフローのテンプレートサーベイ

Playbook の前後に実行するジョブテンプレートにアタッチされたサーベイを更新して、置換可能な値をサポートします。

### Automation Services Catalog

ITSM 統合ワークフローの製品に設定されたサーベイを更新し、置換を有効にします。

## 第2章 AUTOMATION SERVICES CATALOG との統合をサポートするための PLAYBOOK `set_stats` フィールドの更新

Playbook を更新して、Automation Services Catalog に情報を渡して、オーダープロセスワークフローの製品全体にわたる置換可能な変数をサポートすることができます。

### 2.1. 値を AUTOMATION SERVICES CATALOG に戻す PLAYBOOK の作成

値を Automation Services Catalog に戻すように設計された Playbook を作成できます。`set_stats` パラメーターの前に `expose_to_cloud_redhat_com_` を加えると、その値が Automation Services Catalog に戻されます。次に、置換可能な変数を使用して、追加の Playbook でこれらの値を渡すことができます。

以下の Playbook の例を使用して `set_stats` の値について確認し、値を Automation Services Catalog に戻し、これをオーダープロセスの後続の Playbook の置換可能な値として使用します。

#### before order Playbook の例

この `before order` Playbook は、お気に入りの色を Automation Services Catalog `set_stats` 値に戻します。

```
# This playbook prints a simple debug message and set_stats
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
      msg: "Hello, world!"

    - set_stats:
      data:
        expose_to_cloud_redhat_com_favorite_color: "orange"
```

#### product order Playbook の例

この Playbook は、アーティファクトの値として Automation Services Catalog に置換可能な値を渡します。

```
# This playbook prints a simple debug message with given information
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
      msg: "Hello, {{favorite_color}} world!"

    - set_stats:
      data:
        expose_to_cloud_redhat_com_after_data: "{{favorite_color}}"
```

#### after order Playbook の例

以下の after order Playbook の例には、製品 Playbook によって渡されたアーティファクトの **after\_data** 値が含まれます。

```
# This playbook prints a simple debug message with given information
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
      msg: "{{after_data}}"
~
```

## 第3章 オーダープロセスを使用した ITSM 統合での順次アクションの定義

オーダープロセスは、ユーザーがオーダーした製品を ITSM システムと統合するための主要な Automation Services Catalog 機能です。オーダープロセスは、ITSM システムでアクションを実行するジョブテンプレートを実行することを目的とする Automation Services Catalog の製品で設定されています。これらのプロセスは、**Before order** および **After order** アクションで設定されています。オーダープロセスは、ユーザーがオーダーする製品の前後に実行されるように設計された製品として定義し、これにより、情報がオーダーシーケンス全体に渡されるようにします。

### シーケンスの例

- **before order** - ITSM システムでチケットを作成する製品。
- **Product order** - Web サーバー。
- **after order** - ITSM システムでチケットを完了する製品。

製品またはポートフォリオレベルのいずれかに適用するオーダープロセスを作成することにより、シーケンスと実行される製品を定義します。

このワークフローの例では、オーダープロセスを作成し、**before order**、**product order**、および **after order** に割り当てる必要があります。

### 3.1. オーダープロセスの作成

カタログ管理者は、製品が順序付けされる前後に Ansible Tower Playbook を実行できるオーダープロセスを作成します。

#### 前提条件

- Ansible Tower Playbook でオーダープロセスを作成するには、Ansible Tower クラスターをソースとして追加する必要があります。

#### 手順

1. メインナビゲーションから **Order Processes** を選択します。
2. **作成** をクリックします。
3. オーダープロセス名と説明を指定します。
4. ドロップダウンメニューから、オーダープロビジョニング前後に発生するアクションを選択します。



#### 注記

ドロップダウンメニューには、Ansible Tower ソースからプルした Playbook が表示されます。各オーダープロセスでサポートされるのは、1つの Before および After アクションのみです。

5. 新しい Order Process の詳細を確認してから **Create** をクリックします。

## 3.2. 製品の順序プロセスの設定

単一製品に適用される順序プロセスを設定します。

### 手順

1. メインナビゲーションから **Products** を選択し、製品を選択します。
2. **More actions** をクリックし、**Set order processes** を選択します。
3. ドロップダウンメニューを展開し、オーダープロセスを選択します。



### 注記

現在、Ansible Automation Platform は製品プロビジョニングごとに1つのオーダープロセスのみをサポートします。

4. **Save** をクリックします。

## 第4章 オーダープロセスワークフロー全体で置換可能な変数の有効化

オーダープロセスワークフローで使用する製品全体で変数を置き換えることができます。置換可能な変数を実装するには、Ansible Tower でジョブテンプレートのサーベイを作成または編集し、Automation Services Catalog の製品にアタッチされたサーベイで置換を有効にする必要があります。

### 置換形式および要件

置換可能な値は、`{{substitution_express}}` 形式で表され、波括弧と式の間にスペースは使用できません。置換に使用できる各変数の詳細は、[Substitute variables](#) を参照してください。

### 4.1. オーダープロセスワークフローへの置換可能な変数の実装

本セクションでは、本書に記載の Playbook の例を使用して、Playbook の Automation Services Catalog に公開するデータを置換可能な変数に合わせる方法を説明します。

#### 4.1.1. before order プロセスを作成して製品オーダーへの値の指定

##### before order プロセスの Playbook `before_order.yml` の例

```
# This playbook prints a simple debug message and set_stats
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
      msg: "Hello, world!"

    - set_stats:
      data:
        expose_to_cloud_redhat_com_favorite_color: "orange"
```

この Playbook は、値が **オレンジ** 色の Playbook アーティファクト **favorite\_color** を Automation Services Catalog に戻します。ユーザー入力が必要ないため、Ansible Tower でジョブテンプレートサーベイを作成する必要はありません。代わりに、before オーダープロセス製品に設定された Automation Services Catalog の Survey を編集することにより、**favorite\_color** を製品 Playbook に渡すことができます。

#### 4.1.2. 置換変数を使用するための製品オーダーの設定

この製品 Playbook の例では、`{{favorite_color}}` の置換された変数を受け入れ、アーティファクト `after_data` を Automation Services Catalog に戻します。

##### Playbook `product_order.yml` の例

```
# This playbook prints a simple debug message with given information
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
```

```

msg: "Hello, {{favorite_color}} world!"

- set_stats:
  data:
    expose_to_cloud_redhat_com_after_data: "{{favorite_color}}"

```

`favorite_color` を \*Answer Variable Name として設定する `product_order.yml` Playbook のジョブテンプレートを Ansible Tower に作成します。

以下を含む `product_order.yml` ジョブテンプレートの Survey を作成します。

1. **Prompt** フィールドに **What is your favorite color?** と入力します。
2. **Answer Variable Name** フィールドに `favorite_color` と入力します。
3. **UPDATE** をクリックします。

Automation Services Catalog の `product order` Survey を有効にして、`before_order.yml` から渡された `'favorite_color'` の値を使用します。

`product order` に割り当てられた Survey で、以下を行います。

1. 以下の値でフィールドを更新します。
  - a. **Name:** `favorite_color`
  - b. **Initial Value:** `{{before.before_order.artifacts.favorite_color}}`
  - c. **Label:** `What is your favorite color?`
  - d. **Placeholder:** `{{before.before_order.artifacts.favorite_color}}`
2. **Disabled** スイッチが有効になるように切り替えます。
3. **Substitution** スイッチを有効になるように切り替えます。

これで、置換可能な変数として `before_order.yml` Playbook から `favorite_color` アーティファクトを使用するように `product order` を設定できました。

#### 4.1.3. 製品オーダーから `after_data` アーティファクトの値を受け入れるために `after order` 製品の設定

Survey を使用して、代入変数を使用する `after_order.yml` に、`product_order.yml` アーティファクト `after_data` を渡します。

`after order` 製品の Playbook `after_order.yml` の例

```

# This playbook prints a simple debug message with given information
- name: Echo Hello, world!
  hosts: localhost
  gather_facts: yes
  tasks:
    - debug:
      msg: "{{after_data}}"

```

`after_data` を \*Answer Variable Name として設定する `after_order.yml` Playbook のジョブテンプレートを Ansible Tower に作成します。

`after_order.yml` ジョブテンプレートに割り当てられた Survey で、以下を実行します。

1. **Prompt** フィールドに Enter your after data と入力します。
2. **Answer Variable Name** フィールドに `after_data` と入力します。
3. **UPDATE** をクリックします。

Automation Services Catalog の **after order** Survey を有効にして、`after_order.yml` から渡された `after_data` の値を使用します。

**after order** に割り当てられた Survey で、以下を行います。

1. 以下の値でフィールドを更新します。
  - a. **Name:** `after_data`
  - b. **Initial Value:** `{{product.artifacts.after_data}}`
  - c. **Label:** "Enter your after data"
  - d. **Placeholder:** `{{product.artifacts.after_data}}`
2. **Disabled** スイッチが有効になるように切り替えます。
3. **Substitution** スイッチを有効になるように切り替えます。

Playbook は、ジョブテンプレートの実行時に **after data** に渡される値を表示します。

## 4.2. 置換変数の参照

置換の形式は `{{substitution_express}}` です。波括弧と式の間スペースを使用しないでください。

置換される値は、Ansible Tower のジョブテンプレートのサーベイに設定されます。置換した変数は、設定に応じて、オーダープロセス内のあるプロダクトから別のプロダクトに渡されます。

ジョブテンプレートの Survey の更新に関する詳細は、Ansible Tower **ユーザーガイド** の [Surveys](#) を参照してください。

サーベイに置換可能な変数を含める場合は、以下の表に従ってください。

表4.1 オブジェクトモデルの参照テーブル

オブジェクト	式	備考
--------	---	----



オブジェクト	式	備考
Order	<b>order.order_id</b> <b>order.created_at</b> <b>order.ordered_by.name</b> <b>order.ordered_by.email</b> <b>order.approval.updated_at</b> <b>order.approval.decision</b> <b>order.approval.reason</b>	order.created_at` は、オーダーを開始した日時です。  日付形式: <b>2007-02-10T20:30:45Z</b>  例: <code>{{order.approval.reason}}</code>
Before order	<b>before.&lt;order_process_name&gt;.artifacts.&lt;fact_name&gt;</b>  <b>before.&lt;order_process_name&gt;.parameters.&lt;parameter_name&gt;</b>  <b>before.&lt;order_process_name&gt;.status</b>	order_process_name にスペースが含まれる場合、引用符は必要ありません。  例: <code>{{before.Sample Order.artifacts.ticket_number}}</code>
Product	<b>product.name</b> <b>product.description</b> <b>product.long_description</b> <b>product.help_url</b> <b>product.support_url</b> <b>product.vendor</b> <b>product.platform</b> <b>product.portfolio.name</b> <b>product.portfolio.description</b> <b>product.artifacts.&lt;fact_name&gt;</b> <b>product.parameters.&lt;parameter_name&gt;</b> <b>product.status</b>	例: <code>{{product.artifacts.after_data}}</code>

オブジェクト	式	備考
After order	<b>after.&lt;order_process_name&gt;.artifacts. &lt;fact_name&gt;</b>  <b>after. &lt;order_process_name&gt;.parameters. &lt;parameter_name&gt;</b>  <b>after.&lt;order_process_name&gt;.status</b>	order_process_name にスペースが含まれる場合、引用符は必要ありません。  例: {{after.SampleOrder.status}}